Robust Reputation System for Web Services

Zhou Xin
Department of Social Informatics
Kyoto University, Japan

Robust Reputation System for Web Services

Doctoral Thesis Series of
Ishida & Matsubara Laboratory
Department of Social Informatics
Kyoto University

Abstract

With various services available on the Internet nowadays, the interactions between human are increasingly intensified. Those interactions may out of internal pleasure or external factors such as profit etc. When one has no or little experience with the service candidates, latent risk may case one suffer loss when engaged in the interaction. Fortunately, reputation system arisen as a tool to support users make decision about which potential service to choose from. A reputation system collects and aggregates feedback from users and predicts the future behavior of a service provider. A couple of reputation systems are proposed in recent years, but they are either vulnerable to unfair rating or suffer heavily from time lag problem. Furthermore, as most reputation system are built upon the ratings given by service consumers. When the rating is sparse, uncertainty on the reputation generated by those reputation systems will increased.

The objective of this thesis is to design a robust reputation system to facilitate the selection of service for consumers. For the dynamic variation of reputation, we present two approaches to distinguish the unfair ratings and build robust reputation system from endogenous and exogenous (micro and macro) view respectively. And when the service is in its early stage, few ratings will lead reputation into vulnerable state, we explore extra information to boost up the reputation to an equilibrium. We address the following research topics with respect to time lag, unfair rating attacks and rating scarcity:

1. Building robust reputation timely under dynamic environment.

In the open, dynamic environment, reputation variation can be caused by unfair ratings or management of service. To distinguish the unfair ratings and reflect the reputation timely, we propose a dynamic sliding window model based on the Bayesian linear regression approach that is capable of reflecting the reputation values according to the latest changes in services. Furthermore, we implement a statistical strategy based on the hypothesis test method to filter out unfair ratings by calculating the distribution of the ratings after using linear regression to transpose the two-dimensional linear window into a constant one-dimensional window. Experiments not only validate the effectiveness of the proposed model, but also show that it outperforms the existing reputation system by 45% in relieving the time lag problem based on 5 test cases.

2. Building robust reputation that can resist the coordinate unfair rating attacks.

   Malicious service consumers may collude with each other to perform unfair rating attacks. We present a clustering-based reputation model that is robust to various unfair rating attacks. The model categorizes consumers as either honest or dishonest according to their rating ratio. It utilizes the Dirichlet distribution in determining reputation values. We analyze the profits and costs attained by the attacker and elucidate the conditions under which an attack is profitable. Through analysis, we assert that our model is able to deal with the situation where large data size received each day. Besides, we illustrate the heuristic power of our model for designer to implement their specific sanctioning function to capture the property of the service with different types by example. Experiments demonstrate that our clustering-based reputation model is more robust than the state-of-art model against currently successful attacks.

3. Providing reputation in rating scarcity environment.

   Fewer ratings will lead the reputation system into useless or vulnera-

ble situation. We address the rating scarcity problem through a novel reputation model that uses the Elo algorithm to consider consumer implicit information in a graph analysis approach. Theoretical analysis is conducted to identify the sufficient and necessary condition for the model to converge to a stable state. To facilitate the selection of Web services for specific preference clusters, we further introduce the reputation metric wise algorithm to rank the Web services according to consumer preference. Furthermore, experiments confirm our model outperforms the widely adopted reputation algorithm in both accuracy and convergence in the situation of rating scarcity. Especially, on real services, the proposed algorithm can improve the ranking availability by 60.4% on average for services in their cold-start stage.

In general, the proposed reputation model aims to provide selection assistance for service consumers and act as an incentive for good performance of service providers in the platform. The reputation model not only can deal with services that have ran for a long time but also give support for new deployed service to establish their early reputation value without transactions.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Toru Ishida, for his acceptance, patience, supervision, advice and guidance throughout my doctoral study. He leaded me the right way to conduct research, even pushed me in the right direction sometimes. I am amazed by his immense knowledge, he referred Confucius's words to illustrate me the importance of practice when doing research. I am extremely fortunate to have you as my teacher who inspired me to be a superior man and to grow as a research scientist. I am also thankful for giving me this lifetime opportunity in the amazingly beautiful Kyoto.

I also owe my sincere gratitude to my thesis committee members, Professor Hajime Kia and Professor Katsuya Yamori for their insightful advice and contribution on my research.

I am very grateful to Associate Professor Shigeo Matsubara, Associate Professor Hiromitsu Hattori, Associate Professor David Kinny, Assistant Professor Yohei Murakami and Assistant Professor Donghui Lin who had paid so much time to accompany me with fruitful discussions and gave their practical advices and closely monitor on my research progress. I thank them for co-authoring papers with me.

I would like to thank all faculty members of Ishida&Matsubara Lab: Researcher Masayuki Otani, Researcher Takao Nakaguchi. I also greatly appreciate our coordinators, Ms. Hiroko Yamaguchi, Ms. Terumi Kosugi, Ms. Yoko Kubota for their help in administrative affairs before I am enrolled.

Special thanks also go to all my lab mates: Ari Hautassari, Andrew W.

Vargo, Huan Jiang, Chunqi Shi, Mairidan Wushouer, Amit Pariyar, Kemas Muslim Lhaksmana, Trang Mai Xuan, Shinsuke Goto, Hiroaki Kingetsu, Xun Cao, Nan Jin, Wenya Wu, Nguyen Cao Hong Ngoc, Arbi haza Hasution, Victoria Abou Khalil, Shohei Hida, Akihiko Itoh, Hiromichi Cho, Taketo Sasaki, Kaori Kita, Daisuke Kitagawa, Jun Matsuno and many others. Thank you all for your assistance on my life in Japan. I will always cherish the time we spent together. I believe there is time we will meet again in somewhere.

I also want to thank my friends and family for their directly or indirectly support: my wife Man Shen, parents Hongxian Zhou and Xueqin Zhang, my uncle Zuyin Zhang and his family, my elder brother Heng Zhou, my younger sister Na Zhou and my son Yilu Zhou for their love and support over the years; my friends in JCCCH church for praying on my research.

My stay in Kyoto University was supported by the Japanese Government (Monbukagakushon) Scholarships from October 2012 to September 2015. This research was partially supported by a Grant-in-Aid for Scientific Research (S) (24220002, 2012-2016) from Japan Society for Promotion of Science (JSPS). Also many thanks to the support from the Telecommunication Advancement Foundation on my travel to attend the international conference.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The booming popularity of service-oriented computing environment, such as Amazon Web Services and Language Grid [Ishida, 2011], inevitably attracts malicious users intending to benefit from the systems illegally. As service providers, they can deliberately increase their reputation to mislead consumers who have no or little experience and consequently boost their transaction volume. A service consumer, when using an existing service or creating a new service by combining several existing services, may discover several functionally equivalent service providers with, it is assumed, different level of qualities. Usually, a service consumer has no or little direct experience with the candidate service providers. Thus decisions must be made as to which service provider will maximize the consumers benefit. Existing markets use reputation systems to allow the service consumer to evaluate the candidates.

The value of reputation systems has been well supported by both research and the success of reputation-centric e-commerce [Livingston, 2005]. We define reputation as a public metric that is visible to all users, such as the reputation system operated by Amazon, eBay etc. Usually, service providers with high reputation achieve better average outcomes. A study of eBay conducted by Resnick et al. revealed that consumers were willing to pay 8% more to sellers with established reputations than to new sellers [Resnick et al., 2006]. Unfortunately, open e-commerce systems can make service providers vulnerable to attack from malicious providers who collude with others to deliberately give unfair ratings to specific competitors; examples include blackening the reputation of targeted service provider for personal gain [Dellarocas, 2000]. Despite ongoing research on this problem, even advanced reputation models have difficulty in distinguishing the truthful changes caused by service updates from malicious ratings.

Besides, various forms of unfair attacks have been observed and are being studied by the trust and reputation community [Jøsang, 2012, Zhang et al., 2012]. The key unfair rating attacks are listed here: **Constant**: An individual dishonest rater gives constant and unfair ratings to a service provider. **Camouflage**: Dishonest raters gain and then abuse the trust of providers. **Whitewashing**: Dishonest rates try to escape their reputations by using new accounts that have the default value of trust. **Sybil**: A group of dishonest raters gives constant and unfair ratings to a service provider. **Sybil Camouflage**: A group of dishonest raters act together in mounting a Camouflage attack. **Sybil Whitewashing**: A group of dishonest raters act together in conducting a Whitewashing attack. Some reputation system are established on protection of anonymous consumers, it is hard to identify who launched the attack. As for different platforms, we proposed two different approaches in this proposal.

Beyond the unfair rating problem, another problem that has been ignored by most researchers is the time lag in reputation value. The simple and widely used reputation engine in current e-commerce is the averaging method, which is easily understood by both service providers and service consumers. Unfortunately, it cannot timely reflect the dynamic changes of service providers. In some e-commerce companies, a variation of the averaging algorithm, called the fixed sliding window algorithm (with window size of 30 days or so), is adopted to deal with this problem. Previous works [Teacy et al., 2012, Liu et al., 2011] on the unfair rating problem either assume the users have personal experience or ignored the time lag problem. A recent work by Wu et al. [Wu et al., 2013] proposed an olfaction-based algorithm that takes both problems into consideration. However, they assume that there must exist one fair rating at least in 10 consecutive ratings. In this research, we relax the constraint and propose a dynamic sliding window model that addresses both unfair rating and time lag problem. Here, we define the dynamic sliding window as a window whose size varies from time to time. The key issue is how to find the appropriate window size to correctly evaluate recently received ratings. We show that the proposal covers all online services, not just e-market services like eBay or Amazon, and also suits various services such as the service platform Language Grid [Ishida., 2006] and Amazon Mechanical Turk.

Existing service reputation systems, mainly based on the ratings given by service consumers, are one of the most important guides that the consumer has in making a decision, as they reveal how other consumers evaluated the services true ability in real scenarios. An example scenario is illustrated in Figure 1.1, a service consumer try to select one service for interaction among a list of functionally equivalent services. After the interaction, the service consumer gives his opinion on the satisfaction of service by rating. However, the ratings may be very sparse or unreliable for the following reasons:

1. Ratings are *skewed* towards high values [Hu et al., 2009]. Consumers cannot express their opinion truthfully if only numerical ratings are used [Ramn et al., 2014]. Moreover, they care about the impact of their feedback on the services future benefits in the marketplace, and

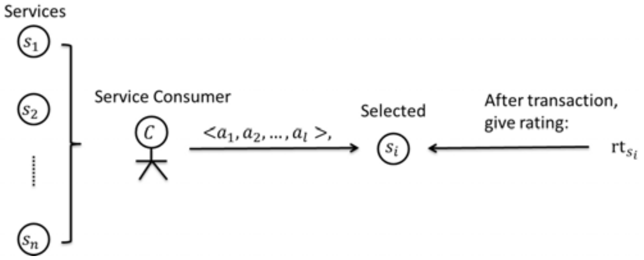


Figure 1.1: Scenario for a service consumer interact with services.

so tend to offer a relatively high value unless extremely unsatisfied.

2. Not all customers rate the transaction [Cabral and Horiacęsu, 2010]. As a result, transaction volume is much larger than the number of ratings received. Normal customers, those who pay for the service, have little interest in entering their ratings unless they are extremely satisfied or unsatisfied.

3. No rating is available at the cold-start stage of a service [Arazy et al., 2009]. Upon the introduction of a service, no consumer has interacted with the service, so no historical evidence can be used to derive a reputation score for the service.

And the rating scarcity problem is rarely addressed in the literature of the service domain.

1.2 Objective

With various services available on the Internet nowadays, the interactions between human are increasingly intensified. Those interaction may out of internal pleasure or external factors such as profit etc. When one has no or little experience with the potential service provider, latent risk may case one suffer loss when engaged in the interaction. The objectives of this thesis to design robust reputation system that deployed in the service platform to facilitate the service selection process of new comers, even under various dynamic or rating sparsity environments. Three motivations are lead into these solutions:

1. Help service consumers who have no or little experience with the services to select the reputable service for interaction. In commercial environment, reputation system may vulnerable to unfair rating attacks from malicious buyers, which may result in some mislead users will transact with dishonest service providers.Methods need to be taken to mitigate the influence caused by those issues.

2. Help service providers act honest in the service platform. Reputation is viewed as the trade name of a firm. In this sense, reputation system can act as an incentive for service providers to improve their service quality. If the reputation system can detect the dishonest behavior of a service provider and give penalty on it, then the reputation system can prevent malicious behaviours and lead positive competition among the service providers.

3. Popularize the newly deployed services even there is no explicit evidence is observed. It is hard for consumers to select new services as there is no ratings to evaluate the performance of the services, which lead to large risk to interact with. We focus on how to provide reputation value for these services in the cold-start stage.

1.3 Issues and Approach

To build a robust reputation system for service platform, towards different types of platform different models are proposed. In this research, we use dynamic sliding window model to predict the reputation value and filter the ratings from anonymous consumers. While for identified ratings, we proposed clustering-based approach and punish the unfair rater by sanction function. For rating scarcity, we explore the implicit factors to promote accuracy and robustness of the reputation system for new deployed services, especially in their cold-start stage. The framework for doctoral research is illustrated in Figure 1.2.

1. Dynamic Sliding Window Approach
   For anonymous ratings, in dynamic sliding window model, the window size for aggregating the received ratings are dynamic changed according to the distribution of the data. When the service provider upgrades or degrades its service quality, the model will detect the changes and adopts itself into the new window. We implement a s-tatistical strategy to filter out unfair ratings by calculating the stan-

dard deviation of the ratings after transposing the two-dimensional linear window into the constant one-dimensional window using linear regression approach.

2. Clustering-based Approach
   The clustering-based approach categorizes identified consumers as either honest or dishonest according to their rating ratio. We accumulate the trustworthiness of the identified consumer by rewarding or punishing it when a honest rating or dishonest rating given at a specific time by the sanction function. The model utilizes the Dirichlet distribution based on the trustworthiness of consumers in determining reputation value.

3. Providing reputation for new service based on implicit information.
   The first research topic deals with unfair ratings problem in anonymous system and reflect the reputation timely. The second research topic focused on the various attacks performed by groups of identified malicious consumers. In those researches, the key problem is how to detect the unfair rating based on ratings. However, if the ratings are sparse or unavailable, the approaches above may not able to generate a reliable reputation value. The third topic focused on the rating scarcity problem and proposed a reputation model that based on the implicit information of the platform to derive the reputation value for service without ratings.

   In a service platform, the decision context is unique because only the platform can observe the behavior of the consumer. Supposing there is always functional equivalent services existed for selection. And when consumers choose between those competing services and make their final decision on the favorite service. The consumer act as a evaluator giving a higher expected reputation value on the selected service over others. Our last research will take advantage of this decision-making process to build reliable support to the accuracy and robustness of the existed reputation systems even in the cold-start stage.

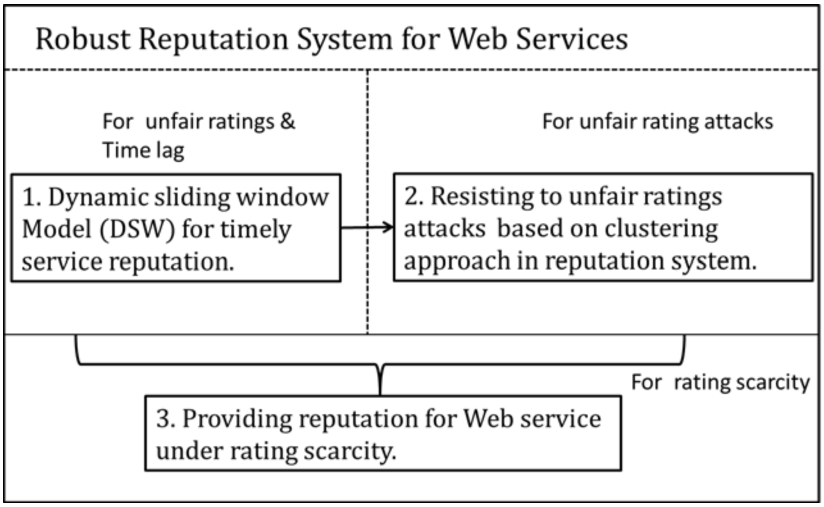

Robust Reputation System for Web Services
For unfair ratings & Time lag
For unfair rating attacks
1. Dynamic sliding window Model (DSW) for timely service reputation.
2. Resisting to unfair ratings attacks based on clustering approach in reputation system.
For rating scarcity
3. Providing reputation for Web service under rating scarcity.

Figure 1.2: The research issues and approaches of the doctoral thesis.

1.4 Thesis Outline

This thesis consists of six chapters including Chapter 1. The content of each of the remaining chapters are summarized next.

Chapter 2 introduces the background of the reputation system. Previous works on unfair rating and time lag problem of reputation system are discussed in this chapter. Besides, researches on rating scarcity are also presented at last.

Chapter 3 proposes a dynamic sliding window model based on the Bayesian linear regression approach that is capable of reflecting the reputation values according to the latest changes in services. Furthermore, we implement a statistical strategy based on the hypothesis test method to filter out unfair ratings by calculating the distribution of the ratings after using linear regression to transpose the two-dimensional linear window into a constant one-dimensional window. Experiments not only validate the effectiveness of the proposed model, but also show that it outperforms the existing reputation system by 45% on average in relieving the time lag problem based on 5 test cases.

Chapter 4 presents a clustering-based reputation model that is robust to various unfair rating attacks. The model categorizes consumers as either honest or dishonest according to their rating ratio. It utilizes the Dirichlet distribution in determining reputation values. We analyze the profits and costs attained by the attacker and elucidate the conditions under which an attack is profitable. Through analysis, we assert that our model is able to deal with the situation where large data size received each day. Besides, we illustrate the heuristic power of our model for designer to implement their specific sanctioning function to capture the property of the service with different types by example. Experiments demonstrate that our clustering-based reputation model is more robust than the state-of-art model against currently successful attacks.

Chapter 5 addresses the rating scarcity problem through a novel reputation model that uses the Elo algorithm to consider consumer implicit information in a graph analysis approach. Theoretical analysis is conducted to identify the sufficient and necessary condition for the model to converge to a stable state. To facilitate the selection of Web services for specific preference consumers, we further introduce the reputation metric wise algorithm to rank the Web services according to consumer preference. Furthermore, experiments confirm our model outperforms the widely adopted reputation algorithm in both accuracy and convergence in the situation of rating scarcity. Especially, on real services, the proposed algorithm can improve the ranking availability by 60.4% on average in the cold-start stage of a service.

Finally Chapter 6 concludes this thesis by discussing the summary of contributions made for building robust reputation system and also suggesting possible future directions.

Chapter 2

Background

Since Resnick et al. [Resnick et al., 2000] pointed out the issues posed by web site reputation, various reputation systems based on feedback have been published. We study the research background of reputation system in this chapter.

2.1 Reputation Model

Reputation systems are typically based on public information such as ratings or reviews given by users in order to reflect the community's opinion. However some reputation system not only take public information into account, but also consider the private experience. When user's preference is integrated into the system, this system is more like a recommendation system, which recommends items to users according to their propensity [Ricci et al., 2011]. Actually, personal information is normally considered more reliable than third party information in reputation system. In this section, we will describe various models for computing reputation value. While some models are wildly used in commercial applications, whereas others have been proposed by the academic community.

2.1.1 Bayesian model

From Bayesian statistic perspective, a posterior probability indicts the future performance of a service. When new ratings observed, the posterior probability will be updated according to prior reputation and prior probability. [Jiang et al., 2012, Jøsang and Ismail, 2002, Whitby et al., 2004, Mui et al., 2001, Mui et al., 2002b, Mui et al., 2002a]

In paper [Teacy et al., 2012], the authors present a model called Hierarchical And Bayesian Inferred Trust Model (HABIT) to assess the trust of a agent in an open and dynamic environments. The model is based on statistical techniques and can integrate other information of a agent to improve the assessment. Simulation and real-world experiments show that the proposed method can predict the agent behavior and is up to twice as accurate as the state-of-the-art model. In an open and dynamic environments, the agent in the system can provide data or services for consuming. From the perspective of consumers, they need to decide which service provider to rely on. Thus, the problem of trust arise as a fundamental issue in this system. Consumers have to assess the trustworthiness of a service provider before making their decisions. In this context, trust assessment models is required to aid decision making and to estimate future behavior of a service provider, so that consumers can decide which provider they will interact with, to minimise their risk and maximise the expected gain. Moreover, in such an open system, malicious service providers can disrupt the system easily using false data or fraudulent service. Thus, the trust model need to be robust enough with the information that is malicious or inaccurate. The HABIT model is a two level Bayesian Network. As in Figure 2.1, in the bottom level, various opinions from different reputation sources are modeled – *confidence* model, and the relationships of different opinions and agent behaviour is modeled in the top level – *reputation* model. In general, for each truster tr, and trustee te, confidence model is used to represent the probability distribution $p(O_{tr \to te} | \theta_{tr \to te})$. Where $O_{tr \to te}$ records the outcome of each interaction as a truster assess the performance of a trustee te. $\theta_{tr \to te}$ is a parameter vector that specifies the distribution. This parameter vector controls how a te will behave and what utility tr will receive. Each $\theta_{.\to j}$ is formed by concatenating all parameter vectors, $\theta_{i\to j}$, where $i \in \Lambda$, Λ is the set of all agents (1, 2, ..., n).

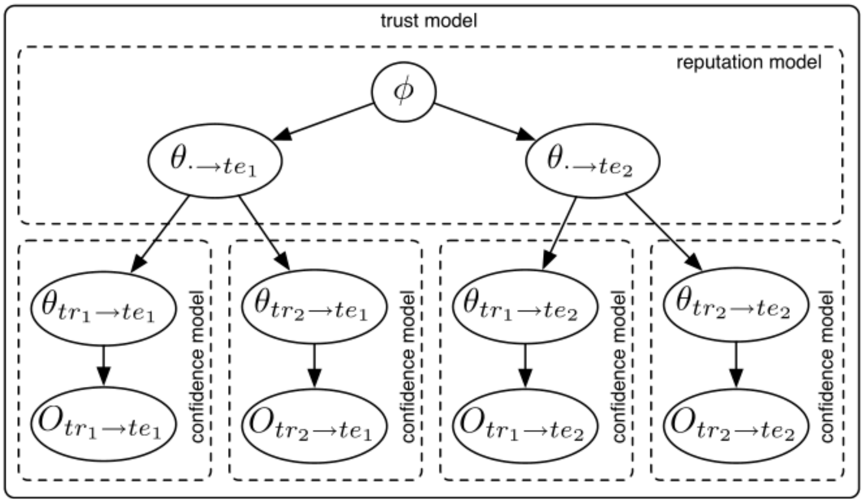


Figure 2.1: The generic HABIT model.

Another Bayesian network model based on three trust information are proposed in [Nguyen et al., 2010]. Those three different kinds of trust sources including the recommendation from other consumers, QoS monitoring and direct experience of the requester. Those trust sources are integrated to reach the reputation value: $T_x(i) = T_{rx}(i) \times \omega_r + T_{cx}(i) \times \omega_c + T_{dx}(i) \times \omega_d$. The weigh factors for the above three sources ω_r, ω_c, ω_d subjected to $\omega_r + \omega_c + \omega_d = 1$. Reputation is viewed as a subjective conception, which predicts future behavior of a service based on its past behaviors. Moreover, the Bayesian network is adopted to calculate the three factors $T_{rx}(i)$, $T_{cx}(i)$, $T_{dx}(i)$ respectively.

The Bayesian model provides a theoretically sound basis for computing reputation value, and the disadvantage is that it might be too complex to understand. Actually, the most easiest way to calculate reputation is to average all the ratings, which will be described below.

2.1.2 Averaged model

Most commercial companies use this type of model. Since it is simple and easy to understand by summing all the ratings together and dividing the number of ratings. Which denotes as $\sum_{i=1}^{n} r_i/n$, where i is i-th rating from users, n the total number of ratings [Schneider et al., 2000]. This model is wildly used by numerous commercial web sites, such as Amazon and Taobao, etc.

Similar way to calculate the reputation value in eBay is simply sum the number of positive ratings and negative ratings separately, and keep the total score as the positive score minus the negative score [Resnick and Zeckhauser, 2002].

More complex extension of averaged model is proposed in paper [Wu et al., 2013]. Based on $\bar{r}_n = \sum_{i=1}^{n} r_i/n$, the author deduced:

$$\begin{aligned} \bar{r}_n &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} r_i \\ &= \tfrac{1}{n}r_1 + \tfrac{1}{n}r_2 + ... + \tfrac{1}{n}r_n \\ &= \tfrac{n-1}{n} \cdot \tfrac{1}{n-1}(r_1 + r_2 + ... + r_{n-1}) + \tfrac{1}{n}r_n \\ &= \tfrac{n-1}{n}\bar{r}_{n-1} + \tfrac{1}{n}r_n. \end{aligned}$$

Based on this, substitute $\frac{1}{n}$ as p, we have $\bar{r}_n = (1-p)\bar{r}_{n-1} + pr_n$. By determining the value of p, the model can lay appropriate weight on the latest rating. Given $p = 0.2$, which means the weight of the latest rating is 0.2,and the following weights are sequential $0.2 \times 0.8, 0.2 \times 0.8^2, 0.2 \times 0.8^3, ..., 0.2 \times 0.8^n$. The proposed olfaction based algorithm also uses the nonlinear model of olfactory fatigue to assign the weight of unfair ratings. As in figure 2.2, after unfair rating is detected, the model enter perception stage. After the unfair rating is cleared, the model moves into fading stage. And the authors proposed OACR1 and OACR2 for the time lag and unfair rating problem. The difference between those two models is in OACR2 the authors used some fixed parameters instead of a nonlinear function to assign the weight of a rating. And experiments proved it performs more accurate to calculate the reputation value than the above averaged model.

The advantage of averaged model is that it easy to understand and accepted by users, the disadvantage is that it is can not express exactly with the behavior of a community or participant. Especially when unfair rating is provided by malicious users.

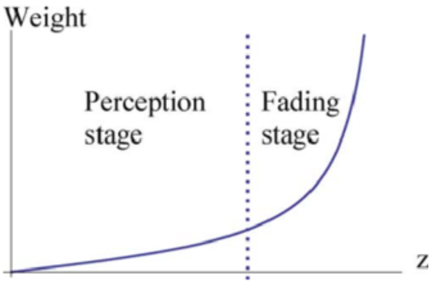


Figure 2.2: The generic HABIT model.

2.1.3 Fuzzy model

Fuzzy logic provides rules to deduce the result from input [Wang, 1999]. It builds some rules which translates the input knowledge into output. For example, if the rating is provided recently, and the similarity between the witness and the truster is high, and the witness is confident in providing rating, then the rating weight is high [Liu et al., 2013]. To run the system, rule table need to be built mapping the input variables to its corresponding degree of reputation. The REGRET reputation system falls into this category [Sabater and Sierra., 2001].

Despite the fuzzy model can effectively mitigate the adverse impact of unfair rating, it requires to determine the weight of the factors when calculates the reputation value. In real situation, it is hard to list all the factors and give a rational weight for each of them.

2.2 Rating Scarcity

Serval studies have proposed reputation models and analyzed the rating scarcity problem in the service-oriented environment [Malik et al., 2009, Jøsang and Quattrociocchi, 2009, Al-Sharawneh et al., 2010]. Rating scarcity or sparsity mainly occurs at the cold-start stage of a service or when a service experiences a long period of inactivity.

The Bayesian reputation system proposed by Jøsang and Quattrociocchi [Jøsang and Quattrociocchi, 2009] addresses the importance of base rate in the cold-start stage. The system assigns an initial value to new services according to the reputation distribution of the community. Although the approach can assign meaningful initial values to services, we argue that its performance is deficient in two aspects: First, the assigned initial value is unfair to new services. Even though it is clear that the base rate can be biased either negatively or positively, we lack the evidence needed to correct the base rate distribution. Otherwise, assigning an arbitrary initial value may unfair to some services. Second, the Bayesian reputation system cannot boost the convergence rate. The convergence rate is not changed by assigning an initial value, the reputation of the service is unstable until enough ratings have been aggregated.

A prediction model based on historical data is proposed in [Malik et al., 2009]. In the situation of rating scarcity, reputation is predicted by a Hide Markov model. Another recommendation system derives the reputation of a service by injecting pseudo users into the system [Park et al., 2006]. The pseudo users rate the service according to attributes of items or users. Unfortunately, it is difficult to establish a valid relationship between attributes and service reputation. Without historical data, a Location-based Matrix Factorization technique via Preference Propagation (LMF-PP) is proposed by Kwangkyu Lee et al. to improve the cold start problem in web service QoS prediction [Lee et al., 2015]. The algorithm try to build connections between location and consumer preference to predict QoS values.

Chapter 3

Building robust reputation timely under dynamic environment

In many dynamic, open and service-oriented computing environments, service consumers must choose one of services to complete their tasks. Due to the scale and dynamic characteristics of these environments, the service consumer may have little or no past experience with the service candidates. To this end, reputation systems are proposed and they have played a crucial role in the success of online service-oriented transactions. Especially in commercial environments, it is necessary to present reputation value in a timely and robust manner by resisting unfair ratings from malicious users.

3.1 Introduction

In recent years, there has been an upsurge of interest in service-oriented computing, which seeks to integrate various computational resources seamlessly and dynamically beyond organizational boundaries. However, the booming popularity of these commercial services, inevitably attracts malicious users intending to benefit from the systems illegally. As service providers, they can deliberately increase their reputation to mislead consumers who have no or little experience and consequently boost their transaction volume. A service consumer, when using an existing service or creating a new service by combining several existing services, may discover several functionally equivalent service providers with, it is assumed, different level of qualities. Usually, a service consumer has no or little direct experience with the candidate service providers. Thus decisions must be made as to which service provider will maximize the consumers benefit. Existing markets use reputation systems to allow the service consumer to evaluate the candidates.

The value of reputation systems has been well supported by both research and the success of reputation-centric e-commerce [Livingston, 2005, Nepal et al., 2011]. We define reputation as a public metric reflecting the consumers' general opinion on the performance of a service and that is visible to all consumers [Jøsang et al., 2007], such as the reputation systems operated by Amazon, eBay etc. The opinion is a subject view from consumers represented by rating or feedback given after the transaction. By definition, service heterogeneity is a key factor for the convergence of reputation value. That is, from the consumer's perspective, the ratings given by consumers should differ from each other. In the extreme case, if all consumers offer the same opinion on one metric of a service, then the reputation value based on those ratings will be biased towards this group of consumers. Usually, service providers with high reputation achieve better average outcomes. A study of eBay conducted by Resnick et al. revealed that consumers were willing to pay 8% more to sellers with established reputations than to new sellers [Resnick et al., 2006]. As reputation systems are increasing important in triggering purchasing decisions by service consumers, service providers will take any and all opportunities to promote their reputation values. In such open e-commerce systems, service providers are vulnerable to attacks from malicious users who collude with other service providers to deliberately give unfair ratings to a specific competitor s, and alter the reputation of the targeted service provider for personal gain [Dellarocas, 2000]. Typically, the malicious ratings involve unwarranted praise or deformation [Wang et al., 2012]. Despite ongoing research on advanced reputation models to deal with this problem, it is still difficult to distinguish the true changes caused by service updates from or purposely unfair ratings.

Beyond the unfair ratings, one problem has been ignored by most researchers: the time lag problem of reputation value. The widely applied reputation engine in current e-commerce is the averaged model, which is easily understood by both service providers and service consumers. Unfortunately, serval studies have shown that it is vulnerable to malicious feedback [Srivatsa et al., 2005, Hoffman et al., 2009]. Besides, it cannot reflect the dynamic changes of service providers in a timely manner. For example, assume a service has received 700 ratings with reputation value of 50. This service improves its quality and next receives 300 ratings with scores of 90. However, the average is $(700 * 50 + 300 * 90)/1000 = 62$, which does not reflect its current or latest reputation. To overcome the time lag problem, some e-commerce companies use a variation of the average algorithm, the fixed sliding window algorithm (with window size of 30 days or so). Previous works [Teacy et al., 2012, Liu et al., 2011] on the unfair rating problem either assume the users have personal experience or failed to address the time lag problem. A recent work by Wu et al. [Wu et al., 2013] proposed an olfaction-based algorithm that takes both problems into consideration. However, they assume there at least one fair rating in 10 consecutive ratings. In this paper, we lift this constraint and propose a dynamic sliding window model that can handle the unfair rating and time lag problems, simultaneously. We also clarify that the applicable domain of the proposal is not limited to e-market services like eBay or Amazon, but includes web services such as the service platform Language Grid [Ishida, 2011] and Amazon Mechanical Turk. Although the concept of sliding window with dynamic size is widely accepted in domains such as networking [Reiser, 1979] and activity recognition [Laguna et al., 2011], but the approaches used to determine the key parameters of window size and shift differ with the domain. Here, we use the concept of the dynamic sliding window, both window size and shift are dynamically updated by the Bayesian linear regression approach. The reason we applied Bayesian linear regression is on the following consideration: 1). Normal regression models need to decide their model complexity by leveraging the over-fitting problem, while the Bayesian linear regression will lead to automatic methods of determining model complexity [Bishop, 2006]. 2). Instead of generate a estimated point alone, the Bayesian linear regression can recover the whole range of inferential solutions.

3.2 Background

The average algorithm with fixed sliding window is widely used in e-commerce due to its emphasis on recent data. Old ratings are removed from the window as new ratings arrive. Therefore, the sliding window algorithm can reflect the latest ratings. However, older ratings are removed regardless of whether service quality is updated or not, which makes the algorithm vulnerable to unfair rating attacks. In this study, we show the dilemma of the fixed sliding window algorithm as background information, and address the dynamic sliding window model in the following section.

3.2.1 Fixed Sliding Window Algorithm

Currently, the most widely used reputation algorithm is the average algorithm as is used on Amazon, Taobao, Hotels.com. It calculates the average of all received ratings n, $\bar{r}_n = \frac{1}{n}\sum_{i=1}^{n} r_i$. Where $\bar{r}_n$ is the reputation value and r_i is the ith rating value. Because the average algorithm divide all ratings by n, this algorithm suffers badly from the time lag problem. In the rapid e-market, this time lag will cause significant loss to the service provider.

Even though the average algorithm suffers heavily from the time lag problem given the variation in the reputation value, it is more effective than other algorithms in some conditions, which we will address in later sections.

In order to counter the time lag problem, the fixed sliding time window is used in commercial reputation systems such as eBay's feedback forum to reflect the service provider's most recent behavior [Jøsang et al., 2007]. However, time lag is still a severe problem, as shown in Figure 3.1, if the fixed window size is a large value. With small fixed window sizes, the system may suffer from unfair ratings. In the figure, the expectation value

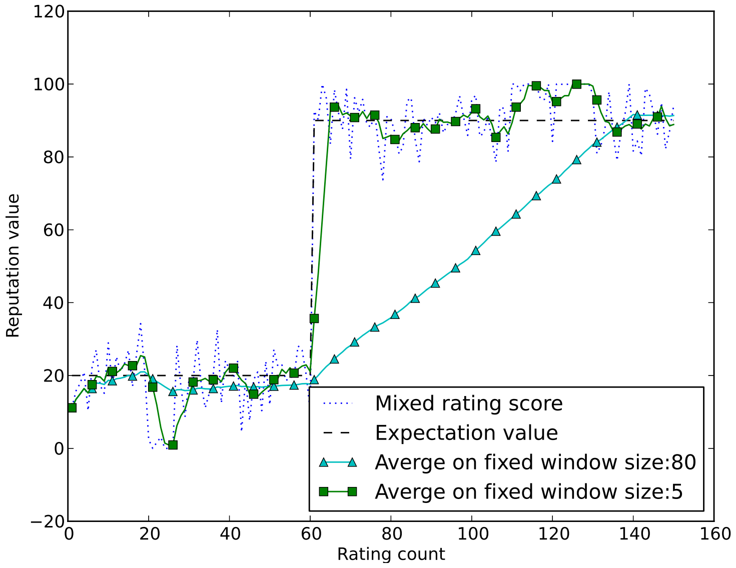


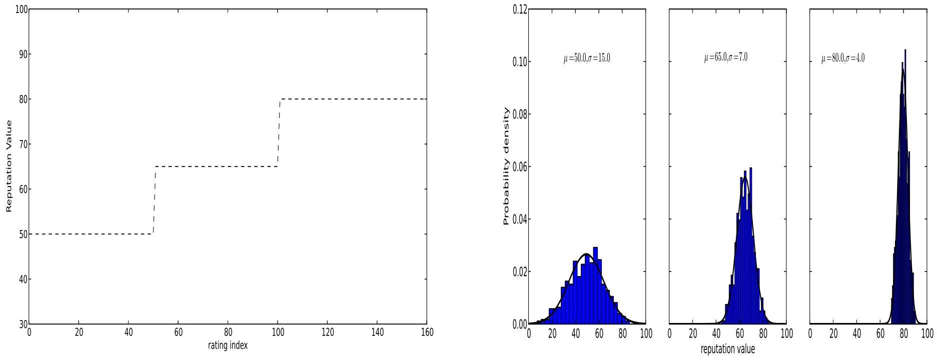
Figure 3.1: The dilemma of fixed sliding window.

is generated by a pairwise function:

$$f(x) = \begin{cases} 20, & 1 \leq x \leq 60 \\ 90, & 60 < x \leq 150. \end{cases} \tag{3.1}$$

to reflect the changes of the service provider. The mixed rating score is the expectation value mixed with fair ratings following normal distribution $N(0, 6.5)$ and unfair ratings; this is treated as the observed rating data. This rating assumption is exactly the same as that of the Olfaction-based Algorithm to Calculate Reputation (OACR) model [Wu et al., 2013], and to realize valid comparisons, exactly the same parameters are adopted in the experiments. In Figure 3.1, a moderate window size of $n = 20$ is taken as an example. As the rating index changes from 50 to 70, the average algorithm needs to weight the ratings from 30 to 50 to reflect the changes in the service provider's behavior.

The above observation inspires us to detect behavior changes and totally drop the influence of prior ratings. It is rational that ratings that exhibit the same behavior can be covered by the same window, which leads us



(a) 1

(b) 2

Figure 3.2: An illustration of the dynamic sliding window concept in service reputation.

to change the window according to the behavior of the service provider. An intuitive view of the concept is illustrated in Figure 3.2, where the reputation of the service changes with time. In detail, the ratings from the beginning to 40 show that the opinions about the service are dispersed. But after the service improvement, the opinions from service consumers exhibit tighter distribution with higher mean $\mu = 4.0$. In addition, previous research has proved the effectiveness of using a variable-size sliding window model for data stream mining of both synthetic and real data [Deypir et al., 2012]. Similar, the window size in our dynamic sliding window model is determined by the behavior changes of the providers. However, a different change detection method is applied because of the difference in data sources.

3.3 Dynamic Sliding Window Model

We have been rather liberal with the usage of the term "sliding window" as strictly speaking it is a sliding window scheme where the window moves in multiple time units to match the behavior of the service provider. Accordingly, we hereafter refer to it as the dynamic sliding window (DSW) and set dynamic window size according to the life-time of the current behavior. Whitby et al. assume that ratings provided by different raters on a given service will follow more or less the same probability distribution. When service provider S_i changes its behavior, it is assumed that all honest raters who interact with S_i will change their ratings accordingly [Whitby et al., 2004]. Our proposed model adopts the same assumption where the rating distribution changes due to changes in the service providers behavior. That is, the fair ratings follow a certain distribution around the reputation value. Here, we adopt a Normal distribution for three reasons:

1. Normal distribution can capture a number of important aspects of service behavior, such as the mean and variability of a service's performance.

2. Normal distribution yields tractable statistical relationships.

3. Dellarocas [Dellarocas, 2000] assumed that fair ratings followed a Normal distribution. Later researchers adopted the same assumption to evaluate reputation models [Wu et al., 2013, Wang et al., 2012].

Figure 3.3 is an illustration of the assumption in the dynamic reputation environment. The distribution of fair ratings (*green dots*) changes with the changes in reputation value (*red dashed line*). Specifically, the mean of the Normal distribution for fair ratings changes with the reputation of the service, while the standard variance is unchanged. In this paper, although a Normal distribution is used to filter out unfair ratings, other distributions can be used if they are able to accurately represent the data distribution.

3.3.1 Basic Dynamic Sliding Window Model

The concept of the dynamic sliding window model leverages the advantages offered by the average algorithm with fixed sliding window; the average algorithm is an optimal algorithm for constant reputation evaluation under the assumption of fair ratings following a Normal distribution. To extend the

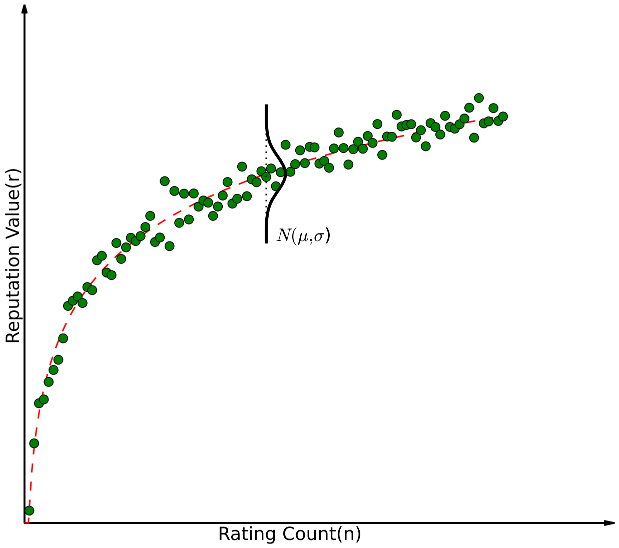


Figure 3.3: An illustration of fair ratings distribution in the dynamic reputation environment.

dynamic sliding window model, the key issues are: 1). Determining the window size based on service provider behavior. The challenge is how to determine if the rating changes result from service quality changes or are merely unfair ratings. We define window size in our model as W; it is the maximum number of ratings in window W following the same distribution with fewer than K consecutive unfair ratings. The window is terminated when the count of newly received unfair ratings exceeds K; subsequent ratings are allocated to a new window. 2). Finding the potential real reputation value function $f(x)$ in a specific window, and then using $f(x)$ to mitigate the influence of unfair ratings. We address the 2nd issue first. If we can detect unfair ratings, it is easy to determine whether we should set a new window or not. The dynamic sliding window derives from the average algorithm, as the average algorithm is preferable to other algorithms when Theorem 1 holds. First we define the **optimal** algorithm to evaluate the reputation of a service if the reputation value $\bar{R}_i$ calculated by it and the real reputation value R_i satisfied: $\lim_{i\to\infty}(\bar{R}_i - R_i) \to 0$ with the number of ratings increased.

Theorem 1. *The average algorithm is optimal for evaluating the reputation value R_n of a service if the reputation value is static and no unfair rating exists.*

Proof. Assume that the real reputation value R_n follows the function $f(i) = C$, C is constant, and the fair ratings X following Normal distribution $N(C, \sigma)$. In this case, the expectation of X is approaches C as n increases: $\lim_{n\to\infty}(\bar{R}_n - C) \to 0$. That is, without observing C, the average algorithm is the optimal algorithm to measure C with two reasons:
1) The expectation of the algorithm converges on the real reputation value C, that is:

$$\begin{aligned} E(X) &= C \\ &\approx \tfrac{1}{n}\textstyle\sum_{i=1}^{n} r_i \end{aligned} \tag{3.2}$$

2) The algorithm assigns equal weight to the received fair ratings. By the average algorithm, we can derive:

$$\begin{aligned} \bar{r}_n &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} r_i \\ &= \tfrac{1}{n} r_n + \tfrac{1}{n}\textstyle\sum_{i=1}^{n-1} r_i \\ &= \tfrac{1}{n} r_n + \tfrac{n-1}{n}\bar{r}_{n-1} \\ &= p \cdot r_n + (1-p)\cdot \bar{r}_{n-1}. \end{aligned} \tag{3.3}$$

□

In equation 3.3, without knowing standard deviation σ, the average algorithm can suppress the noise generated by different raters by assigning the same weight to each received rating. The DSW algorithm uses two steps to generalize the average algorithm in the dynamic reputation environment: 1). Determine the most likely parameter vector $\mathbf{w} =< w_0, w_1, ..., w_M >$ by linear regression using Bayesian theory, where M is the order of the polynomial. Accordingly, $f(i+1) = \sum_{j=0}^{M} w_j \cdot (i+1)^j$ is the most likely value for the $(i+1)$-th rating. As the linear function depends on the sliding window, index i here is defined as the i-th rating in a specific window with parameter $\mathbf{w}$. On the other hand, index n is the n-th rating received by the service provider. 2). Transpose the received ratings in two-dimensions into a constant situation by $f(i) = \sum_{j=0}^{M} w_j \cdot (i)^j$. That is, subtracting $f(i)$ from rating

r_i, coverts the dynamic reputation problem into the static reputation problem. This yields the following theorem:

Theorem 2. *Let σ denote the standard deviation of the span between fair ratings r_i and real reputation value R_i in window W under the dynamic environment. The distribution of $p(r_i - f(i))$ converges on $N(0,\sigma)$ in window W. That is:*

$$\lim_{i\to\infty} p(r_i - f(i)) \to N(0,\sigma) \tag{3.4}$$

Proof. $\because$ For fair rating i in window W, we have:

$$\lim_{i\to\infty} p(r_i - R_i) \to N(0,\sigma)$$

and by Bayesian linear regression, in a clean environment without unfair ratings, the predictive reputation value $f(i)$ generated by Bayesian linear regression will converge on the real reputation value R_i in window W. Because all ratings in the same window, W, follow the same distribution, we get the following limitation:

$$\lim_{i\to\infty} (R_i - f(i)) \to 0$$

Given the function composition $u = R_i - f(i)$, and $h(u) = p(r_i - R_i + u)$, we can get:

$$\lim_{i\to\infty} u \to 0$$

$$\lim_{u\to 0} h(u) \to N(0,\sigma)$$

$\therefore \lim_{u\to 0} h(u) = p(r_i - f(i)) \to N(0,\sigma)$.

□

Based on Theorem 2, without knowing the real reputation value, the weight of the latest received rating i will be determined by standard deviation σ and the expected reputation value $f(i)$. The following formula is used to aggregate the most recent received rating r_n based on the average algorithm 4), assuming that the rating r_n is the i-th rating in current window W_c:

$$R_n = \rho f(n) + (1-\rho)R_{n-1}, and\ f(n) = \sum_{j=0}^{M} w_j \cdot (n)^j. \qquad (3.5)$$

Where ρ is the weight for the predictive reputation value of latest received rating in window W_c. When the behavior of a service changes, R_n will be updated using a new window, therefore, the current window may be updated as the new window. For example, the ratings received by a service provider s are denoted as $r_1, r_2, ...r_n$, and we assume they are covered by the same window W_c. However, ratings r_{n-2}, r_{n-1}, r_n need to be identified as fair or not because their probability deviates a lot, relatively, from the standard deviation. When new rating r_{n+1} is received, and the dynamic sliding window algorithm has determined that it should move to a new window W_n because the service provider has changed its behavior pattern, then all prior pending ratings r_{n-2}, r_{n-1}, r_n will be used as the initial ratings for window W_n, and the algorithm is applied to the new window. Index i is set to 4 in the new window, because it is the 4-th rating received in the new window. Consequently, weight ρ is refreshed based on the new window. Thus, the dynamic changes and reputation calculation can be evaluated in one equation. The scheme for determining the window changes and ρ is addressed in the next subsection.

3.3.2 Parameter Calibration

In the dynamic sliding window model, the key issues are detecting the behavior pattern changes of service providers and identifying the unfair ratings of malicious users. The dynamic property of the model can adapt to the latest behavior of a service. When aggregating the latest received rating r_i, the current ratings distribution is used to determine window size w and weight ρ.

Our proposal uses the Bayesian model to solve the linear regression problem. For a given sequence of ratings $r_1, r_2, ..., r_n$, the Bayesian model can find the maximum likelihood of $\mathbf{w}$ based on the ratings. That is, by maximizing the posterior distribution:

$$p(\mathbf{w}|\mathbf{r},\alpha,\beta) \propto p(\mathbf{r}|\mathbf{w},\beta)p(\mathbf{w}|\alpha). \tag{3.6}$$

parameter **w** can be derived based on the observed ratings **r**, where $\beta = 1/\sigma^2$ is the noise precision parameter and α is the regularization parameter. The Bayesian linear regression approach not just minimizes the sum-of-squares error, but also considers the over fitting problem by applying the regularization parameter to parameters complexity. If the prior distribution of $p(\mathbf{w}|\alpha)$ follows a normal distribution, according to [Bishop, 2006], the log of posterior distribution takes the form of:

$$ln\ p(\mathbf{w}|\mathbf{r}) = -\frac{\beta}{2}\sum_{i=1}^{n}\{r_i - f(i)\}^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} \tag{3.7}$$

Maximizing the posterior distribution with respect to **w** is equivalent to minimizing the sum-of-squares error function plus a parameter regularization term. The last part in the above equation is mainly to limit parameter **w** in order to avoid over-fitting. By partial differentiation on w_j, the parameter can be derived as:

$$w_j = \frac{\partial ln\ p(w_j|\mathbf{r})}{\partial w_j}, j = 0,1,...M \tag{3.8}$$

The hyperparameters α and β can be found by regarding **w** as a latent variable and applying the Expectation-Maximization algorithm [Barber, 2012]:

$$\frac{1}{\alpha} = \frac{1}{M}(Trace(S) + m^T m) \tag{3.9}$$

$$\frac{1}{\beta} = \frac{1}{N}\sum_{n=1}^{N}[y^n - m^T \o(x^n)]^2 + Trace(S\hat{S}) \tag{3.10}$$

where m and S are the mean and covariance of $p(\mathbf{w}|\mathbf{r},\alpha,\beta)$ respectively, and $\hat{S}$ is the empirical covariance of the basis-function vectors $\o(x^n), n = 1,...N$.

When the distribution of ratings changes with time, the parameters can learn from the changes and so suit the latest observations. To detect unfair ratings, we eliminate rating r_n with linear function $f(i)$ in its window, so the span value of ratings $f(1)-r_1, f(2)-r_2, ..., f(n)-r_n$ is converted into the constant reputation situation. The standard variance of $f(1)-r_1, f(2)-r_2, ..., f(n)-r_n$ is calculated and used to determine the occurrence probability, p_{n+1}, of next rating r_{n+1}. If p_{n+1} fails the hypothesis test, the model records index $n+1$ as a new rating that has high probability of indicating the start of a new window. We observe the next few ratings, and if their accumulated probability exceeds the maximum threshold then the ratings from $n+1$ are abnormal in the current window. At this point, those ratings are either unfair ratings or fair ratings, and should be dropped or allocated to another window. Here, we first adopt the assumption that the system security strategy ensures that there is at least one fair rating in K consecutive ratings. That is, when the count of detected abnormal ratings exceeds K, the conclusion can be drawn that the rating has changed because the quality of the service provider was updated. However, we avoid this assumption by introducing the transaction volume variable in next section.

Assuming that the linear function is $f(x) = \sum_{j=0}^{M} w_j \cdot (x)^j$ and $f(i)$ is the expected value of rating i. Using the property of linear regression on ratings, as $i \to \infty$, the error between predicted and the real reputation value R_i approaches 0 in the same window: $\lim_{i \to \infty} |f(i) - R_i| \to 0$. Similar to the averaging algorithm, as the number of accumulated ratings increases, the variation in ratings has less impact on the current reputation value. The weight of the latest received rating is given by this formula:

$$\rho = e^{-|f(i)-r_i|/w} \tag{3.11}$$

where w is the size of the current window, and $f(i)$ is the predicted value of r_i.

3.3.3 Filtering Unfair Ratings based on Hypothesis Test

To make our DSW model robust against malicious attacks, we introduce the hypothesis test and threshold value τ to detect malicious ratings. Let H_0 be the hypothesis that the rating is honest. From above linear regression, the predicted reputation value is $f(n+1)$ for the $(n+1)$-th rating. Given a system without malicious ratings, the span value $f(1)-r_1, f(2)-r_2, ..., f(n)-r_n$ between predicted reputation value and real received rating value should follow a zero-mean normal distribution with standard deviation σ.

To detect a potentially malicious rating, the hypothesis testing evaluates whether the deviation between the rating r_n and the predicted reputation value $f(n)$ is normal enough. Given significance level δ, which determines the confidence level of the test, the problem is to find threshold value τ such that:

$$P(|r_n - f(n)| \geq \tau | H_0) = \delta \tag{3.12}$$

Under hypothesis H_0, $(r_n - f(n))$ follows a zero-mean normal distribution with standard deviation σ. This also yields:

$$P(|r_n - f(n)| \geq \tau | H_0) = 2 * \theta(\tau/\sigma) \tag{3.13}$$

where $\theta(x) = 1 - \Phi(x)$, with $\Phi(x)$ being the cumulative distribution function (CDF) of a zero-mean unit variance normal distribution. Solving (3.12) and (3.13), we can get

$$\tau = \sigma\theta^{-1}(\delta/2) \tag{3.14}$$

If the deviation between the rating and the predicted reputation value exceeds threshold τ, the hypothesis is rejected. Therefore, the rating is flagged as suspicious, and the following ratings will be checked to determine whether they are malicious or not. Instead of using a fixed threshold value, here the threshold τ is dynamic adjusted according to the distribution of the received ratings.

Under the above hypothesis test, we can get the following theorem:

Theorem 3. *For consecutive unfair rating vector $\mathbf{r}_u$ with size K, the probability $p(s)$ of treating $\mathbf{r}_u$ as fair ratings under confidence level δ is:*

$$p(s) = \delta^K \tag{3.15}$$

Proof. If the unfair ratings are independent of each other, then the probability for its occurrence under confidence level δ is δ. According to probability theory, we can derive the probability of K events occurring simultaneously as δ^K.

□

If one specific fair rating r_n fails the hypothesis test, the rating is only tagged as a suspicious unfair rating. The following ratings will be used to determine whether r_n is unfair or not. On the contrary, based on Theorem 3, the probability of the proposed model ignoring a consecutive string of k unfair ratings is δ^k. In the following evaluation section, a service with wide variation in offered quality is simulated to test the performance of our unfair rating detection scheme. In test case 4, the consumers have drastically different ratings on the same service in the same time period. The proposed model can dynamically detect that the fair opinions of consumers heavily diverge from each other. Thus the proposed model will move into a new window with a wide standard deviation.

Finally, as shown in Algorithm 1, the proposed reputation model will run on every service when new ratings are received. First, based on the received rating vector, it updates its parameters in line 5 and predicts the reputation value $\hat{R}$ in line 6. To detect malicious ratings, the model calculates the malicious rating threshold according to the hypothesis test in line 7. If the deviation between r_n and $f(n)$ exceeds the threshold, the rating is marked as pending awaiting future evidence. When the number of pending suspicious unfair ratings exceeds threshold $K = 10$, the model has collected enough evidence and accepts the pending ratings as fair and moves into a new window. Otherwise, if the deviation is less than the threshold, the model determines the ratings are unfair and does not count these ratings.

3.3.4 Rating Ratio Based Dynamic Sliding Window Model

The classic approach to distinguish quality updates from unfair ratings uses a fixed number of consecutive ratings. There are two reasons why we use the rating rate on transaction volume to facilitate this process: 1) Some researchers argue that raters cannot express their opinion accurately if only

```
procedure DSW
    Inputs: r
    Output: $R_n$
    supposing the received rating $r_n$ is in current sliding window
    based on current received rating vector r, run equation (9), and (10), (11)
    predictive reputation value: $f(n) = \sum_{j=0}^{M} w_j \cdot (n)^j$
    compute the malicious feedback threshold $\tau$ using equation 15.
    if $r_n - f(n) > \tau$ then
        push $r_n$ into pending list $l$
        if $sizeof(l) > K$ then
            based on assumption, pending ratings are fair
            move into new window
    else
        if $sizeof(l) > K$ then
            pending rating are unfair and mitigated
        else
            fair rating, accept rating $r_n$ in current window
        calculate weight of $r_n$ by equation 12
        update reputation value by equation 6
```

Algorithm 1: Basic Dynamic Sliding Window Algorithm

numeric rating scores can be assigned [Ramn et al., 2014]. They proposed some methods to facilitate the collection of opinions. Here, the transaction volume of an agent is used to verify the ratings given by raters. For example, if the transaction volume is proportional to ratings, then we have a solid belief that the ratings are fair. However, if the transaction volume is excessive relative to the rating value, then the latest received ratings may be unfair and should be given low weight. 2) Observations of the data of Taobao, Amazon and Hotels.com showed that not all customers have the time or interest to input their ratings.

In fact, nearly 50%∼60% of EBAY customers did not leave a rating for various reasons [Cabral and Horiacçsu, 2010]. Which means transaction volume was much larger than the number of received ratings. Some customers did not express their opinion of the service, but their opinion can be discerned from the transaction volume. Normal customers, those who pay for the service, have little interest in entering their ratings unless they are extremely satisfied or unsatisfied. On the contrary, malicious users tend to seize every opportunity to increase or decrease the reputation of the interacted service deliberately. We try to facilitate the detection process for unfair ratings by examining the ratio of rating number to transaction volume. We denote the improved DSW algorithm as Rating Ratio based Dynamic Sliding Window algorithm (RRDSW). Given service provider s_i, with rating scores from 0 to 60, the ratio $r_r = N_r/T_r$ is around 50%∼60%, where N_r and T_r are the number of ratings received and the transaction volume in a given time span, respectively. After a service update, the difference between the previous ratio r_r and the new ratio $r_{r+1} = N_{r+1}/T_{r+1}$ will stay within a certain range. The reason is that the users cannot change their behavior mode immediately. Based on this observation, we can better identify unfair ratings.

The results in Figure5.2(a) are for the improved DSW algorithm, RRDSW, and show that it has better performance than OACR in all test cases.

3.4 Evaluation

This section presents a series of numerical experiments designed to evaluate accuracy of our model in a comparison with the state-of-the-art competitor [Wu et al., 2013].

3.4.1 Experimental Environment

The best way to evaluate a reputation algorithm is use actual reputation values. The comprehensive testbed proposed in [Irissappane et al., 2012] uses an actual fair rating environment, and the unfair ratings are simulated. It seems impossible to acquire, for research, unfair rating data from real service environments. Moreover, since it is impossible to know the real reputation value of a service, it is hard to create a baseline with which to compare different models. Here we adopt the same evaluation environment as [Wu et al., 2013]. In the paper, the authors use a set of cases and each case has 150 expectation values and 150 rating scores. We adopt the same simulation environment. In that paper, expectation values are generated by expectation functions such as linear, quadratic, sinusoidal, exponential and logarithmic function, while the rating scores are generated by the expectation function plus a median distribution of fair ratings. We use normal distribution $N(\mu, \sigma^2)$ with parameter $\mu = 0, \sigma = 6.5$ in the tests, the noise precision β can be learnt from the observed ratings, and α is predefined as 0.005. To simulate the unfair rating, some of the fair ratings are replaced with unfair rating. In the experiments, the distribution of the unfair rating is restricted as at most 9 consecutive unfair ratings must be followed by 1 fair rating. The test data set is generated by mixing fair ratings with unfair ratings. The mean absolute error (MAE) is calculated to evaluate the accuracy of the algorithm.

$$e = \frac{(\sum_{i=1}^{t} |x_i - E(x_i)|)}{t} \tag{3.16}$$

where t is the total set of ratings, x_i is the $i-th$ rating, and $E(x_i)$ is the expectation value of $i-th$ rating.

To compare robustness, we adopt the classical false/true positive/negative indicators. Specifically, a positive is a malicious reputation feedback which should be rejected by the trust model, and a negative is a normal reputation feedback which should be accepted. The number of positives (resp. negatives) in all feedbacks is n_p (resp. n_n). In general there are four cases: (1) malicious ratings are provided and appropriately mitigated by the reputation system; (2) malicious ratings are ignored by the reputation system; (3) the rating is fair but the reputation system detects it as malicious; and (4) the rating is fair and the reputation system considers it as normal. Cases (1)and(4)represent fair situations. However, cases (2) as false negative and (3) as false positive are failures, which decrease reputation system performance. The number of false positives (resp. false negatives) reported by the reputation model is n_{fp} (resp. n_{fn}). The false positive rate (FPR) is the proportion of all normal ratings that have been wrongly detected, thus

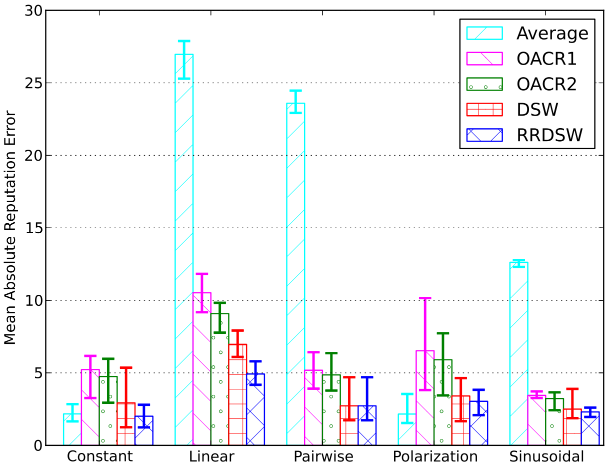


Figure 3.4: Reputation accuracy evaluated on various situations.

$FPR = n_{fp}/n_p$. Similarly, the true positive rate (FNR) is the proportion of malicious ratings that have been ignored, which is $FNR = n_{fn}/n_n$. To detect malicious rating, DSW model uses the significance level δ to decide the confidence of the detection. Normally, a higher significance level will increase both the true and false positive rates. According to many experiments in other tests [Morris, 1976], [Maybeck, 1979], a significance level of 5 percent offers a good compromise between true and false positive rates. Hence, we also set δ as 5 percent in our experiments. This means, when a continuous sequence of unfair ratings K is observed, based on Theorem 3, assuming $K > 3$, the probability of K unfair ratings occurring is lower than $(0.05)^3$ in our experiment.

3.4.2 Design of Test Cases

The proposed dynamic sliding window algorithm (DSW) is implemented and compared with a novel method called the Olfaction-based Algorithm (OACR) [Wu et al., 2013] on various data cases. The proposed model and

OACR are evaluated using the following patterns, which are widely used to reflect the different behaviors of service providers:

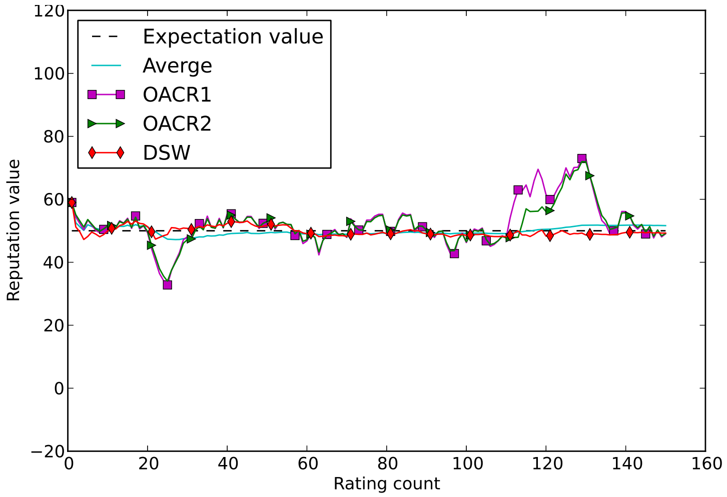


Figure 3.5: Constant Reputation function.

1. **Constant:** This group of service providers either behaves consistently with high reputation value or low reputation value. The providers with high reputation value are rational, while those with low reputation may always provide low quality service to take advantage of the consumer [Zaki and Bouguettaya, 2009, Vogiatzis et al., 2010].

2. **Linear and pairwise:** The service providers change their strategies halfway, either from high reputation value to low in order to abuse their earned reputation [Xiong and Liu., 2004, Wang et al., 2011], or they learn from their previous mistakes and ameliorate their behavior accordingly [Sabater and Sierra., 2001, Zaki and Bouguettaya, 2009, Vogiatzis et al., 2010, Wu et al., 2013].

3. **Sinusoidal:** These service providers perform in a random manner or deliberately increase and then abuse their reputation value periodically. [Vogiatzis et al., 2010] The quality of the service may degrade with time and the service provider updates it periodically.

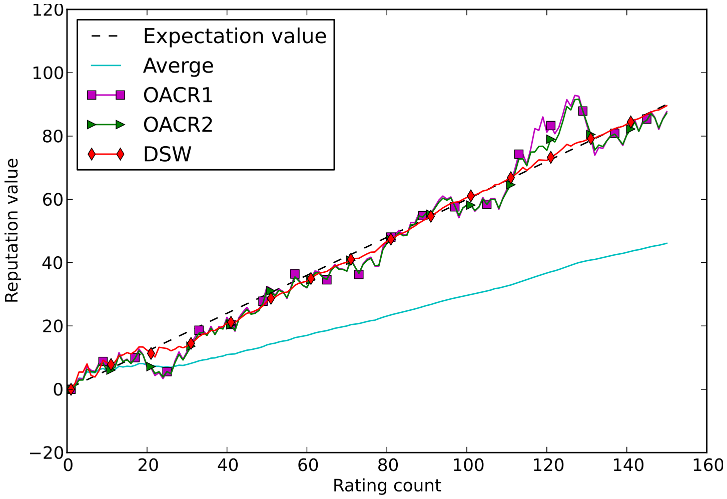


Figure 3.6: Linear reputation function

Even though the above behavior patterns have been mentioned in the literature, they have not been used to establish test cases for experiments. We thus design the following test cases for benchmark tests. Each test case is executed 10 times and the average value is compared in Figure 3.4. In the detailed case figures, we did not draw the RRDSW algorithm line for clarify as it closely tracks DSW.

- Test case 1: $y = 50, 1 \leq x \leq 150$. First, we compare the average algorithm, the OACR algorithm, and our proposal for constant reputation. This test case corresponds with the constant behavior pattern of service providers. Figure 3.5 plots the results of the three algorithms. The results prove that the average algorithm works best in the constant fair rating situation with a small set of unfair ratings.

- Test case 2: $y = 0.6x, 1 \leq x \leq 150$. In this test case, the reputation value of a service provider is keep updated in linear mode. When a service provider becomes aware that more benefit can be derived from the reputation value, it will continue update its service quality. The three algorithms are tested on linear function where y = 0.6x.

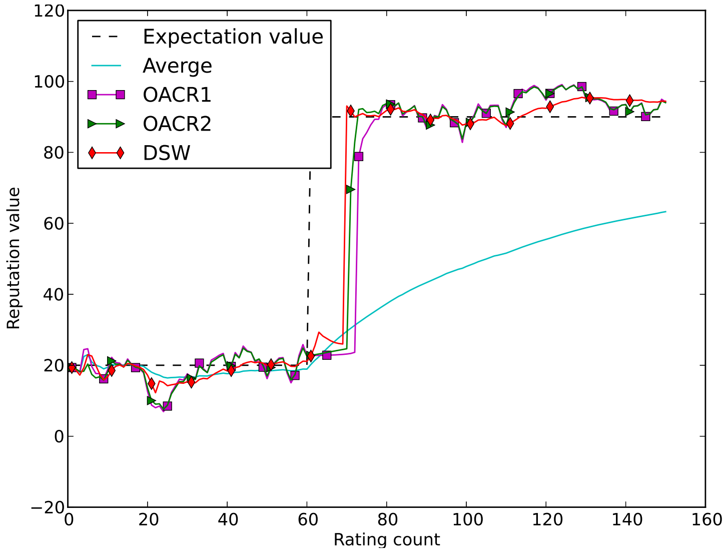


Figure 3.7: Pairwise reputation function

Figure 3.6 shows the results of the three algorithms. We repeat the test for 10 times and plot the results in Figure 3.4.

- Test case 3: Reputation value is changed midway. The expectation function is equation 3.1. This test case demonstrates the behavior pattern that the service providers learn from their experience and update their service quality halfway. The opposite behavior pattern is also examined, in which the service provider abuses their established reputation. The three algorithms are also tested 10 times; the average, maximum and minimum values are plotted in Figure 3.4. Detailed reputation values for each algorithm are illustrated in Figure 3.7.

- Test case 4: Rating distribution changed midway. The expectation function is:

$$y = \begin{cases} 50, \sigma = 6.5, & 1 \leq x \leq 60 \\ 50, \sigma = 18.0, & x > 60. \end{cases} \tag{3.17}$$

Some service providers updated their reputation halfway and the qual-

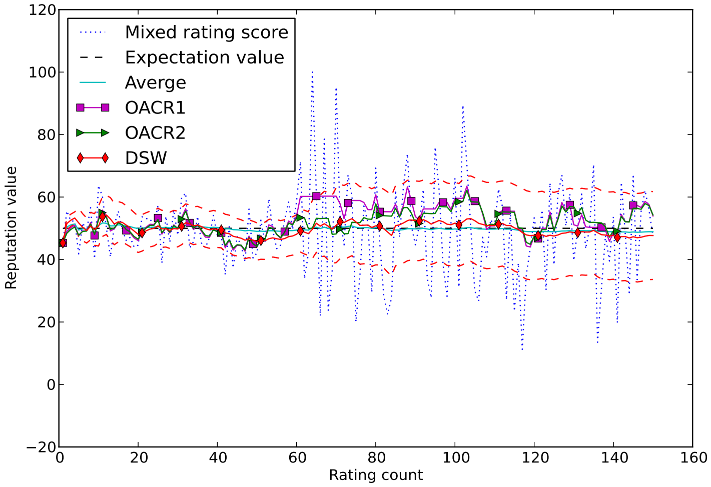


Figure 3.8: Divergence rating distribution.

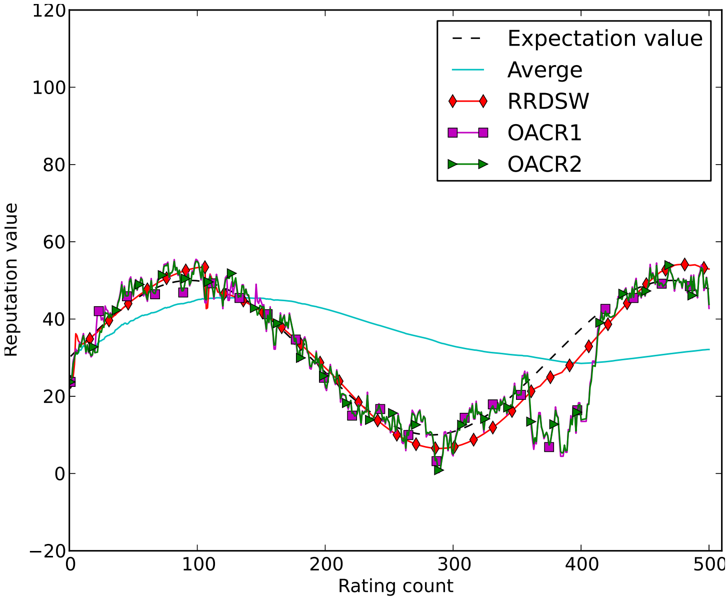


Figure 3.9: Sinusoidal reputation function.

ity of the service is updated, but this updated version is controversial. Large divergence in consumer's opinion yields polarization as regards their ratings. The heterogeneity of the service is simulated as the divergence in consumer opinions. Different consumers might have drastically different fair ratings on the service. Hence, in this test case, the rating variance is changed from 6.5 to 18 midway. The results of the three algorithms are illustrated in Figure 3.8. The σ value for the DSW model is denoted by the red dash line, which verified the proposed model can dynamically adapt its threshold value with different distribution of ratings. In the result, the original ratings are also plotted in blue dashed line. In the figure, the observed rating values fluctuate after rating count 60.

- Test case 5: y=20.0*$sin(x/60.0)$+30.0, 1$\leq x \leq$500. The test case here replicates the sinusoidal behavior of some service providers. The results of the three algorithms on this test case shows in Figure 3.9. We observed that OACR1 adapted better than OACR2 because the weight was changed with rating count in the olfactory phase. Our proposal, DSW, is better than the other algorithms because after it established a new window, and approached the expectation value as more ratings were received.

Overall, the accuracy of our proposal in Figure 3.4 is better than OACR for all test cases. In our last paper [Zhou et al., 2015], the result on the sinusoidal test cases is improved by the full Bayesian linear regression approach because the straight-line version retains a long tail on the turning points.

3.4.3 Performance Evaluation

Robustness

As previously mentioned, the OACR algorithm depends on the assumption that there must be at least one of 10 consecutive ratings must be fair. We lift this constraint by introducing the rating ratio based dynamic sliding window algorithm. We use the pairwise reputation function to confirm robustness as it is widely used in the literature. The scenario is similar to the previous

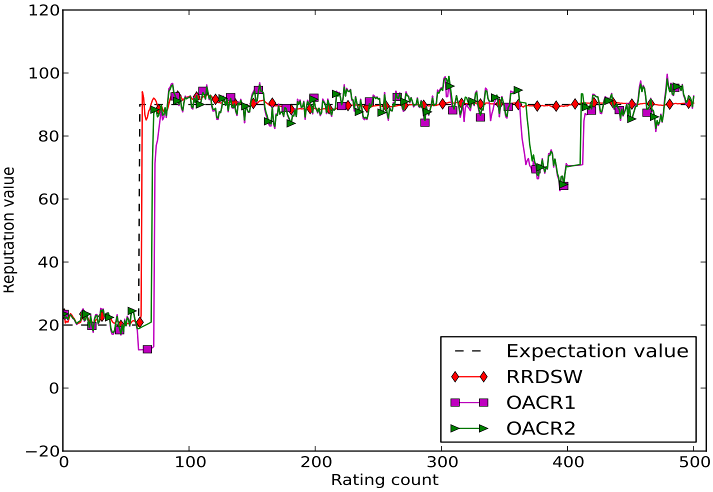


Figure 3.10: Robustness of OACR and DSW.

Table 3.1: Mean absolute error of reputation models under attacks.

	Average	OACR1	OACR2	RRDSW
Constant	**0.80** ± **0.12**	3.85 ± 0.25	3.62 ± 0.21	**0.80** ± **0.14**
Linear	75.42 ± 0.40	4.40 ± 0.25	4.42 ± 0.20	**1.25** ± **0.25**
Pairwise	18.19 ± 0.19	5.61 ± 0.20	5.02 ± 0.26	**1.58** ± **0.23**
Sinusoidal	12.30 ± 0.11	4.05 ± 0.27	3.86 ± 0.24	**2.73** ± **0.24**

tests except that more than 10 consecutive unfair ratings were possible. The result on the pairwise case is shown in Figure 3.10 indicating RRDSW can mitigate the unfair ratings in rating period [360, 400] effectively. While OACR2 can partially mitigate the influence of unfair ratings from 360 to 370, it enters the fading stage after rating index 370, and thereafter misjudges the unfair ratings.

Table 3.1 presents the mean and standard deviation (over 10 tests) for mean absolute error of reputation evaluation on 4 test cases. We find the RRDSW is robust against coalition attacks and outperforms the OACR algorithm by 62% on average.

Based on the false/true positive/negative robustness metrics, we ran the

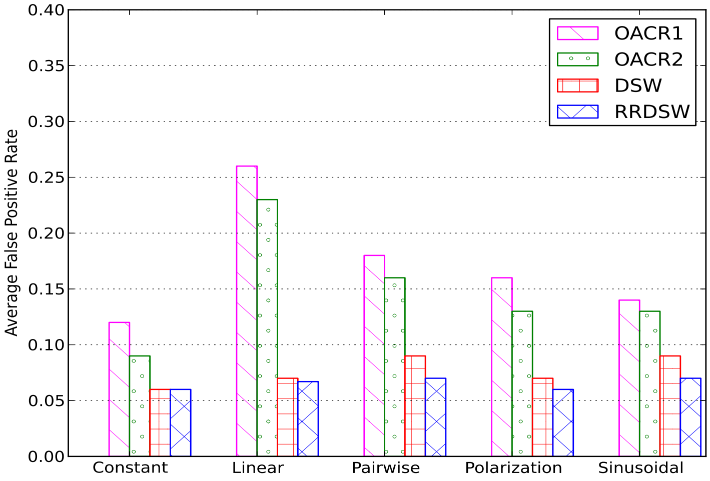


Figure 3.11: Average FPR of different model under pairwise test case.

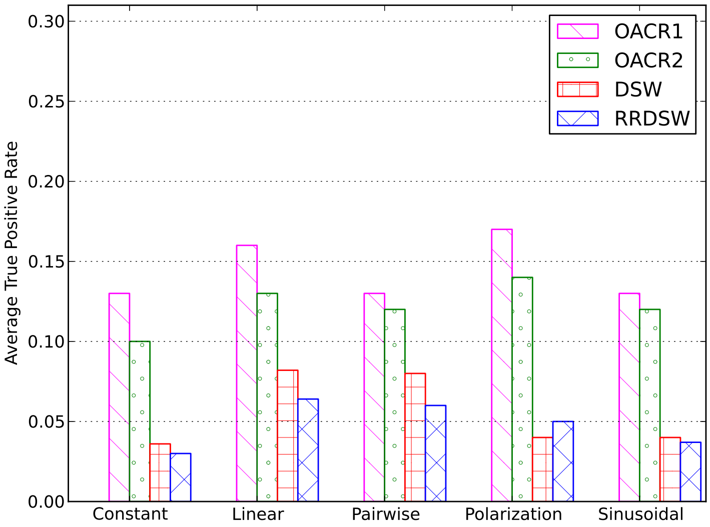


Figure 3.12: Average FNR of different model under pairwise test case

experiments on the test cases and plotted the results Figure 3.11 and Figure 3.12. The figures show that the proposed model outperformed the other models in al test cases in terms of both average false positives rate and average false negatives rate. In the average false positive result, the value for DSW is around 0.05, which means a randomly generated fair rating may have a probability of about $\delta = 0.05$ of being mislabeled as a malicious rating. This result is consistent with the confidence level in the experiment.

Convergence

In order to reduce the algorithm computational complexity from $O(w)$ to constant level, where w is the window size, we introduce the concept of cutting the window length at the point at which DWS becomes stable. The averaging algorithm and the Olfaction-based Algorithm (OACR) [Wu et al., 2013], both OACR1 and OACR2, are evaluated against the dynamic sliding window (DSW) algorithm in terms of mean absolute error metric. Observing a large data set of ratings will allows us to determine whether the reputation system can converge to a fixed mean absolute error or not. And we focus on the constant reputation value scenario, with a normal distribution $N(0,6.5)$ of fair ratings. The result is shown in Figure 3.13.

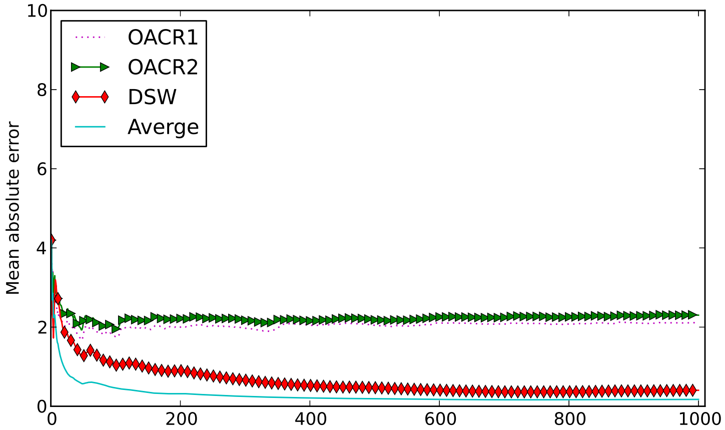


Figure 3.13: Convergence Test on OACR and DSW.

In Figure 3.13, despite the fluctuation in DSW at the beginning, DSW converged quickly to the averaging algorithm in the above environment.

Ideally, the mean absolute error of DSW algorithm and averaging algorithm should approach 0 when the ratings approach infinity. Neither OACR1 nor OACR2 matched this property. The reason is that DSW algorithm uses a linear regression algorithm to detect the real reputation values that underlie the ratings. Thus, as more observations are accumulated, the error between the expectation reputation value and the calculated value approaches 0. OACR allocates a fixed weight to the latest rating, as the rating follows $N(0,6.5)$, and so is not assured of converging to the underlying reputation value. The mean absolute error of reputation values at rating index 1000 are 0.16, 2.27, 2.50 and 0.48 for averaging algorithm, OCAR1, OACR2, and DSW, respectively. The results prove that our proposed dynamic sliding window model, and the averaging algorithm, can accurately converge to the underlying real reputation value.

Extreme Tests

Deviating from the simulation environment of [Wu et al., 2013], we designed additional experiments to test the performance of the models in the worst situation. As the pairwise test case is used most often in the related works, we use the pairwise function to test all models. The mean absolute error is used to evaluate the accuracy of the models given *cm*, the number of consecutive malicious ratings. For *cm*, values of 10, 20, 30, and 40, the results are plotted in Figure 3.14. Because the OACR model assumes $K = 10$ for detecting the unfair ratings, the results show that its performance degrades as *cm* is increased. RRDSW, on the other hand, uses the rating ratio and hypothesis test to detect unfair ratings, so even though it cannot detect all unfair ratings, it shows a considerable improvement in performance.

3.4.4 Analysis and Discussion

The results shown in Figure 3.4, show the proposed basic DSW model outperforms all OACR variants in almost all test cases. Figures from 3.5 to 3.9 clearly show that the plots of DSW are closer to the expected values than

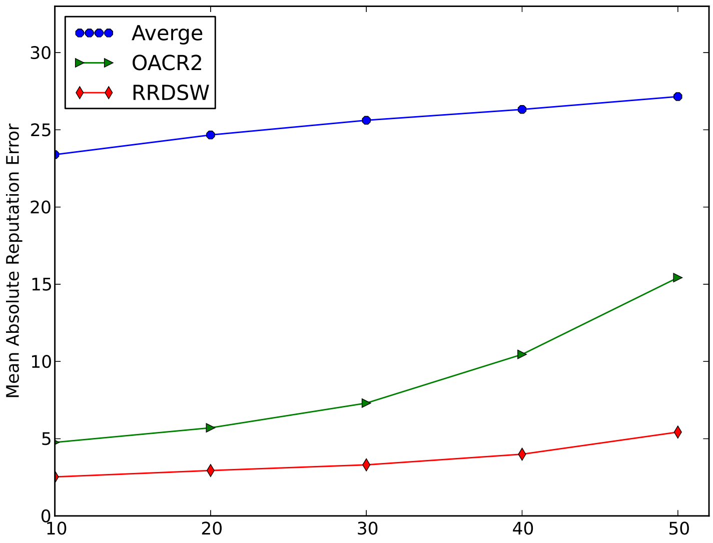


Figure 3.14: Extreme testing on different models.

the OACR variants. This is because the DSW algorithm is based on the following theoretical advantages:

1. Bayesian linear regression. The Bayesian theory offers a sound probability foundation for choosing parameters $\mathbf{w}$. By maximizing the posterior distribution with respect to $\mathbf{w}$ on fair ratings X will yield a plausible value for $\mathbf{w}$ [Bishop, 2006].

2. Probability theory. After subtracting the fair ratings with their expected rating calculated by linear function $f(n)$, the variance σ of the data is calculated. For newly arrived rating r_n, if its occurrence probability less than $\delta = 0.05$, DSW pushes it into suspicious list and continues to accumulate the following ratings. If the accumulated occurrence probability P_a exceeds δ^K, which indicates small probability events in the same direction (all the ratings below or above 0), DSW begins to mitigate the unfair ratings. Probability P_a is changed with the distribution of ratings. This process is different from the OACR algorithm

in [Wu et al., 2013], in which the author uses a predefined threshold value for detecting unfair ratings, and so is robust against dynamic changes in the environment.

However, the basic DSW is inferior to the OACR variants in the sinusoidal test case. In Figure 3.9, DSW assumes the service provider is in its current behavior pattern when the algorithm cannot distinguish the changes by service changes made by unfair ratings. As mentioned in Section IV, in order to distinguish the changes caused by service updating from those by unfair ratings, a constant parameter, K, is introduced in our model as well as OACR [Wu et al., 2013]. In the K ratings, DSW assumes the latest received ratings are unfair and eliminates the impact of this suspicious period. Therefore, in Figure 3.4, the rating ratio based dynamic sliding window algorithm can improve the result to equal the performance of the OACR algorithm in the worst case. However, DSW has the advantages of convergence to the real reputation value and robustness against coalition attacks.

Test case 4 widened the rating variation after rating index 60. The distribution developed from the initial 60 ratings does not suit the subsequent ratings. When the proposed model detected that the first rating after rating index 60 failed the hypothesis test, it tagged the rating as a suspicious unfair rating and added it to the pending list. If all successive K ratings fail the hypothesis test and they have the same positive or negative opinion, the model determines that the suspicious ratings in the pending list are fair and move into a new window for reputation evaluation. Because according to Theorem 3, the probability is only δ^K, there is no reason to assume this change is caused by unfair ratings. Otherwise, the ratings in the pending list are unfair and their influence will be mitigated by our model. This is the situation occurred in test case 4. For a suspicious rating, the following rating pass the hypothesis test, which give evidence on identifying prior pending rating.

Although various reputation systems are proposed in the literature, no reputation system can resist all unfair rating attacks. Some reputation systems employ machine learning algorithms, but unfair raters can also learn from the system and launch carefully tailored unfair ratings. In this situation, it is really hard to tell if the service reputation change is due to unfair ratings or some problem with the service, because they have the same pattern. Our Rating-Ratio based model applies a supplementary factor, the rating ratio, to distinguish the unfair ratings. If the malicious raters provide unfair ratings at a relatively low rating ratio, this method will fail. However, our proposal dramatically raises the cost of malicious attacks and thus reduces the incentive to launch unfair rating attacks as discussed in [Zhou and Matsubara, 2015].

3.5 Conclusions

This paper tackled the time lag and unfair rating problems in reputation systems by introducing a new algorithm. In the proposed algorithm, the distribution of ratings is continuously monitored to dynamically resize the sliding window. When the service provider changes its behavior, the algorithm sets a new window to remove the influence of previous behavior such that the latest reputation reflects the latest quality of the service provider. The Bayesian Linear Regression approach is used to aggregate received ratings and detect reputation changes even in service environment that exhibit violent fluctuations in quality. In order to facilitate the detection of unfair ratings, we improve the basic dynamic sliding window based on the observation that 50%∼60% of consumers fail to rate their transactions. This mechanism lifts the restrictive assumption made by the existing algorithm that a fixed number of consecutive ratings must be observed before unfair ratings can be identified.

Simulations showed that our algorithm was more accurate than published algorithms and a method used by current commercial services The proposed algorithm adapts itself to dynamic changes on the rating distribution unlike the existing algorithm that uses a fixed threshold for unfair rating detection. Furthermore, by introducing the ratio of rating number to transaction volume as an indicator, the improved algorithm outperformed the compared algorithm by 45% on average in relieving the time lag problem.

Chapter 4

Building robust reputation under coordinated unfair rating attacks

Service computing is playing a more and more important role in current Internet activities, especially with the rapid adoption of electric markets, more and more individuals are engaging with commercial services. As the potential profit of service computing is becoming clear, malicious users are ramping up unfair rating attacks that can mislead honest service consumers into transacting with dishonest service providers. Moreover, some dishonest service providers may collude with dishonest service consumers to damage the reputation of service rivals.

4.1 Introduction

With the rapid adoption of electric markets such as eBay, Amazon and Taobao has throw into strong relief two major problems: 1). Even though large number of service consumers have no or little past experience with the services, each service consumer must attempt assess and identify reliable interaction partners by themselves. 2). Unfair rating attacks from dishonest service consumers can mislead honest service consumers to transact with dishonest service providers. Moreover, some dishonest service providers behave differently toward different consumers, and can collude with dishonest service consumers to boost their reputation in the e-commerce system. These problems are currently countered, to mitigate the potential risk to the consumers, by adopting some form of reputation system [Huhns and Singh, 2005, Jøsang et al., 2007] or reputation-based recommender systems [Xiong and Liu, 2004, Xu et al., 2007, Goto et al., 2011, Qiu et al., 2013]. Such systems not only aggregate and filter information for service consumers, but also act as an incentive for service providers to improve their service quality. Reputation is defined as a subjective assessment of service quality and is typically determined by collecting ratings or feedback from service consumers. It acts as a global and public value that can be observed by all the participators in the system. Hence, new comers who have no experience can utilize the reputation system to mitigate the risks of selecting a partner.

Various approaches [Wang et al., 2011, Liu et al., 2011, Jiang et al., 2013] have been proposed for building robust reputation systems for service providers, and most use personalized similarity-based credibility to evaluate the reputation of a service provider. However, those techniques are usually unreliable when the rating distributions are marginally effective. In which situation not only are novices exposed to significant risks, but also experts are unable to exploit the rating information efficiently even if they accumulate a lot of data. Others models [Jøsang and Ismail, 2002, Teacy et al., 2006, Wang et al., 2012] use statistical theory to handle unfair ratings; they are designed to filter out the ratings that deviate in some way from the mainstream ratings. TRAVOS, however, evaluates rating accuracy against the consumers past opinions, and hence can avoid the Sybil attack. These methods, however, cannot respond to the dynamic changes possible in service quality, which fails in providing a timely accurate reputation of the service.

To overcome the drawbacks of the previous methods, we take an approach of employing a clustering method. The clustering methods have already been studied and proven to be effective in immunizing reputation systems against unfair ratings [Dellarocas, 2000]. However, different from previous proposals that cluster the rater based on similarity among raters, the model proposed in this paper uses the rating ratio information to detect unfair raters. This information reflects the inherent behavior of customers; 50%∼60% of eBay customers do not leave ratings for various reasons [Cabral and Horiacọsu, 2010]. If customers leave ratings for almost all transactions, it is likely that they are conducting a fraud. Further, we update the trustworthiness of each rater by applying a sanctioning function. Service provider reputations are aggregated by using a Dirichlet distribution. Based on this model, we analyze the costs and profits associated with effective attacks and reveal the conditions under which such attacks become attractive. As discussed in [Khosravifar et al., 2010b], our reputation system can not mitigate the unfair attacks completely but reduce the incentive for unfair rater to perform illegally.

Actually, in a system where service providers can collude with consumers, there is no way to remove the impact of unfair rating completely if the malicious consumers can rapidly modify their attack strategies. An interesting solution is to analyze the cost of performing an effective unfair rating attack against a specific reputation model and negate the incentive that drives unfair rating attacks.

4.2 The clustering-based reputation model

The key point of a reputation model is how to detect the dishonest service raters and decrease their trustworthiness when calculating the reputation of a service provider.

4.2.1 Clustering the rating vector

We assume a service computing environment with M service providers $S = \{s_i | i = 1,2,...,M\}$ each having one functional equivalently service with different quality, and N service consumers $C = \{c_j | j = 1,2,...,N\}$. The rating given by consumer c_j to provider s_i after the transaction at time t denoted as $r_{t,c_j,s_i} \in [0,1]$. The perfectly satisfied service is expressed with rating value 1, otherwise given 0 extremely. As the ratings accumulate, rating vector

R_{t,s_i} received in time period t by s_i can be expressed as:

$$\overrightarrow{R}_t^{s_i} = [r_{t,c_1,s_i}, ..., r_{t,c_j,s_i}, ...] \tag{4.1}$$

The number of element in $\overrightarrow{R}_t^{s_i}$ is no more than $t \cdot N$, because not all the consumers will transact with s_i in time t and not all consumers will leave their ratings after the transaction. And each consumer can use the service at most one time at a time. The rating ratio of consumer c_j is given by:

$$\rho(t)_{c_j} = \gamma_t^{c_j} / \eta_t^{c_j} \tag{4.2}$$

where $\gamma_t^{c_j}$ is the number of rating for c_j, and $\eta_t^{c_j}$ is the number of total transaction for c_j. Rating vector $\overrightarrow{R}_t^{s_i}$ is used adopted by the clustering algorithm when classifying the ratings into Z clusters $T_1, T2, ..., T_Z$. A fast and robust cluster algorithm is applied on the rating vector to generate a set of clusters [Rodriguez and Laio, 2014]. The advantage of this cluster algorithm is that it can classify the clusters without choosing the appropriate threshold to discard the noise points when compared with approaches based on the local density of data points [Ester et al., 1996]. For large data sets, the clustering algorithm can find the density peaks that are robust with respect to the choice of cutoff distance between points. Density peaks are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. Take figure 4.1 as an example, in day 90, two clusters occur simultaneously. In this situation, an additional mechanism is needed to detect the dishonest cluster.

4.2.2 Detecting the dishonest clusters

As we have derived Z clusters $T_1, T_2, ..., T_Z$, in each cluster, the set of ratings are denoted as $T_k = \{c_j | c_j \in C\}$. For each consumer c_j, if his rating on s_i at time t is classified into the honest cluster, the trustworthiness value $\phi(t)_{c_j}^{s_i}$ of c_j on s_i will be updated by the sanctioning function $h(t, c_j, s_i)$. The trustworthiness value of fair raters increases and carries a high weight when calculating the reputation of s_i, while unfair raters are deweighted. To

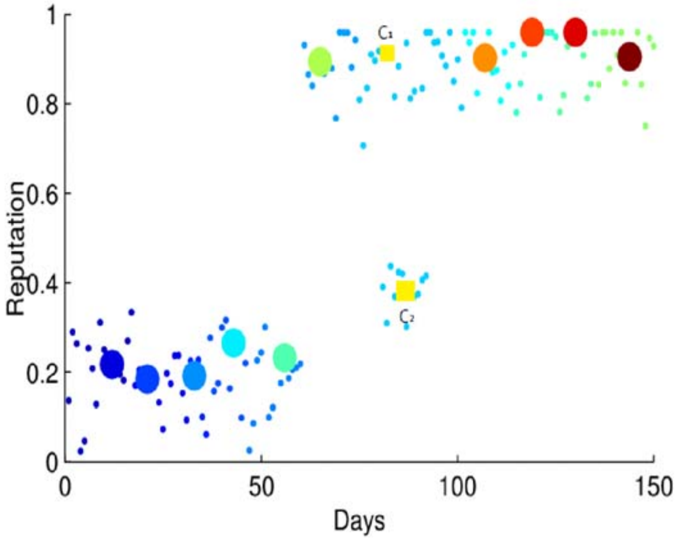


Figure 4.1: An example of clustering.

detect dishonest clusters, there must be some property that can distinguish unfair from fair raters. The underlying statistics of eBay usage showed that normal customers, c_j , exhibit a rating ratio, $\rho(t)_{c_j}$, around 0.4∼0.5. The perpetrators of each unfair rating attack naturally attempt to minimize their cost in performing the attack and they do so by increasing their rating ratios. As a result, the rating ratio ρ can be used to detect the unfair raters. While for honest raters, usually they can gain high trustworthiness in the reputation system. Hence, a ρ against ϕ graph can be plotted for rating vector $\overrightarrow{R}^{s_i}_t$. The graph is plotted with rating ratio ρ as its x axis and trustworthiness as its y axis. Hence, for Sybil attack, the unfair consumers will be categorized as high ρ, low ϕ group and detected as such. The reason is that the unfair raters will seize every opportunity to enter malicious ratings. That is, for every transaction, they will give an unfair rating to decrease or increase the reputation of the provider deliberately. However, for Sybil Camouflage attack, the unfair raters must secure a relative high degree of trust ϕ and high ρ because they initially pretend to be fair raters. Rating ratio ρ plays a key role in distinguishing the dishonest clusters. We define rating ratio for a cluster T_k as:

$$\rho(t)_{T_k} = \frac{1}{K}\sum_{j=1}^{K}\rho(t)_{C_j}, and\ C_j \in T_k \tag{4.3}$$

where K is the number of raters in cluster T_k. The above equation reveals that the rating ratio of a cluster is the average rating ratio of all its members. For example, in figure 4.1, the rating ratios of cluster C_1 and cluster C_2 are calculated to detect the dishonest cluster.

4.2.3 Calculating the reputation based on honest cluster

As each honest rater c_j has their direct experience with the quality of s_i, variable $\phi(t)_{c_j}^{S_i}$ indicates how much trust should be ascribed to rating r_{t,c_j,s_i}. Suppose the number of consumers in honest cluster T_k is K, the current reputation value can be any of those K ratings. Each rating is tagged with the trustworthiness parameter to determine the contribution they made to the service provider. In the case of discrete distributions parameterized by $\phi(t)_{c_j}^{s_i}$, the weight of rating from c_j is usually calculated with a K-dimensional Dirichlet distribution [Teacy et al., 2012]:

$$Dir(C_l|\alpha) = \frac{1}{Beta(\alpha)}\prod_{l=1}^{K}(c_l)^{\alpha_l} \tag{4.4}$$

The $Beta(\alpha)$ is defined in terms of the gamma function as:

$$Beta(\alpha) = \frac{\prod_{l=1}^{K}\Gamma(\alpha_l)}{\Gamma(\Sigma_{l=1}^{K}\alpha_l)} \tag{4.5}$$

where $\alpha =< \alpha_1, \alpha_2, ..., \alpha_K >$, and $\alpha_l = \phi(t)_{C_l}^{s_i}$, C_l is the l-th member in cluster T_k. Given that the expected value of the Dirichlet distribution is defined as $E[c_l|\alpha] = \alpha_l / \sum_{l=1}^{K} \alpha_l$. The reputation of s_i can be derived as:

$$\begin{aligned}
\mathbf{R}_{t,s_i} = E[r_{t,c_j,s_i}|\alpha] &= \sum_{l=1}^{K} r_{t,c_l,s_i} \cdot p(c_j = c_l|\alpha) \\
&= \sum_{l=1}^{K} r_{t,c_l,s_i} \cdot E[C_l|\alpha] \qquad (4.6) \\
&= \sum_{l=1}^{K} r_{t,c_l,s_i} \cdot \frac{\alpha_l}{\sum_{n=1}^{K} \alpha_n}
\end{aligned}$$

where r_{t,c_l,s_i} is the rating value here.

4.2.4 Sanctioning function

The sanctioning function acts here as a sanction mechanism to reward the honest rater, while devaluing the ratings of dishonest raters. Given this background, we define the admissible function as: A sanctioning function is said to be **admissible** if it always rewards the honest rater and punishes the dishonest rater at time t. That is, for honest rater C_h and dishonest rater C_d at time t, the following equations hold for the admissible function $h(t, c_j, s_i)$:

$$\begin{aligned}
\phi(t)^{s_i}_{c_h} &= h(t, c_h, s_i) \\
&> h(t-1, c_h, s_i) = \phi(t-1)^{S_i}_{c_h} \qquad (4.7) \\
\phi(t)^{s_i}_{c_d} &= h(t, c_d, c_i) \\
&< h(t-1, c_d, s_i) = \phi(t-1)^{S_i}_{c_d} \qquad (4.8)
\end{aligned}$$

A sanctioning function sensitive to dishonest raters can effectively mitigate their impact. However, an honest rater may be erroneously devalued especially when the service requests are stacked. For simple balance, the sanctioning function adopted in this paper uses the admissible function defined as:

$$h(t, c_h, s_i) = h(t-1, c_h, s_i) + 1 \qquad (4.9)$$

$$h(t, c_d, s_i) = max(h(t-1, c_d, s_i) - 1, 0) \qquad (4.10)$$

where $h(t, c_h, s_i) >= 0, h(t, c_d, s_i) > 0$ and $h(0, c_h, s_i) = 0, h(0, c_d, s_i) = 0$ in equation 4.9 and 4.10 . For honest and dishonest raters, their trustworthiness values are updated by equation 4.9 and 4.10 respectively. Assuming that by calculating the rating ratio and trustworthiness of cluster C_1 and C_2, the model classify C_1 as the honest cluster. And the reputation value on time t is aggregated based on the honest cluster C_1. The trustworthiness of each rater in both clusters will be updated by equation 4.9 and 4.10. That is, the weight for raters in cluster C_1 will be added while lighten the trustworthiness for raters in the dishonest cluster C_2.

The sanctioning function can be flexibly replaced to reflect the inherent features of service provider. For example, the sanctioning function could be designed to punish heavily on those camouflaged consumers who pretend to be honest by giving favorable ratings. Those camouflaged consumers can be detected and punished by the sanctioning function when the quality of service is upgraded or degraded. For those who keep grade fair ratings will receive more reward from the sanctioning function. Also, the sanctioning function can be replaced to ones that reflects the inherent feature of consumers. Take the variance of the ratings given by a specific consumer for example. If the variance of ratings from C_i on service S_A is smaller than the variance of C_j on the same service, then C_i seems more trustworthy because the ratings given by C_i is more consistent with the underlying reputation value. And assume that consumer C_i often gives overly negative ratings, C_j often gives overly positive ratings, C_k often gives neutral ratings. In this situation, although consumer C_i and C_j may show lower variance than C_k, but C_k is more trustworthy as it has a large chance to reflect the real reputation value.

When service provider s_i receives the rating vector, the Clustering-based Reputation Model (CRM) procedure will respond by reflecting the latest reputation of s_i. It first categorizes the ratings into several clusters, and detects unfair clusters by their rating ratios. This clustering approach makes CRM preferable with large data set as its computational complexity is only sensitive to the number of recently received ratings. The threshold value of $\rho(t)_{T_k}$ delineating honest from dishonest clusters can be learnt by a supervised machine learning algorithm and thus updated over time. The detailed mechanism is illustrated in later section. The rest of the code in Algorithm 2 is straightforward; rater trustworthiness is updated and the final reputation value is derived.

4.2.5 Parameter Calibration

By Algorithm 2, the key parameter is the rating ratio value ρ and trustworthiness ϕ to delineate the honest service consumers from dishonest. We use the rating ratio of a cluster to detect the unfair ratings. Supposing we have get Z clusters $T_1, T_2, ..., T_Z$, and the rating ratio for each cluster at time t is denoted as $\rho(t)_{T_1}, \rho(t)_{T_2}, ..., \rho(t)_{T_Z}$. To get the optimal discrimination, we use the linear supported vector machine model (SVM) to obtain the optimal separator value that divide the clusters into two groups: fair clusters and unfair clusters. For example, in the ρ-ϕ graph, the division line derived by SVM can distinguish the two type of clusters.

For training data $\mathscr{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i = (\rho(t)_i, \phi(t)_i), y_i \in \{-1, 1\}\}_{i=1}^{Z}$, the hyperplane that separate the clusters can be written as:

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \tag{4.11}$$

where $\cdot$ denotes the dot product and $\mathbf{w}$ the normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$. For clusters are fair, the value $\mathbf{w} \cdot \mathbf{x} - b = 1$, while $\mathbf{w} \cdot \mathbf{x} - b = -1$ otherwise. For clusters that have different distribution, the hyperplane is dynamic changed according to the ratings received. The dynamic property of the hyperplane can prevent the attacks from unfair users as it is hard to guess what is the current threshold value.

The pseudo-code summary of the clustering-based reputation model is given in Algorithm 2.

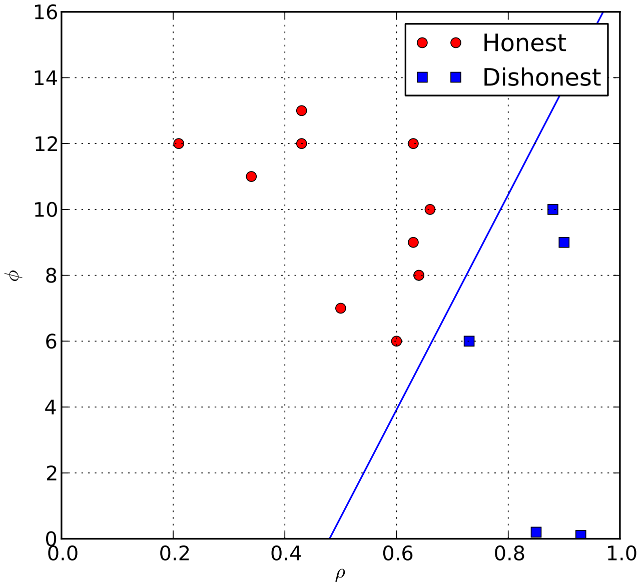


Figure 4.2: An example of SVM on clusters. Each dot denotes one rater.

```
procedure CRM($s_i$, $\overrightarrow{R}_t^{s_i}$)
    Inputs: $s_i$, evaluated service provider;
            $\overrightarrow{R}_t^{s_i}$, rating vector received by $s_i$ at time $t$;
    Output: $\mathbf{R}_{t,s_i}$, the reputation of $s_i$ at time $t$.
    $T_1, T_2, ..., T_Z$ = CLUSTERING($\overrightarrow{R}_t^{s_i}$)
    $\forall T_k, (1 \leq k \leq Z)$
        Calculate $\rho(t)_{T_k}$ by equation (3)
    Tag cluster $T_k$ as dishonest using equation (11):
    $\forall T_k, (1 \leq k \leq Z)$
        Update trustworthiness by sanctioning function (9) and (10)
    $\forall$ honest clusters
        Calculate the reputation $\mathbf{R}_{t,s_i}$ by equation (6)
    return $\mathbf{R}_{t,s_i}$
```

Algorithm 2: Clustering-based reputation model

Initially, the trustworthiness of the consumer are always from 0, and the only factor we can use is the rating ratio ρ. But, at the very beginning, ρ may fluctuate markedly with few transactions. The initial value of ρ is assigned to 1 to aggregate all the ratings to learn the parameters. When $\rho = 1$, the SVM algorithm is equivalent to the average algorithm widely used in commercial reputation system. The average algorithm simply calculate the mean of all received ratings. The above mechanism will be executed when the model has learnt the trustworthiness of raters. At the first run of the system, the system always assume the raters are honest, as it has no evidence to distinguish the dishonest from honest ones. After the model get the different distribution of ρ, the SVM model can use this parameter for a one dimensional division. And eventually, when the trustworthiness of the clusters can be derived from transaction, the SVM model will divide the clusters with two dimensional.

4.3 Experimentation

To evaluate the proposed model, we first introduce the duopoly service providers testbed used in paper [Jiang et al., 2013]. Different attack strategies will be simulated to assess the robustness and accuracy of the proposed model CRM relative to the Multi-agent Evolutionary Trust model (MET) as the paper concludes the model is more robust and effective than the state-of-art models against typical attacks. MET first builds its network by selecting the agents with a fitness function from one generation to another, the network is updated when new ratings are available. The reputation value is evaluated by aggregating the opinion from all agents in its network.

4.3.1 Simulation Setup

As the papers on different reputation models used their own evaluation method even a comprehensive testbed is proposed in [Irissappane et al., 2012]. But the data in the testbed is lack of rating ratio information. Hence, we reuse the e-market testbed designed for simulating ”Duopoly Market” where two service providers occupy a large proportion of the transaction volume [Jiang et al., 2013]. The dishonest duopoly provider may collude with the dishonest consumers to perform various attack to the damage the reputation of honest provider. The setting of the testbed here follows that in [Jiang et al., 2013]. The simulation assumes that of the 198 common service providers, half are honest and the other half are dishonest. Furthermore, for the non-Sybil based attack case, there are 12 dishonest consumers (attackers) and 28 honest consumers. The number is switched in the Sybil attack case, that is, 28 attackers and 12 honest consumers. When the dishonest consumers perform the Camouflage attack, all attackers pretend to be honest in the first 20 days to increase their trustworthiness and rate unfair rating afterward. Each consumer interacts with one provider each day, assume the consumer has the probability of 0.5 of interacting with a common service provider. When choosing which common provider to access for service, the honest consumer tends to select the provider randomly. In the duopoly case, the honest consumer uses the reputation model to decide which one should be accessed. The attacker will choose the duopoly service provider according to the attack modes. After each transaction, each consumer rates the service provider with probability P_r, which is the willingness to give their rating. The rating scores given by honest consumers following Normal distribution $N(\mu, \sigma)$, where μ is the reputation of a service and $\sigma = 0.05$. For fairness, we use the setting as that in MET [Jiang et al., 2013] except giving each consumer a probability to rate the service. In MET, the number of dishonest buyers are occupied 30% of the total buyers. The reason is that trust models are most effective when only 30% of buyers are dishonest [Whitby et al., 2004]. However, we would like to discuss how the ratio of honest and dishonest raters will affect the performance of different models. For CRM, its performance depends on how much the clustering algorithm can detect the honest raters, if it distinguishes the honest clusters on 100% success, then CRM can give plausible reputation value even if only one honest rater exist. MET uses personalized similarity-based credibility to evaluate the reputation value. If such similarity can not be established because of lack of similar peers.

Table 4.1: Mean Absolute Error (MAE) of Reputation Estimation for Honest Duopoly Service on Static Reputation.

Models	Constant	Camouflage	Whitewashing
MET	0.014±0.005	0.013±0.005	0.014±0.004
CRM	**0.009±0.002**	**0.008±0.002**	**0.009±0.001**
Models	Sybil	Sybil Cam*	Sybil WW*
MET	0.027±0.018	0.027±0.009	0.027±0.009
CRM	**0.014±0.003**	**0.012±0.003**	**0.014±0.003**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

Table 4.2: Mean Absolute Error (MAE) of Reputation Estimation for Dishonest Duopoly Service on Static Reputation.

Models	Constant	Camouflage	Whitewashing
MET	0.057±0.018	0.054±0.018	0.052±0.015
CRM	**0.030±0.025**	**0.038±0.026**	**0.029±0.021**
Models	Sybil	Sybil Cam*	Sybil WW*
MET	0.087±0.030	0.088±0.032	0.090±0.029
CRM	**0.030±0.025**	**0.034±0.026**	**0.032±0.021**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

The robustness of reputation model(M) against attack model (Atk) is defined as:

$$\mathscr{R}(M,Atk) = \frac{Tran(s_H)}{c_H \times Days \times Ratio} \tag{4.12}$$

where $Tran(s_H)$ is the transaction volume of the honest duopoly provider by honest consumers, c_H is the number of honest consumers, and $Ratio$ is the dominance ratio and assigned 0.5 in this paper. The value of $\mathscr{R}(M,Atk)$ normally is in [0, 1], where 0 indicates the model is completely vulnerable to attack type Atk; while 1 denotes the model is completely proof against

Table 4.3: Robustness of Reputation Models under Different Attack Models on Static Reputation.

Models	Constant	Camouflage	Whitewashing
MET	0.960±0.025	**0.966±0.020**	0.966±0.021
CRM	**0.995±0.019**	**0.966±0.020**	**0.994±0.019**
Models	Sybil	Sybil Cam*	Sybil WW*
MET	0.926±0.039	0.920±0.037	0.934±0.039
CRM	**0.979±0.030**	**0.995±0.033**	**0.990±0.032**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

the attack. The accuracy of the model is evaluated by mean absolute error (MAE):

$$MAE(s_i) = \frac{\sum_t |'R_{t,s_i} - \mathbf{R}_{t,s_i}|}{Days} \tag{4.13}$$

where $'R_{t,s_i}$ is the actual reputation value of s_i, and $\mathbf{R}_{t,s_i}$ is the reputation as estimated by the reputation model. For honest consumers, their opinion can reflect the actual reputation value. As in the MET model, the reputation value of s_i is calculated from the ratings of all honest consumers, hence, the above equation can be transformed into:

$$MAE(s_i) = \frac{\sum_t \sum_{c_j} |'R_{t,s_i} - \mathbf{R}_{t,c_j,s_i}|}{c_H \times Days} \tag{4.14}$$

where $\mathbf{R}_{t,c_j,s_i}$ is the reputation as estimated by the consumer c_j. Small MAE values indicate that the model is more accurate.

Each attack is carried out 50 times to reduce the randomness. The mean and standard deviation values are shown in Table 4.1 and 4.2, and the best results are in bold font.

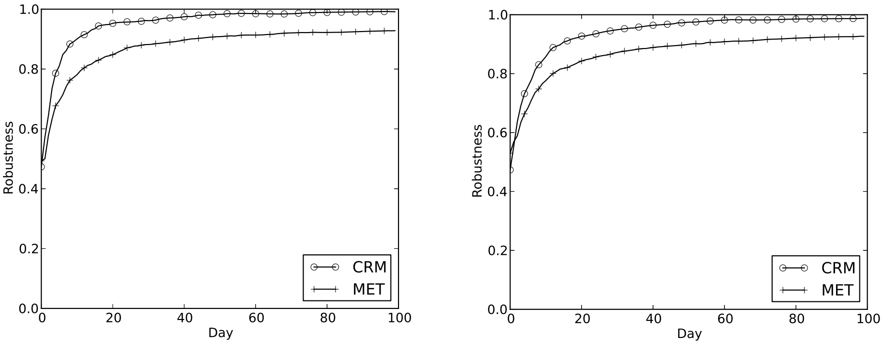
4.3.2 Experiment on robustness on static reputation

Experiments were carried out to evaluate the robustness of the reputation model. In Table 4.3, the two models have almost the same results with

CRM slightly outperforming the MET model. All the results are consistent with the results published on paper [Jiang et al., 2013]. However, when CRM results are observed more carefully, it may be thought strange that $\mathscr{R}(CRM, Sybil\ Cam)$ is more robust than $\mathscr{R}(CRM, Cam)$. The reason is that in performing the Camouflage attack, all attackers must first establish their trust before day 20, and then submit unfair ratings. Luckily, the dishonest consumer give fair rating at the beginning of the attack, the fair rating are used to facilitate the reveal of the actual reputation of the honest service provider. Consequently, in the next few days, honest consumers will choose the provider with high reputation. We plot the daily robustness value for Sybil Whitewashing attack in Figure 4.3(b). This result in Table 1 is consistent with Figure 4.3(a), in which the robustness value increases faster than under Sybil Whitewashing attack 4.3(b). Note that the two models yield high robustness values on the final day. The CRM curves in Figure 4.3(a) and 4.3(b) show that it converges to the excepted robustness value faster than the MET model. This rapid gain property can help the model to resist attacks performed at the very beginning stage. The underlying reason why CRM gains fast robustness value at the beginning is that CRM generates a public reputation value that can be observed by all consumers, while in MET a consumer can learn the quality of only service providers included in its trust network.

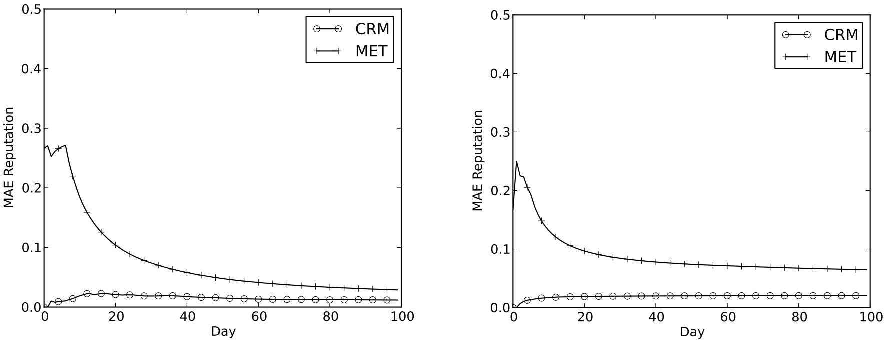
4.3.3 Comparison of MAE

In Tables 4.1 and 4.2, for both honest and dishonest duopoly sellers, the clustering-based reputation model attains the best results. CRM shows significant improvement considering the deviation of rating scores is set as $\sigma = 0.05$ in simulation setup. Both models can mitigate the influence of malicious raters. The MAE reputation value for dishonest service providers is much higher than that of honest providers, and this result is consistent with the MAE result shown in [Jiang et al., 2013]. The MAE for non-Sybil attack is generally lower than that of the corresponding Sybil attack; the main reason is that the number of fair ratings on honest and dishonest providers is



(a) Robustness vs. Sybil Cam (b) Robustness vs. Sybil WW

Figure 4.3: Static Reputation Situation: Robustness of Reputation Model under Attack.



(a) Honest duopoly provider (b) Dishonest duopoly provider

Figure 4.4: Static Reputation Situation: MAE of Duopoly Provider Reputation under Sybil WW attack.

decreased. For MET, this makes the trust network sparse, hence, it is hard to build up the level of trust in the advisor. Sybil Camouflage is an exception as the malicious rater would like to subvert the reputation of a provider. It first pretends to be a fair rater and tries to gain a high level of trust. Such fair ratings help the other consumers to select the fair service provider. Hence, the MAE is not sensitive to Sybil attack.

The daily MAE reputation shown in Figure 4.4(a) and 4.4(b) reveals that the CRM converges to the true reputation value faster than MET. When no honest consumers interact with the provider, the estimated reputation value can not be calculated and keep as it is. Therefore, at the very beginning, the MAE reputation value remains at 0.0.

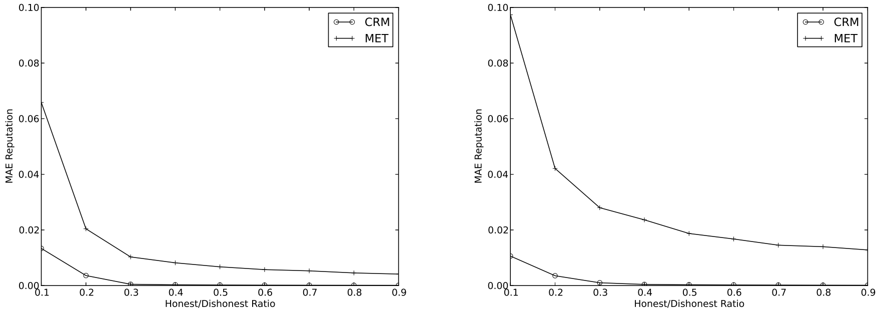
To evaluate the arbitrary ratio of honest and dishonest consumer, we conduct a series of experiments on 100 consumers with the ratio of honest consumers settled from 0.1 to 0.9. The MAE is evaluated on the 50th day. And the result is shown in Figure 4.5(a) and 4.5(b).

The MAE reputation tendency in the figure indicates that the accuracy of the model is somehow is affected by the ratio of honest and dishonest consumers. But, our proposed model have a better performance even in a situation the dishonest consumers are majority.

When the trustworthiness of the consumer is also considered to detect the unfair rating in ρ-ϕ graph. We conduct the experiment on honest seller and the result is shown in Table 4.4. It can facilitate the discrimination of unfair rating in most cases and result in accurate reputation value. But, for the Camouflage attack, it may mistake some unfair rater because they have a high trustworthiness value.

4.3.4 Robustness comparison on dynamic reputation

The reputation of a service provider is always dynamic and changes with time. At some point, the provider may update its service quality by offering a better service to consumers. A popular used model that can reflect the dynamic quality changes of a service is the pairwise model. The service providers change their strategies in halfway, either



(a) Honest duopoly provider

(b) Dishonest duopoly provider

Figure 4.5: MAE of Duopoly Provider under Sybil Camouflage attack.

Table 4.4: Mean Absolute Error (MAE) of Reputation Estimation for Honest Duopoly Service on Static Reputation with ρ vs. ρ-ϕ Graph.

Models	Constant	Camouflage	Whitewashing
CRM-ρ	0.009±0.002	**0.008±0.002**	0.009±0.001
CRM-ρ-ϕ	**0.006±0.003**	0.012±0.005	**0.007±0.001**

Models	Sybil	Sybil Cam*	Sybil WW*
CRM-ρ	0.014±0.003	**0.012±0.003**	0.014±0.003
CRM-ρ-ϕ	**0.009±0.002**	0.013±0.003	**0.011±0.004**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

from high reputation value to low in order to rip off the attained reputation [Wang et al., 2011, Xiong and Liu., 2004]. On the other hand, they learn from their previous mistakes and ameliorate their behavior accordingly [Zaki and Bouguettaya, 2009, Vogiatzis et al., 2010, Wu et al., 2013]. In this subsection, we conduct further experiments to compare the dynamic adaption ability of reputation models. The service quality for a dishonest service provider is updated to 0.9 at day 50, but for the dishonest service provider, they keep attacking the honest providers with various strategies. One problem is that all honest consumers know which provider has the high-

Table 4.5: Robustness of Reputation Models under Different Attack Models on Dynamic Reputation.

Models	Constant	Camouflage	Whitewashing
MET	0.521±0.067	0.528±0.086	0.523±0.062
CRM	**0.564±0.065**	**0.561±0.052**	**0.575±0.052**

Models	Sybil	Sybil Cam*	Sybil WW*
MET	0.517±0.089	0.485±0.144	0.485±0.130
CRM	**0.570±0.082**	**0.557±0.085**	**0.561±0.104**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

Table 4.6: Mean Absolute Error (MAE) of Reputation Estimation for Honest Duopoly Service on Dynamic Reputation.

Models	Constant	Camouflage	Whitewashing
MET	**0.007±0.002**	**0.007±0.002**	0.008±0.002
CRM	**0.007±0.002**	0.008±0.003	**0.007±0.003**

Models	Sybil	Sybil Cam*	Sybil WW*
MET	**0.012±0.005**	0.014±0.006	**0.012±0.005**
CRM	0.013±0.005	**0.012±0.005**	0.013±0.005

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

er reputation before day 50. Consequently, all of them will select the better provider for interaction, there is no chance of discovering the emergence of a potentially good provider. We force the consumer to interact with other service providers by setting a random service selection probability, e_i, of 0.1, it represents a balance between exploration and exploitation. In order to give more time for the model to adapt to the changes, the experiments were extended to 200 days. As the robustness function defined before can not be directly applied to determine the reputation value of the dishonest duopoly provider (updated to 0.9), we update the definition of robustness as

Table 4.7: Mean Absolute Error (MAE) of Reputation Estimation for Dishonest Duopoly Service on Dynamic Reputation.

Models	Constant	Camouflage	Whitewashing
MET	0.103±0.024	0.101±0.031	0.102±0.024
CRM	**0.063±0.021**	**0.064±0.018**	**0.060±0.018**

Models	Sybil	Sybil Cam*	Sybil WW*
MET	0.122±0.034	0.136±0.050	0.132±0.047
CRM	**0.068±0.029**	**0.073±0.029**	**0.074±0.039**

* Sybil Cam: Sybil Camouflage; Sybil WW: Sybil Whitewashing

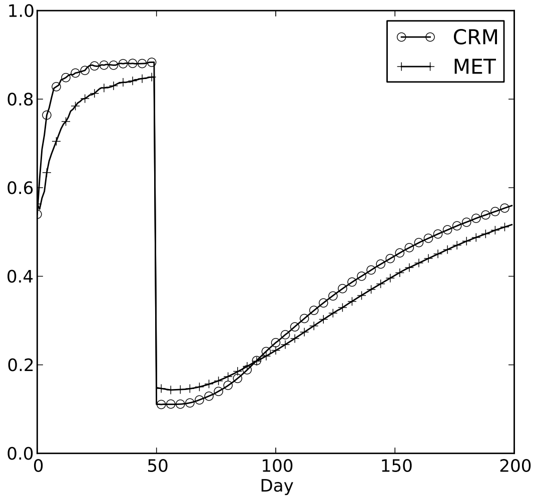


Figure 4.6: Robustness of Reputation Models on Dynamic Reputation.

follows:

$$\mathscr{R}(M,Atk) = \frac{Tran(s_i)}{c_H \times Days \times Ratio} \tag{4.15}$$

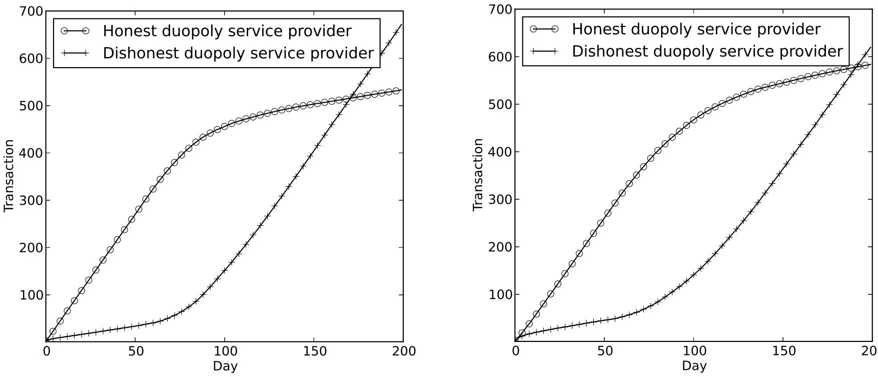
where $Tran(s_i)$ is the transaction volume of duopoly provider s_i with higher reputation value. Each experiment was conducted 50 times, the averaged results are listed in Table 4.5. Bold font indicates the best value at each time. They show that CRM outperforms MET on all attack model-

s. As in Figure 4.3, for static reputation, the robustness of CRM increases rapidly. However, when the dishonest provider updates its quality, according to equation 4.15, the robustness value of CRM should be smaller than that of MET because robustness is proportional to the transaction volume of duopoly provider with higher reputation value. Further detailed experiments examined the daily change in robustness. Figure 4.6 shows that CRM gain rapidly increases and when the dishonest duopoly updates its reputation value, its (CRM) robustness value is smaller than that of MET. These CRM characteristics are reasonable since at first more honest consumers interact with the honest duopoly provider, and fewer interact with the dishonest duopoly provider as the actual reputation value can be observed publicly. This process is confirmed in Figure 4.7(a) and 4.7(b). In the figure, we can observe that CRM adapts faster than MET as at day 70 the transaction slope of the higher reputation provider (previously dishonest duopoly provider) is larger than that of the lower reputation provider (honest duopoly provider). The fast gain metric makes the model more accurate than the state-of-art model according to the reputation evaluation model MACAU [Hazard and Singh, 2013].

4.3.5 MAE reputation comparison on dynamic reputation

As the reputation value of the dishonest duopoly provider is updated halfway, its MAE reputation value is obviously increased, which can be observed in Figure 4.8(b). This is made clear by comparing Table 4.7 with Table 4.2, the quality improvement made by the dishonest duopoly provider nearly doubles its MAE reputation value compared to the value in the static reputation situation on both CRM and MET. However, CRM not only can reduce the MAE reputation value to nearly half that of the MET model, but also can learn the actual reputation value faster than MET. This makes CRM adapt to the changes more quickly.

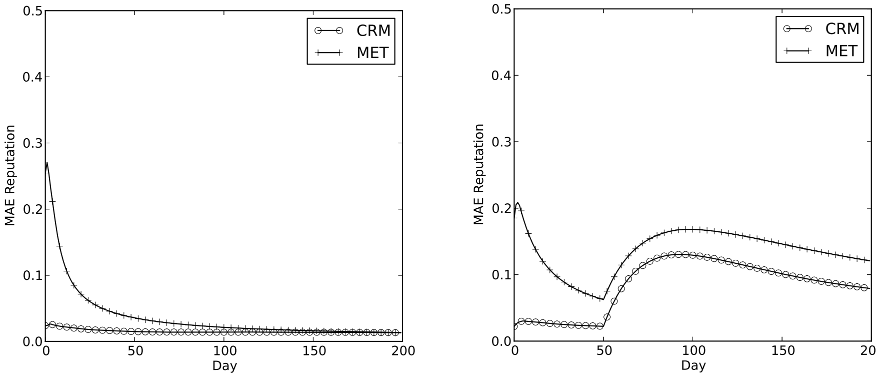
For honest consumers, the two models generate almost the same results in Table 4.6. Given a 200 day period, the two models both converge to the actual reputation value but with different convergence rates. MET is slower



(a) CRM vs. Sybil Cam

(b) MET vs. Sybil Cam

Figure 4.7: Dynamic Reputation Situation: Transaction of Reputation Model under Attack.



(a) Honest duopoly provider

(b) Dishonest duopoly provider

Figure 4.8: Dynamic Reputation Situation: MAE of Duopoly Provider under Sybil WW.

than CRM because the different trustworthiness networks must communicate with each other and learn their rating value on the service, while in CRM, the information is public and every consumer can use the estimated reputation value to make rational decisions. The transaction volume evolved in Figure 4.7(a) for the honest duopoly provider is consistent with the result shown in Figure 4.6.

4.4 Analysis and Discussion

In large e-commerce systems, large size of transaction data will be received each day or each hour, it is necessary for the models process large volume of ratings. In this section, we first compare the computational complexity of our model and MET. Then we analyse the behaviors of dishonest raters and their corresponding results. Furthermore, based on those behavior, we are discuss on the heuristic power to design the sanctioning function. An instance for customizing the sanctioning function is shown to illustrate the flexibility of CRM. CRM makes it possible for the designer to create their own sanction function to catch the properties of various type of services.

4.4.1 Computational complexity

In MET, each service consumer maintain their own trust network and update the network whenever new observation received. To obtain the optimal trust network, one service consumer needs to compare its network with all the networks from its advisors. Suppose the trust network size is n, which is the number of advisors in the trust network. To compare the trust network with each other, one service consumer will iterate all its ratings N_r to calculate the fitness value on an advisor. That means, for each generation, to choose the advisors, the time cost is at $n \cdot {N_r}^2$. Actually, the cost for the evolutionary operation of crossover and mutation is only depends on the size of the candidate trust network. Thus, the computational complexity for MET is $O(n \cdot N_r^2)$.

While for CRM, it accumulates the trustworthiness of one service consumer by the sanctioning function. And on each time new evidence about a service provider is received, it only classify the evidence rating into clusters and update the trustworthiness value. By defining the sanctioning function above, it costs constant time for updating. While for clustering, supposing the number ratings that will be clustered denoted as N_c and $N_c < N_r$. The clustering algorithm by [Rodriguez and Laio, 2014] takes $O({N_c}^2)$ time to classify the ratings. However, in each time, only a small time slice of ratings are mixed for clustering. The reason behind is that the ratings are received according to the time they given. When ratings are clustered in previous time period, then, there is no need to classify those ratings again as the ratings in those clusters are not changed any more. That means, more previous ratings presented in the clustering algorithm will not affect the result of new ratings. This property makes CRM is more preferable to large size of data. For MET, although it can limit the compared ratings to certain number, but it may loss the accuracy for comparison between two consumers. Two consumers that can pass the fitness examination may fail with only few ratings and vice versa. That is additional mechanism should be added to balance the trade-off between accuracy and computational complexity.

4.4.2 Strategy and cost analysis

All attacks are performed to benefit the dishonest provider. The main objective of introducing the reputation system is to increase transaction volume. We analyze the potential profit and the corresponding cost of each attack. Assume that a service platform has M service providers and N service consumers. The dishonest provider s_j colludes with dishonest consumers and intends to launch the unfair attack on s_i. To manipulate the reputation of s_i, dishonest consumers have to buy a service from s_i. And μ_{s_i} represents the payment from a dishonest buyer to s_i and can be considered as the cost of manipulation. The corresponding profit from an interaction with $s_j (j \neq i)$ is denoted τ_{s_j}, where $\tau_{s_j} < \mu_{s_i}$ and they can be measured by money or time etc. For CRM, the most effective attack is Sybil Camouflage attack. For whitewashing attack, it is hard to establish the trust of a service provider as the sanctioning function is admissible. In order to launch a successful attack in the clustering-based model CRM, the dishonest raters should keep their rating ratio ρ the same as the honest consumers. That is, transactions are not rated at the rate of $1-\rho$. The total loss for the attack is $\tau_{s_i} \times (1-\rho)$. Assume new consumers following the Poisson distribution with dynamic rate $\lambda(t)$ at time t [Khosravifar et al., 2010a], then the number of new consumers is $\int_1^t \lambda(x)dx$. Assume all the new consumers are misled by the dishonest provider S_j because of the successful attack. Profit is attained if the following inequalities are satisfied:

$$\int_1^t \lambda(x)dx \times \tau_{s_j} > \mu_{s_i} \times (N_d/\rho_d) \tag{4.16}$$

$$N_d > N_h, and\ \rho_d \approx \rho_h \tag{4.17}$$

where N_d, ρ_d is the number of ratings given by dishonest consumers and their rating ratio, while N_h, ρ_h is for honest consumers. Inequality 17 holds because of Sybil attack. Inequality 4.16 is to ensure the provider achieves profit from the attack. Conclusions can be made as follows:

- Decreasing rating ratio ρ increases attack cost and thus suppresses attack likelihood.

- The minimized number of new interactions required to ensure profit is: $(\mu_{s_i} \times (N_d/\rho_d))/\tau_{s_j}$. If value τ_{s_j}/μ_{s_i} can be viewed as the profit ratio of one type of service, we can derive the more profitable of one type of service (larger τ_{s_j}/μ_{s_i}), the more worthy to launch the attack. The conclusion is consistent with economic phenomenon.

4.4.3 The heuristic power of the sanctioning function

The model proposed here is actually a framework for a family of algorithms. If the specific sanctioning function reflects the underlying reputation value, then this model will have a better performance. In our model proposed above, we only consider if the received rating is in the fair clusters or not. In

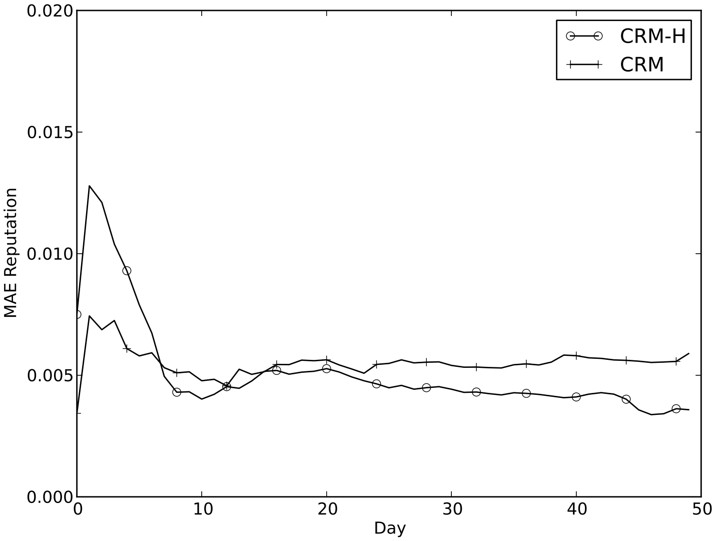


Figure 4.9: MAE of Honest Service Provider on Static Reputation under Sybil Camouflage attack.

this subsection, we will consider rating consistent from consumer to sanction the trustworthy. Rating consistent is referred as the variance from rating and reputation value. Here we update the sanctioning function as:

$$h(t, c_h, s_i) = h(t-1, c_h, s_i) + 1 + \xi \tag{4.18}$$

$$h(t, c_d, s_i) = h(t-1, c_d, s_i) - 1 \tag{4.19}$$

Where ξ is the reward if rating distribution from consumer is consistent with the reputation value, that is, the standard variance σ_{rr} of the span between rating and reputation value is small enough. Here we tentatively define:

$$\xi = 3, \; if \; \sigma_{rr} <= 0.1 \tag{4.20}$$

In the above equation, when the standard variance is smaller than 0.1, then we reward the consumer by increasing its trustworthiness by 3. The experiment uses 70 dishonest consumers and 30 honest consumers and the dishonest consumers perform the Sybil Camouflage attack. And the evaluation result is shown in Figure 4.9. When we change the sanctioning function to give more weight on honest consumers, the model CRM-H gives more accurate reputation value than the original CRM. However, it may fluctuate sharper than the original model when the number of ratings are few. As the ratings are accumulated, the aggregated reputation value is tend to stable.

4.5 Conclusion

In this paper, we proposed a clustering-based reputation model that can resist various types of attacks. The clustering approach is based on the rating ratios of consumers, the honest consumers have no incentive to rate each transaction. Dishonest consumers, however, will utilize every transaction to give an unfair rating to subvert the rating-based reputation system. The proposed model first classifies the ratings into clusters and detects the honest cluster based on the rating ratio of the cluster. It aggregates the reputation values from the honest customers by harnessing the Dirichlet distribution. Besides, the proposed model provides the flexibility for designer to create their own sanctioning function that fits the property of different type of service. Simulations showed that our model is more robust than the state-of-art model and its reputation estimates have low mean absolute error. Finally, as it is impossible to totally prevent unfair attacks, we conducted a preliminary analysis of the conditions under which it is worthwhile to attack our proposed reputation model.

Chapter 5

A reliable reputation model to evaluate Web services under rating scarcity

With the proliferation of Web services, more and more functionally equivalent services are being published by service providers on the Web. Although more services mean more flexibility for consumers, it also increases the risk of choosing as consumers may have little or no past experience with the service they will interact with. To this end, reputation systems have been proposed and have played a crucial role in the service-oriented environmen-t. Current reputation systems are mainly built upon the explicit feedback or rating given by consumers after experiencing the service. Unfortunately, services at the cold-start stage, prior to being rated, face the rating scarcity problem. We proposed the solution for this problem in this chapter.

5.1 Introduction

The service-oriented computing paradigm and its realization provide a promising approach to integrate computational resources seamlessly and dynamically across organizational boundaries [Alrifai and Risse, 2009]. In the service-oriented computing environment, such as Amazon Web Services and Language Grid [Ishida, 2011], two parties are involved: services offered by service providers and service consumers. Service consumers search and review the description of the services offered by service providers and select a service. With more and more web services being deployed on the Web, service consumers have more alternatives to select. As consumers may have little or no past experience with the service they will interact with, the risk of decision making also increases. Reputation systems were proposed to mitigate the risk that consumers faced when selecting a new service [Jøsang et al., 2007]. Existing service reputation systems, mainly based on the ratings given by service consumers, are one of the most important guides that the consumer has in making a decision, as they reveal how other consumers evaluated the services true ability in real scenarios. An example scenario is illustrated in Figure 5.1, a service consumer try to select one service for interaction among a list of functionally equivalent services. After the interaction, the service consumer gives his opinion on the satisfaction of service by rating. However, the ratings may be very sparse or unreliable for the following reasons:

1. Ratings are *skewed* towards high values [Hu et al., 2009]. Consumers cannot express their opinion truthfully if only numerical ratings are used [Ramn et al., 2014]. Moreover, they care about the impact of their feedback on the services future benefits in the marketplace, and so tend to offer a relatively high value unless extremely unsatisfied.

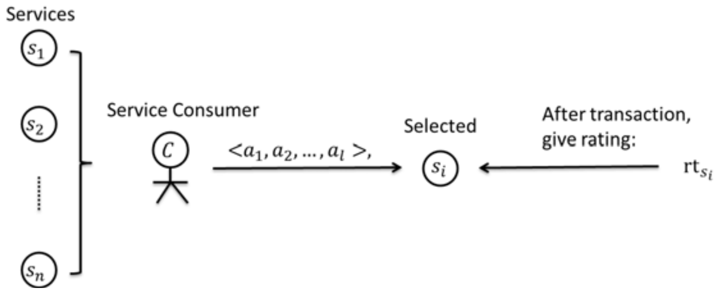


Figure 5.1: Scenario for a service consumer interact with services.

2. Not all customers rate the transaction [Cabral and Horiacȩsu, 2010]. As a result, transaction volume is much larger than the number of ratings received. Normal customers, those who pay for the service, have little interest in entering their ratings unless they are extremely satisfied or unsatisfied.

3. No rating is available at the cold-start stage [Arazy et al., 2009]. Upon the introduction of a service, no consumer has interacted with the service, so no historical evidence can be used to derive a reputation score for the service..

The rating scarcity problem is rarely addressed in the literature of the service domain. Some researchers briefly mentioned that it is a weakness common to rating aggregation systems [Chen and Singh, 2001]. They argue that the accuracy and stability of the system may be compromised by rating scarcity [Malik et al., 2009, Jøsang and Quattrociocchi, 2009]. To overcome the limitations of existing reputation systems in the service-oriented computing domain, we present implicit Reputation model (*imRep*), a new reputation model that integrates the consumers implicit judgments at the service evaluation moment with the explicit ratings given at the moment of transaction completion. The advantages of implicit judgements offer two benefits. First, implicit judgments are more broadly available since the number of alternative services is usually one or two orders of magnitudes higher than the number of ratings. Second, implicit judgments can more truthfully express the consumers preference for the service as the implicit actions of the consumers are not revealed and, consequently, the consumers do not bias their judgements towards high ratings. As a result, the obtained information is not skewed.

To describe our model, we consider the consumer decision of selecting service A, thus ranking A above some other alternative B, as the input of A defeating B in a match. Consumer decisions can thus be interpreted as a set of match outcomes. There are many algorithms [Elo, 1978], [Glickman, 1995], [Herbrich et al., 2007] that can be used to aggregate match outcomes. Our reputation system builds upon the Elo ratings system [Elo, 1978],which is widely used to evaluate chess players. In particular, we assign each service an initial rating and we treat each service in the context of consumer judgement as a participant in a chess tournament. Services that are selected will get their scores increased and those that are ignored get their scores decreased. The extent of the increase or the decrease depends upon the scores of the other services, i.e., the better the quality of the ignored service is, the more the scores of the selected services are increased. Similarly, the worse the quality of the selected service is, the more the scores of the ignored services are decreased.

5.2 The Reputation Model

In this section we describe a reputation model that builds on the service choices of consumers. First, we define the notations used in the model and present the detailed algorithm for ranking the Web services. Second, we extend the algorithm to rank the Web services based on QoS metrics. Finally, we discuss how we can combine our implicit reputation score with the ratings given by consumers to obtain a hybrid reputation model.

5.2.1 Notation

We represent the service-oriented computing environment as directed bipartite graph $G=(S,L,A)$; S is the set of services in the environment; L is the set of selections made by consumers. Edge $(s,l)\in A$ represents a consumer action a_s on the service $s\in S$ based on one selection $l\in L$. For example, in Figure 5.1 for each service candidates, the consumer give his actions as a_i. And we consider the following three consumer actions:

- *select:* the consumer selects the service for interaction;
- *review:* the consumer reviews the service for detailed information;

- *ignore:* the consumer reviews the service's brief description but takes no action it.

Among these three actions, we consider the first two as positive indications of service performance, and the last one as negative. We also assume that the consumer actions indicate a ranking on the Web services in the following decreasing order: select > review > ignore. For example, a service that is selected for interaction is considered better for the consumer than a service that is ignored. The objective of this paper is to compute a score, $r(s)$, for each service, s, that is informative of its QoS. Score $r(s)$ is considered informative if the relative difference between $r(s)$, $r(s')$ for services s and s' is predictive for the relative ranking of s, s' in subsequent matches.

5.2.2 Ranking Web Services with Elo Algorithm

The key idea of our model is to use consumer judgments as implicit information to compute the implicit reputation score $r(s)$ for service s. That is, alternative services $S_l \subset S$ at selection l of the consumer are taken as having competed in a tournament, and service performance at each selection is examined in pairwise manner. Certainly, services with better actions win over services with weaker actions (for example, select wins over review, review wins over ignore). Note that draws of services with identical better actions provide useful information about their relative qualities. The same does not necessarily hold for the case of draws of services with negative actions (such as two services that are ignored). Our reasoning is that using judgments of the very brief descriptions would increase the uncertainty in making decisions, and introduce judgment noise to the final result. Hence our algorithm exclude draws among negative action services. The scores are computed via a reputation calculation process on graph $G_l \subset G$ generated by each selection l, using the Elo constants for t_{elo}, K, as shown in Algorithm 3.

In the algorithm, we first initialize reputation score $r(s)$ for services s to 1.0. Then, when a consumer makes decision in selecting a service in process

```
1:  procedure IMRANK
2:      Inputs: Graph $G_l = (S_l, l, A_l)$.
3:      Output: Implicit reputation score for $s \in S$.
4:      for $s \in S_l$ do
5:          $T_{s,l} = 0, X_{s,l} = 0$
6:          for $s, s' : (s,l) \in A_l, (s',l) \in A_l, s \neq s'$ do
7:              $T_{s,l} += t(s,s',l)$
8:              $T_{s',l} += t(s',s,l)$
9:              $X_{s,l} += t_{elo}(s,s',l)$
10:             $X_{s',l} += t_{elo}(s',s,l)$
            ▷ Update Competition scores:
11:         $\tau(s,l) = T_{s,l} - X_{s,l}$
12:         $\tau(s',l) = T_{s',l} - X_{s',l}$
            ▷ Update implicit reputation scores:
13:         $r^i(s) = r^{i-1}(s) + \frac{K}{n-1} \cdot \tau(s,l)$
14:         $r^i(s') = r^{i-1}(s') + \frac{K}{n-1} \cdot \tau(s',l)$
```

Algorithm 3: Update implicit reputation scores

l, we update the service reputation score by considering every pair (s,s') of services available to the consumer c for judgment as a game in a tournament with possible outcome of matches:

$$t(s,s',l) = \begin{cases} 0, & \text{if } s \text{ lost against } s' \text{ at } l; \\ 0.5, & \text{if } s \text{ came to draw with } s' \text{ at } l; \\ 1, & \text{if } s \text{ won against } s' \text{ at } l. \end{cases} \tag{5.1}$$

After general initialization, for each selection made by consumers, we set the outcome variables $T_{s,l}, X_{s,l}$ to 0 for each alternative service s at selection process l made by consumer c. We compute $T_{s,l}$ as the sum of the actual points that s scored in selection process l against the other service candidates. Also, we compute the sum of expected points $X_{s,l}$ that s would earn against other service candidate $s' \neq s$ in the selection of process l, according to Elo's formula [Elo, 1978]. As each service competes with other services (lines 6-10 in Algorithm 3), the accumulated expected points and the actual points earned in selection l can be derived.

Finally, we update the reputation score of service s in an iterative way in line 13. $r^{i-1}(s)$ is the current reputation score while $r^{i}(s)$ is the updated reputation scored based on current value. K-factor represents the maximum possible adjustment per game (set here to 32), and is normalized by $n-1$, where n is the number of the services in one selection process. Without normalization, we would have dramatic inflation/deflation of scores with each selection process. The reputation score of a service is updated according to the average competition results between other services in each selection process. This averaging ensures that services gain according to their relative position in the service candidate ranking, rather than the number of alternative services. For example, for one selection process l_1 of consumer c_1, the number of alternative services is 32. While in another selection process l_2 performed by consumer c_2, the number of alternative services is 2. Assume among those alternative services, service s_1 gains 31 competition scores in l_1 where $\tau(s_1,l_1) = 31$, and service s_2 gains 1 competition scores in l_2 with $\tau(s_2,l_2) = 1$. As a result, if without considering normalization, the reputation score of s_1 gains 31 and s_2 only gains 1. In this situation, the judgement based on the preference of consumer c_1 is overwhelmed just because the number of alternative services in l_1 is larger than in l_2. On the contrary, the normalized gained scores for s_1 and s_2 is comparable considering the preference of c_1 and c_2.

5.2.3 Ranking Web Services on Metrics

The ranking scores generated in Algorithm 3 is a general value for the Web services. However, to accurately select the optimal service based on specific metric concern cluster, such as only focused on response time and cost, we extend the algorithm into metric-wise Web services ranking as following.

Initially, we calculate the implicit score $r^{i}(s)$ derived from Algorithm 3 for each service. Based on consumer c's actions on the service candidates, we update the implicit scores on each metrics on line 6-9. Different consumer have different preference, and they select the service according to

```
procedure MetricRanking
    Inputs: Graph $G_l = (S_l, l, A_l)$,
            Consumer $c$'s preference $\overrightarrow{p}$.
    Output: Implicit ranking score on metric $m$ for $s \in S$.
    for $s \in S_l$ do
        Calculating implicit scores $r^i(s)$ for $s$ by Algorithm 3.
        ▷Based on $c$, update implicit scores on metrics:
        for $p_j \in \overrightarrow{p}$ do
            $m^i_j(s) += r^i(s) \cdot p_j$
            $w_j(s) += p_j$
            $r^i_j(s) = \frac{m^i_j(s)}{w_j(s)}$
```

Algorithm 4: Metric-wise Ranking for Web Services

their preference. The preference $\overrightarrow{p}$ denotes all the QoS attributes, such as response time, throughput, availability etc. And when implicit score is used to update the implicit reputation of a service, the implicit scores on each metrics is also calculated. $m^i_j(s)$ represents the accumulated implicit scores from the evaluation of all consumers, and $w_j(s)$ is the accumulated weight on a specific metric p_j. Larger p_j on metric j means the consumer c think highly of p_j metric when selecting the services. And the consideration of consumer c on p_j will have a large influence on calculating $m^i_j(s)$. For short, for one service s, the implicit scores on metric p_j can be derived based all consumers C as:

$$r^i_j(s) = \frac{\sum_{c \in C} r^i(s) \cdot p_j}{\sum_{c \in C} p_j} \tag{5.2}$$

5.2.4 The Proposed Model

Our Algorithm 3 yields the implicit ranking score of the services, but explicit ratings will be available for some services. We expect that a hybrid reputation model that combines both types of information would yield better results. This subsection introduces the *imRep* model; it integrates the ranking yielded by implicit reputation scores and rating-based ranking into a service list to predict the true ranking with higher accuracy. Along with the ranking, the reliability of the reputation score for each service is allocated under the rating criteria.

When mapping the implicit reputation scores into rating scores, we consider both the implicit reputation ranking and the ranking from feedback. For all ratings of service s, we define two parameters to evaluate the ratings: mean of the ratings, μ_s, and the number of ratings, n_s. The higher of n_s is, the more reliable is the average rating, μ_s, of service s. Thus, for s, the final reputation score will approach μ. Assuming that the rating value is lies in the range [0.0, 1.0], the *imRep* method is illustrated in Algorithm 5.

```
1: **procedure** IMREP
2:     Inputs: $S$; implicit reputation score $ri_s$ for $s \in S$.
3:         Average rating $\mu_s$ for $s \in S$.
4:         Number of ratings $n_s$ for $s \in S$.
5:     Output: Normalized reputation value for $s \in S$.
6:     Let services with $max(ri_{s_i}), min(ri_{s_i})$ as $s_x, s_y$.
7:     $ri_{max} = (ri_{s_x} < 1.0\ ?\ 1.0 : ri_{s_x})$
8:     $ri_{min} = (ri_{s_y} > 1.0\ ?\ 1.0 : ri_{s_y})$
9:     $\mu_{max} = (\mu_{s_x} \neq 0\ ?\ \mu_{s_x} : 1.0)$
10:    $n_{max} = (n_{s_x} \neq 0\ ?\ n_{s_x} : \infty)$
11:    **for** $s_i \in S$ **do**
12:        $v_i = \mu_{max} \cdot \frac{ri_{s_i} - ri_{min}}{ri_{max} - ri_{min}}$
13:        $\alpha = 1/e^{n_{max}/(n_{s_i}+1)}$
14:        $r(s_i) = v_i \cdot (1 - \alpha) + \mu_{s_i} \cdot \alpha$
```

Algorithm 5: Generate normalized reputation value

Initially, we iterate over the implicit reputation score of each service and record the services with maximized score as s_x, and with minimized score as s_y. The first ranked service is taken as a fiducial reputation value. Other service reputations are calculated according to it. To avoid meaningless values, lines 7 - 10 of Algorithm 5 check and adjust the maximum or minimum value. Finally, for each service, the final reputation value determined from two parts. The first part is a normalized value from the implicit reputation score, and the second part is the average rating value. To balance these two parts, we use the relevant value between the number of ratings. As a service accumulates more ratings, more trust is laid on its averaged rating value.

Reputations should converge quickly, and be stable [Hazard and Singh, 2013]. In the next section, we will analyze the convergence conditions and the convergence speed of our model.

5.3 Convergence Analysis

In this section, we discuss the situations in which the competition scores can converge and the convergence rate. First, we define the convergence of the proposed reputation model here as the result that ranking is asymptotical to the ranking of QoS for all services with increasing competition. Suppose that in the service-oriented computing environment, the number of services in S is n, $S = \{s_1, s_2, ..., s_n\}$, and P is a family of subsets. Each item in P is a set of services ranked based on consumer selection. The extreme limit is when consumers select from the same set of services every time; no new subset is created in P.

Theorem 4. *We define the relation $\leq$ on set S as less than, by Algorithm 3, $(S, \leq)$ is a linearly ordered set. Set S can converge to the correctly ranked set if all the following conditions hold:*

1. *P does not contain empty set.*

2. *Any subset X of S, also exists in P.*

3. *The judgement number for each subset A of P is large enough.*

In mathematical notation, these conditions can be summarized as:

1. $|P| = 2^n - 1 \;\&\&\; \emptyset \notin P$

2. $\forall A \in P, |A| \to \infty$

Proof. It is easy to show that $\forall A \in P, (A, \leq)$ is a linearly ordered set. When consumer preference is not considered, the ranking result is convergent as any two elements in S are comparable: $\forall s_i, s_j \in S$, as $|P| = 2^n - 1$, either $s_i \leq s_j$ or $s_j \leq s_i$. That is, $\forall s_i, s_j \in S, \exists A \in P$ satisfied $s_i \in A$ and $s_j \in A$.

When we consider consumer preference, $\forall A \in P$ of the judgement result for a particular consumer, the order of $(A, \leq)$ may disagree with the correct order, but as $|A| \to \infty$, the consumer preference is offset and $(A, \leq)$ settles on a general ranking for each element in B. This process is consistent with the definition of reputation. Thus, set S not only can converge to a ranked set, but also can converge asymptotically to the correct order given the above conditions. □

The above analysis addresses the sufficient conditions for convergence, we discuss the necessary conditions for convergence as follows:

1. In the ideal condition, there are at least $n-1$ matches. In the ranked list, the front service s is exactly compared with the service that follows s. But, within $n-1$ competitions, service ranking is highly dependent on the preference of the consumer and inaccurate.

2. In our model, at least $n(n-1)/2$ matches are needed. Each service is compared with all other services. That is, the total number of matches is C_n^2. Ranking is not assured of converging if there are fewer matches than C_n^2.

3. Algorithm 3 will not converge if a certain percentage of consumer judgements is irrational. We will discuss this condition below.

The above analysis finds that the minimum acceptable condition for the convergence of Algorithm 3 is that the number of matches between services must not be less than $n(n-1)/2$.

The factors that control the convergence rate are:

- Service number n. More services mean more matches are needed to rank the services.

- Dependability of consumer judgement. If the judgment of most consumers is rational, then Algorithm 3 can converge because random judgments are offset and the rational judgments will form the ranking. If, however, consumers try to game the rating system consistently, the algorithm will fail to converge. In our research, the consumers are supposed to be dependable.

- The diversity, d, of consumer preference. Popular service usually has a outstanding performance on every QoS metric, an environment with various types of consumers will make the system converge quickly.

- Competition number m. The number of competitions between services contribute proportionally to the convergence rate.

In our paper, we assume that all the consumers are dependable one and so the competition result reflects the QoS of the service and preference. With this assumption, convergence rate λ can be written as:

$$\lambda \propto \frac{m \cdot d}{n} \tag{5.3}$$

Theoretically, the convergence rate is inversely proportional with the number of services. In the following section, we test the convergence and the accuracy of the proposed model with experimental evaluations using data of simulated and actual web services.

5.4 Evaluation

To evaluate the proposed reputation model, we compare *imRep* with both the real reputation value calculated by the WsRF algorithm [Al-Masri and Mahmoud, 2007] and the explicit average reputation model (AV). The average algorithm takes the mean of all explicit ratings as the reputation value and is widely used in commercial services like Amazon [Jøsang et al., 2007].

5.4.1 Environment setting

First, we consider a simulated environment with 100 service consumers and 50 services, the QoS parameters of the service are generated randomly. Second, seven real services listed in Table 5.1 are used to evaluated the performance of the proposed model [Al-Masri and Mahmoud, 2007]. In the following subsection, we try to evaluate the performance of *imRep* on services in the real world. We selected QoS parameters following earlier research [Menascé, 2002, Ran, 2003, Al-Masri and Mahmoud, 2007]:

1. Response Time (RT): the time taken to send a service request and receive a response (unit: milliseconds)

2. Throughput (TP): the maximum number of requests that can be handled per given unit of time (unit: requests/min)

3. Availability (AV): a ratio of the time period which a Web service is available (unit: %/3-day period).

4. Accessibility (AC): the probability that a system is operating normally and can process requests without any delay. (unit: %/3-day period).

5. Interoperability Analysis (IA): a measure indicating whether a Web service is in compliance with a given set of standards. (unit: % of errors and warnings reported).

6. Cost of Service (C): the cost per Web service request or invocation (cents per service request).

To facilitate service selection of a consumer, the preference of the consumers is simulated by weighting the above QoS parameter. The weight is uniformly selected from the range of [0, 1.0]. The simulation ran for 10 days, in each day, the probability of each service consumer searching for a service was 0.5. They first viewed the brief search results for further actions. They may review some services and select one for interaction. The services are reviewed and selected according to the user-centric QoS-based

Table 5.1: Qos Metric for various available mail verification Web Services.

ID	Service Provider & Name	RT	TP	AV	AC	IA	C
1	StrikeIron Email Verification	710	12.00	98	96	100	1
2	ServiceObjects DOTS Email Validation	391	9.00	99	99	90	5
3	StrikeIron Email Address Verification	912	10.00	96	94	100	7
4	CDYNE Email Verifier	910	11.00	90	91	70	2
5	XMLLogic ValidateEmail	720	6.00	85	87	80	1.2
6	Webservicex ValidateEmail	1232	4.00	87	83	90	0
7	XWebservices XWebEmail-Validation	1110	1.74	81	79	100	1

service discovery model [Al-Masri and Mahmoud, 2007]. For example, if the cost of service s_1 and s_2 is 0 cent and 5 cents respectively. And the response time of service s_1 and s_2 is 710ms and 391ms respectively. Suppose consumer c_1 sets all his weights to zero except cost, in this situation, he intend to minimize cost since it represents 100% significance to him. Under the QoS-based service discovery model, s_1 will be selected by consumer c_1. After the interaction, the consumer rates the performance of the service.

5.4.2 Evaluation Metric

Rating-oriented approaches must predict reputation values as accurate as possible. Therefore, differences between the predicted values and the true values are usually employed to evaluate the prediction accuracy. Mean Absolute Error and Root-Mean Square Error (RMSE) metrics are two widely adopted evaluation metrics for rating-oriented approaches. MAE is defined as

$$MAE = \frac{\sum_{i,d} |r_{i,d} - \hat{r}_{i,d}|}{N} \tag{5.4}$$

and RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i,d} (r_{i,d} - \hat{r}_{i,d})^2}{N}} \tag{5.5}$$

where r_i denotes the expected reputation value of service i at day d, $\hat{r}_i$ is the predicted reputation value, and N is the number of predicted values. However, since the object of this paper is to predict service ranking instead of predicting reputation values, we employ the Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2002] metric, which is a popular metric for evaluating ranking results. Given an ideal service QoS ranking (used as ground truth) and a predicted reputation ranking, the NDCG value of the Top-K ranked services can be calculated by

$$NDCG_k = \frac{DCG_k}{IDCG_k} \tag{5.6}$$

where DCG_k and $IDCG_k$ are the discounted cumulative gain (DCG) values of the Top-K services of the predicted ranking and ideal ranking, respectively. The value of DCG_k can be calculated by

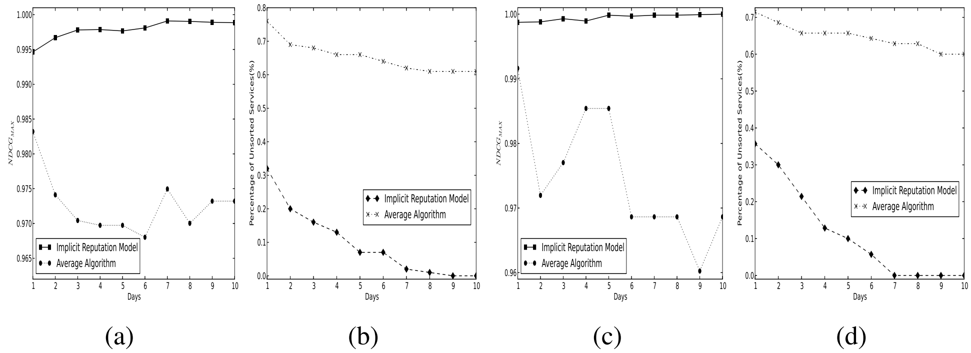
$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2 i} \tag{5.7}$$

where rel_i is the reputation value of the service calculated by WsRF algorithm at position i of the ranking. The premise of DCG is that high-quality service appearing lower in a ranking list should be penalized as the reputation value is reduced logarithmically proportional to the position of the result via dividing by $log_2 i$. The DCG value is accumulated from the top of the ranking to the bottom with the gain of each result discounted at lower ranks. The ideal rank achieves the highest gain among different rankings.

The $NDCG_k$ value is on the interval of 0 to 1, where larger value stands for better ranking accuracy, indicating that the predicted ranking is closer to the ideal ranking. The value of k is in the interval of 1 to n, where n is the total number of cloud services.

5.4.3 Evaluation of simulated Web services

Our proposed model is compared with WsRF and the average reputation algorithm. We assume that the rating criteria is [0, 1.0], and the rating offered by rational consumers on a service s after interaction follows $N(WsRF(s), \sigma)$ with probability $Pr = 0.4$. where WsRF(s) is the reputation value of service s calculated by WsRF. The ratings follow a normal distribution, while the other ratings skew towards high values with probability of $1 - Pr$.



(a) (b) (c) (d)

Figure 5.2: Evaluation results of ranking accuracy and availability of *imRep*.

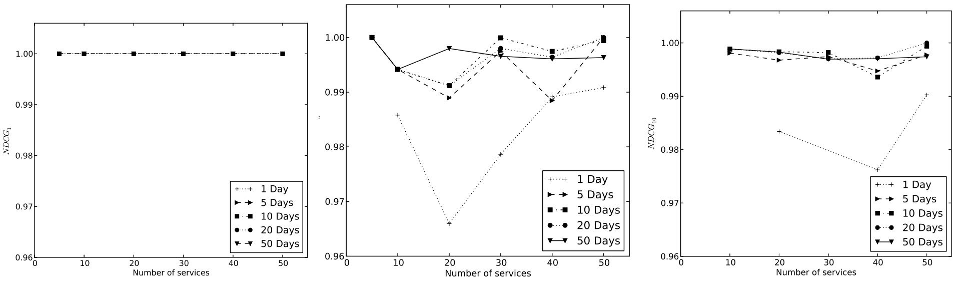
Scenario 1: Ranking accuracy test on 10 simulated web services. The accuracy of the reputation model is measured as the ranking accuracy because the real reputation value is unavailable. The ranking accuracy of the model is evaluated according to equation 5.6; higher values mean more accurate to ideal ranking. The ranking accuracy for *imRep* and the average algorithm is updated every day to show the detailed behavior of the model. Given that more data is accumulated each day, the ranking should converge

Table 5.2: Ranking comparison for *imRep* and the average algorithm.

Web Service	WsRF		imRep			Average Algorithm		
	Value	Rank	Value	Rank	∇ Rank	Value	Rank	∇ Rank
s_1	0.82	1	0.98	1	0	0.98	2	-1
s_2	0.66	2	0.55	2	0	0.99	1	1
s_3	0.55	3	0.42	3	0	0.0	N/A	N/A
s_4	0.53	4	0.35	5	-1	0.0	N/A	N/A
s_5	0.50	5	0.37	4	1	0.0	N/A	N/A
s_6	0.48	6	0.24	6	0	0.0	N/A	N/A
s_7	0.46	7	0.15	8	-1	0.0	N/A	N/A
s_8	0.45	8	0.18	7	1	0.0	N/A	N/A
s_9	0.43	9	0.0	10	-1	0.0	N/A	N/A
s_{10}	0.40	10	0.01	9	1	0.0	N/A	N/A

to the correct ranking. The ranking results of *imRep* and the average algorithm at the 10th day are shown in Table 5.2. In the table, we denote the ranking accuracy of a service without scores as N/A. The table shows that most services cannot be ranked because lack of data. Hence, it is hard for consumers to decide between those services. This situation happens because at the cold-start stage of a service, if the number of potential consumers is not large enough, most consumers will tend to select the service that has a reputation value. As a result, it is hard for newly deployed services to be accepted, and thus rated, by consumers. Although the *imRep* model wrongly ranked some services, the distance between the correct ranking is relatively small. The small QoS performance difference between those service pairs, such as s_7 and s_8, make it hard to distinguish between the two services. Given rating scarcity, with the limited amount of data available, this result is acceptable to consumers.

Besides considering the ranking on the final day, we calculate the ranking accuracy over time. To mitigate randomness, the simulation was run for multiple times, and the average results are plotted in Figure 5.2. On day 10, some services may not receive any implicit or explicit scores, making it impossible to compute $NDCG_k$ for those services. We denote the ranking accuracy of each such service as N/A to simplify the comparison. When plot the figure, it is hard for all reputation model get the same number of ranked service, we use $NDCG_{MAX}$ denotes the $NDCG$ value for the maximized number of service can be ranked. In the figure, the ranking accuracy of *imRep* tends to converge on the zero point. However, on some days, the decisions of some consumers may affect the convergence rate, which we discussed before. Furthermore, the percentage of services can be ranked is calculated in Figure 5.2-(b), the convergence of *imRep* is faster than the average algorithm. The detailed number of unsorted services is listed in Table 5.3. The *imRep* model can improve the ranking availability of service to 82.99% in this situation.



(a) (b) (c)

Figure 5.3: Evaluation results of ranking accuracy of *imRep* with different number of services.

In the previous section, we analyzed the impact of service number n on the convergence rate. We used different service numbers to test the average ranking accuracy of the proposed model on $NDCG_k$ against the number of evaluation days. Here, we use the top 1, 5 and 10 ranking accuracy of service to evaluate the performance of the models. Because most consumer only concern about the top ranked services. The result is plotted in Figure 5.3. In the figure, the accuracy for $NDCG_1$ is 1 denotes from the 1st day to the final day with different number of services, the *imRep* can derive the top one service with 100% percentage. In Figure 5.3-(b) and Figure 5.3-(c), for the 1st day, some ranking value is available due to the small number of services. And note that with certain number of services, the $NDCG_k$ value is fluctuate heavily, such as with 20 and 40 services to evaluate $NDCG_5$. The reason is that the experiments are independent with each other. With 20 services, the consumers used to evaluate the services are different with the situation of 40 services etc. This makes it different in evaluating the top k services. However, the *imRep* model can improve the ranking accuracy of the top services with days increasing. But after certain day, the ranking accuracy is unchanged or even decreased, that is because the lack of diversity of consumers in equation 5.2.

5.4.4 Evaluation on real services

With the popularity of Web services, it is easy to access various services. We adopt the dataset used in paper [Al-Masri and Mahmoud, 2007], all services are intended to validate e-mail. Details of the dataset are listed in Table 5.1. Unlike the original service order in [Al-Masri and Mahmoud, 2007], we reorder the services according to their WsRF values here for ease of comparison. As the QoS parameters have different units, we use min-max normalization to unify the value of each QoS parameter into the range [0, 1.0], as is widely used [Comuzzi and Pernici, 2009, Huang et al., 2009, Shi et al., 2012, Lin et al., 2012, Wang et al., 2015]. Based on our dataset, we use the following equation to normalize the QoS values.

For positive parameters (*TP , AV, AC and IA*):

$$q'_{s_i} = \frac{q_{s_i} - q^{min}}{q^{max} - q^{min}} \tag{5.8}$$

For negative parameters (*RT and C*):

$$q'_{s_i} = \frac{q^{max} - q_{s_i}}{q^{max} - q^{min}} \tag{5.9}$$

where q_{min} and q_{max} are the minimum and maximum values, respectively, for one QoS requirement. q'_{s_i} is the normalized value for service s_i.

Scenario 2: Ranking accuracy test on 7 real web services. As the dataset holds only service configuration, we need to simulate consumer preference.

Table 5.3: Ranking availability for *imRep* and the average algorithm on different test scenarios.

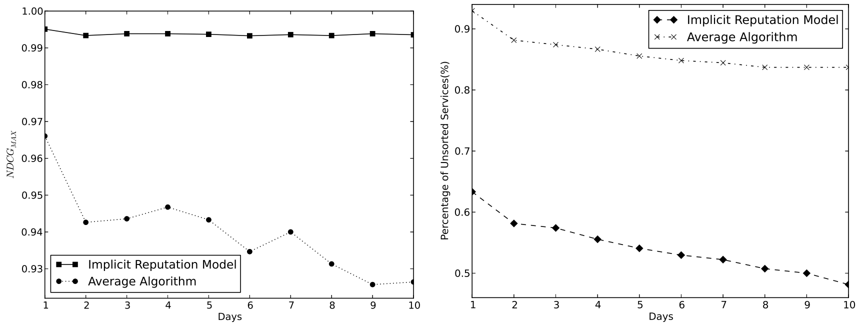
Days	Test Scenario 1			Test Scenario 2		
	AV	imRep	% Improvement (imRep vs. AV)	AV	imRep	% Improvement (imRep vs. AV)
1	7.6	3.2	+57.9%	5.0	2.5	+50.0%
2	6.9	2.0	+71.0%	4.8	2.1	+56.2%
3	6.8	1.6	+76.5%	4.6	1.5	+67.4%
4	6.6	1.3	+80.3%	4.6	0.9	+80.4%
5	6.6	0.7	+89.4%	4.6	0.7	+84.8%
6	6.4	0.7	+89.1%	4.5	0.4	+91.1%
7	6.2	0.2	+96.8%	4.4	0.0	+100.0%
8	6.1	0.1	+98.4%	4.4	0.0	+100.0%
8	6.1	0.0	+100.0%	4.2	0.0	+100.0%
10	6.1	0.0	+100.0%	4.2	0.0	+100.0%

Days	Test Scenario 3		
	AV	imRep	% Improvement (imRep vs. AV)
1	25.1	17.1	+31.9%
2	23.8	15.7	+34.0%
3	23.6	15.5	+34.3%
4	23.4	15.0	+35.9%
5	23.1	14.6	+36.8%
6	22.9	14.3	+37.6%
7	22.8	14.1	+38.2%
8	22.6	13.7	+39.4%
8	22.6	13.5	+40.3%
10	22.6	13.0	+42.5%

The consumer configuration is the same as in the total simulation environment. 100 consumers are active in the environment and they try to choose the service according to their preference and the QoS values of a service. We ran the experiments multiple times and plot the averaged results in Figure 5.2 (c) and (d); detailed number of unsorted services are given in Table 5.3(*Test scenario 2*). In Figure 5.2-(d), *imRep* has faster convergence than in Figure 5.2-(b). The differences between these two experiments are: 1) the number of services; 2) the performance gap between those services. It reasonable that a larger difference between services will make it easier for the algorithm to rank the services.

Scenario 3: Ranking accuracy on 27 real services with search function. To evaluate the reliability of the proposed model under various type of services, we extend our above experiment by using 27 real services with search function collected in paper [Al-Masri and Mahmoud, 2008]. The results are plot in Figure 5.4, the ranking accuracy for services is above 0.99 when compared with average algorithm from the first day to the last day. And the ranking information availability is above 40% from the second day for *imRep*. The figures is consistent with the result shown with simulated scenario.

Beside the overall reputation value on the services, we also evaluate the metric-wise ranking performance against average algorithm. The metrics are response time, throughput, availability, accessibility and interoperability of the service. The detailed results is shown in Table 5.4. For each metric, we evaluate the ranking accuracy on $NDCG_1$,$NDCG_3$,$NDCG_{10}$ on day 1, 5 and 10 respectively. Two points can be derived from the table: First, the proposed model can improve the ranking availability of the services on all metrics. When compared on $NDCG_{10}$, the ranking accuracy of top 10 services, *imRep* always generate ranking information based implicit behavior of consumers. Second, the ranking accuracy is worse than the average algorithm in some situation, the reason is that consumers usually select service with a balanced performance on all metrics. And a top ranked service on overall reputation score always have a good chance to have a top ranking on each metric. The times for consumer try to select this service is more frequently than other services, hence, the implicit information for this service is usually higher than other services. Approaches on how to increase the ranking accuracy of metrics will be a future work.



(a) (b)

Figure 5.4: Evaluation results of ranking accuracy of *imRep* on 27 real services.

Scenario 4: Convergence of reputation model in dynamic environment. Usually, the performance of a service dynamically changes with time. Is the reputation model stable enough to catch the changes and reflect the changes correspondingly? Although some changes are caused by consumers deliberately, such as collective attacks discussed in the literature [Whitby et al., 2004, Zhou and Matsubara, 2015, Zhou et al., 2015, Wang et al., 2012], we assume the consumers are rational and legal.

A previous experiment showed that service 7 had the lowest reputation. The day-to-day results of scenario 2 in Table 5.3 show that *imRep* basically converged on the correct ranking by the 5th day. In the robustness experiment, we update the QoS values for service 7 as (300, 10.00, 98, 95, 100, 1) at day 6. The updated QoS values for service 7 are not the best for every criterion, but have a competitive overall performance.

The experiment ran for ten days, to study the changes in ranking, and we used the implicit competition score output by Algorithm 3. The result is plotted in Figure 5.5. The changes in the scores of service 7 are shown by the solid line. When compared with scenario 2 in Table 5.3, on the fifth

Table 5.4: Ranking accuracy for *imRep* and the average algorithm on different QoS metrics.

Days	Response Time					
	NDCG1		NDCG3		NDCG10	
	imRep	AV	imRep	AV	imRep	AV
1	0.972	**0.997**	0.977	N/A	0.985	N/A
5	0.963	0.963	**0.977**	0.976	0.984	N/A
10	0.978	0.978	0.973	**0.979**	0.978	N/A

Days	Throughput					
	NDCG1		NDCG3		NDCG10	
	imRep	AV	imRep	AV	imRep	AV
1	1	1	0.450	N/A	0.616	N/A
5	0.015	0.015	0.397	**0.586**	0.670	N/A
10	0.246	0.246	**0.492**	0.367	0.580	N/A

Days	Availability					
	NDCG1		NDCG3		NDCG10	
	imRep	AV	imRep	AV	imRep	AV
1	1	1	0.988	N/A	0.955	N/A
5	1	1	0.962	**0.970**	0.949	N/A
10	0.431	0.431	0.721	**0.784**	0.835	N/A

Days	Accessibility					
	NDCG1		NDCG3		NDCG10	
	imRep	AV	imRep	AV	imRep	AV
1	1	1	1	N/A	0.969	N/A
5	1	1	0.899	**0.989**	0.943	N/A
10	0.446	0.446	0.730	**0.789**	0.858	N/A

Days	Interoperability					
	NDCG1		NDCG3		NDCG10	
	imRep	AV	imRep	AV	imRep	AV
1	0.471	0.471	0.645	N/A	0.772	N/A
5	0.706	0.706	0.599	**0.754**	0.678	N/A
10	1	1	0.931	0.931	0.843	N/A

day, the scores of the services indicate that the service ranking is stable. On the sixth day, when we updated the QoS values for service 7, the scores of service 7 in Figure 5.5 indicate that the implicit information based algorithm detected the changes. More consumers tended to consider service 7 because the performance of service 7 better matched the consumer preference. From the sixth day to the final day, the model reduced the scores of service 1 and increased the reputation of service 7. The distribution of the implicit scores converged quickly.

5.4.5 Analysis and discussion

In our experiments, we assumed that consumers with different preferences would select services with different QoS values. Jøsang et al. [Jøsang et al., 2007] defined reputation as what is generally said or believed about a services performance. Hence, reputation is an aggregate value from various consumers. By clustering the ratings by consumer preference, a consumer centered reputation system can be built and is able to provide

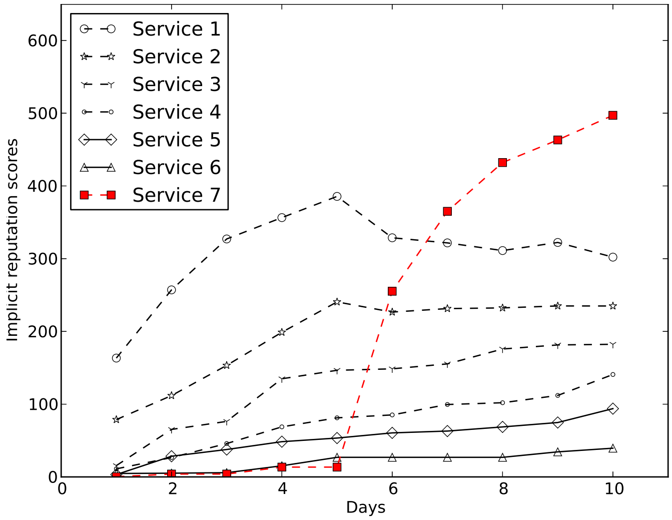


Figure 5.5: The daily changes of imRank scores changes when the 7-th service updates its performance.

more accurate reputation values for a specific consumer cluster.

Although the proposed model is focused on the cold-start stage of a service, Algorithm 5 combines the implicit scores with the ratings given by consumers to generate the reputation value for services. As more and more ratings are accumulated, the number of ratings will lead the reputation value to the rating value, line 13. All reputation systems that are based on ratings can benefit from the implicit scores to reach a more accurate reputation result.

In Scenario 2, only 7 real Web services were used to test the reputation accuracy of the models. To obtain a comprehensive assessment, we simulate service numbers ranging from 5 to 50. And we test our model on 27 real services with search function in scenario 3. Although larger dataset is available in WSDREAM [Zheng et al., 2014], some key information (the function of service) are unavailable for experiment and the QoS attributes are only limit to response time and throughput.

5.5 Conclusion

To overcome the rating scarcity problem, we proposed a reputation model based on the implicit behaviors of consumers. The proposed model considers the judgement actions from consumers on alternative services as a competition tournament among services, where service ranking is updated with each match. The convergence of the model was analyzed and experiments demonstrated the accuracy and convergence of the proposed model. This research provides ranking support for services without ratings at the cold-start stage and can boost the convergence rate towards the correct ranking. The extended metric-wise algorithm can also facilitate consumer with specific preference.

Chapter 6

Conclusion

6.1 Summary of Contributions

The thesis presents three contributions toward robust reputation system for Web services. The first is the mechanism to mitigate the time lag and unfair rating using dynamic sliding window model. The second is a clustering-based algorithm to detect the unfair rating attacks and generate a reliable reputation value for service consumers. The last is a reputation model that is able to support newly deployed services in the service platform to reach a reliable reputation value. We will review these contributions, and round off the thesis with suggested future research.

1. We introduce a novel algorithm to tackle the time lag and unfair rating research issues in reputation systems. In the proposed algorithm, the distribution of ratings is continuously monitored to dynamically resize the sliding window. When the service provider changes its behavior, the algorithm sets a new window to remove the influence of previous behavior such that the latest reputation reflects the latest quality of the service provider. The Bayesian Linear Regression approach is used to aggregate received ratings and detect reputation changes even in service environment that exhibit violent fluctuations in quality. In order to facilitate the detection of unfair ratings, we im-

prove the basic dynamic sliding window based on the observation that 50%$\sim$60% of consumers fail to rate their transactions. This mechanism lifts the restrictive assumption made by the existing algorithm that a fixed number of consecutive ratings must be observed before unfair ratings can be identified.

Simulations showed that our algorithm was more accurate than published algorithms and a method used by current commercial services The proposed algorithm adapts itself to dynamic changes on the rating distribution unlike the existing algorithm that uses a fixed threshold for unfair rating detection. Furthermore, by introducing the ratio of rating number to transaction volume as an indicator, the improved algorithm outperformed the compared algorithm by 45% on average in relieving the time lag problem.

2. We proposed a clustering-based reputation model that can resist various types of attacks. The clustering approach is based on the rating ratios of consumers, the honest consumers have no incentive to rate each transaction. Dishonest consumers, however, will utilize every transaction to give an unfair rating to subvert the rating-based reputation system. The proposed model first classifies the ratings into clusters and detects the honest cluster based on the rating ratio of the cluster. It aggregates the reputation values from the honest customers by harnessing the Dirichlet distribution. Besides, the proposed model provides the flexibility for designer to create their own sanctioning function that fits the property of different type of service. Simulations showed that our model is more robust than the state-of-art model and its reputation estimates have low mean absolute error. Finally, as it is impossible to totally prevent unfair attacks, we conducted a preliminary analysis of the conditions under which it is worthwhile to attack our proposed reputation model. In practical concern, the model can be employed in the rating based web site, such as on restaurant or hotel.com, to detect the group attack in the system.

3. We proposed a reputation model based on the implicit behaviors of consumers. The proposed model considers the judgement actions from consumers on alternative services as a competition tournament among services, where service ranking is updated with each match. The convergence of the model was analyzed and experiments demonstrated the accuracy and convergence of the proposed model. This research provides ranking support for services without ratings at the cold-start stage and can boost the convergence rate towards the correct ranking. The extended metric-wise algorithm can also facilitate consumer with specific preference. In practical concern, the proposed model can help the service platform to promote the newly deployed services to consumers.

6.2 Future Directions

Based on our current research, following future directions are suggested.

- *Facilitating the detection of unfair rating attack based on text reviews.* In our first and second research topics, we focused on how to detect the unfair ratings. Besides the ratings given by consumers, the text review from consumer can also be used to facilitate the detection of unfair ratings. In the literatures, researchers have argued that the numeric based feedback can not expression the consumer's opinion precisely [Ramn et al., 2014]. Hence, the qualitative data (reviews) given by consumer can be combined with the corresponding rating to deduce the consumer's true opinion on the specific service. With the development of text mining techniques, the analysis of text data is possible. And by analyzing the text reviews given by consumers, the consumer's opinion on the service can be expressed more accurate [Hu and Liu, 2004, Pang and Lee, 2008]. However, it is interesting that some malicious consumer may also launch their malicious attack by providing fake review. The detection of those attack pattern is another research topic can be extended as a future work.

- *Game theory can be applied to analyze the unfair rating attacks.*

  In Section 4.4.2, we have conducted preliminary analysis the potential profit and cost for perform a specific type of attack. Different policy can be applied to different type of attacks. Game theory is powerful tool to analyze the benefit and penalty against various situation. It is interesting to extend the research with policies against the attacks and design a optimal situation in which the malicious consumers are not willing to launch their attacks.

- *Building a generalized framework that leads reputation into a stable status from beginning.*

  In our last research topic in Section 5, we proposed a simple algorithm that can leverage reputation from implicit information into explicit ratings smoothly. However, for reputation system proposed by other authors that are only based on explicit ratings, how to apply our proposed implicit reputation model can be extended. That is, a generalized framework that assists those reputation system boost up their reputation system into a stable status can be conducted as a future work.

  Last but not least, it is important and necessary for the proposed models to be tested on real data. We are looking forward to the real data from companies for academic research.

Bibliography

[Al-Masri and Mahmoud, 2007] Al-Masri, E. and Mahmoud, Q. H. (2007). Qos-based discovery and ranking of web services. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 529–534. IEEE.

[Al-Masri and Mahmoud, 2008] Al-Masri, E. and Mahmoud, Q. H. (2008). Investigating web services on the world wide web. In *Proceedings of the 17th international conference on World Wide Web*, pages 795–804. ACM.

[Al-Sharawneh et al., 2010] Al-Sharawneh, J., Williams, M.-A., and Goldbaum, D. (2010). Web service reputation prediction based on customer feedback forecasting model. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International*, pages 33–40. IEEE.

[Alrifai and Risse, 2009] Alrifai, M. and Risse, T. (2009). Combining global optimization with local selection for efficient qos-aware service composition. In *Proceedings of the 18th international conference on World wide web*, pages 881–890. ACM.

[Arazy et al., 2009] Arazy, O., Kumar, N., and Shapira, B. (2009). Improving social recommender systems. *IT Professional*, 11(4):38–44.

[Barber, 2012] Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning.*, volume 1. New York: springer.

[Cabral and Horiacçsu, 2010] Cabral, L. and Horiacçsu, A. (2010). The dynamics of seller reputation: Evidence from ebay*. *The Journal of Industrial Economics*, 58(1):54–78.

[Chen and Singh, 2001] Chen, M. and Singh, J. P. (2001). Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 154–162. ACM.

[Comuzzi and Pernici, 2009] Comuzzi, M. and Pernici, B. (2009). A framework for qos-based web service contracting. *ACM Transactions on the Web (TWEB)*, 3(3):10.

[Dellarocas, 2000] Dellarocas, C. (2000). Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 150–157. ACM.

[Deypir et al., 2012] Deypir, M., Sadreddini, M. H., and Hashemi, S. (2012). Towards a variable size sliding window model for frequent itemset mining over data streams. *Computers & Industrial Engineering*, 63(1):161 – 172.

[Elo, 1978] Elo, A. E. (1978). *The rating of chessplayers, past and present.* Arco Pub.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[Glickman, 1995] Glickman, M. E. (1995). The glicko system. *Boston University.*

[Goto et al., 2011] Goto, S., Murakami, Y., and Ishida., T. (2011). Reputation-based selection of language services. *Services Computing (SCC)*, pages 330–337.

[Hazard and Singh, 2013] Hazard, C. J. and Singh, M. P. (2013). Macau: a basis for evaluating reputation systems. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 191–197. AAAI Press.

[Herbrich et al., 2007] Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems 20*, pages 569–576. MIT Press.

[Hoffman et al., 2009] Hoffman, K., Zage, D., and Nita-Rotaru, C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1.

[Hu and Liu, 2004] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’04, pages 168–177, New York, NY, USA. ACM.

[Hu et al., 2009] Hu, N., Zhang, J., and Pavlou, P. A. (2009). Overcoming the j-shaped distribution of product reviews. *Commun. ACM*, 52(10):144–147.

[Huang et al., 2009] Huang, A. F., Lan, C.-W., and Yang, S. J. (2009). An optimal qos-based web service selection scheme. *Information Sciences*, 179(19):3309–3322.

[Huhns and Singh, 2005] Huhns, M. N. and Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*, 9(1):75–81.

[Irissappane et al., 2012] Irissappane, A. A., Jiang, S., and Zhang, J. (2012). Towards a comprehensive testbed to evaluate the robustness of reputation systems against unfair rating attack. In *UMAP Workshops*, volume 12.

[Ishida., 2006] Ishida., T. (2006). Language grid: an infrastructure for intercultural collaboration. *Proc. IEEE/IPSJ Symp. Applications and the Internet (SAINT ',06)*, pages 96–100.

[Ishida, 2011] Ishida, T. (2011). *The language grid: Service-oriented collective intelligence for language resource interoperability*. Springer Science & Business Media.

[Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

[Jiang et al., 2012] Jiang, D., Xue, J., and Xie, W. (2012). A reputation model based on hierarchical bayesian estimation for web services. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 88–93. IEEE.

[Jiang et al., 2013] Jiang, S., Zhang, J., and Ong, Y.-S. (2013). An evolutionary model for constructing robust trust networks. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 813–820. International Foundation for Autonomous Agents and Multiagent Systems.

[Jøsang, 2012] Jøsang, A. (2012). Robustness of trust and reputation systems: Does it matter? In *Trust Management VI*, pages 253–262. Springer.

[Jøsang and Ismail, 2002] Jøsang, A. and Ismail, R. (2002). The beta reputation system. In *Proceedings of the 15th bled electronic commerce conference*, pages 41–55.

[Jøsang et al., 2007] Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644.

[Jøsang and Quattrociocchi, 2009] Jøsang, A. and Quattrociocchi, W. (2009). *Advanced features in bayesian reputation systems*. Springer.

[Khosravifar et al., 2010a] Khosravifar, B., Bentahar, J., and Moazin, A. (2010a). Analyzing the relationships between some parameters of web services reputation. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 329–336. IEEE.

[Khosravifar et al., 2010b] Khosravifar, B., Bentahar, J., Moazin, A., and Thiran, P. (2010b). On the reputation of agent-based web services. In *AAAI*.

[Laguna et al., 2011] Laguna, J. O., Olaya, A. G., and Borrajo, D. (2011). A dynamic sliding window approach for activity recognition. In *User Modeling, Adaption and Personalization*, pages 219–230. Springer.

[Lee et al., 2015] Lee, K., Park, J., and Baik, J. (2015). Location-based web service qos prediction via preference propagation for improving cold start problem. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 177–184. IEEE.

[Lin et al., 2012] Lin, D., Shi, C., and Ishida, T. (2012). Dynamic service selection based on context-aware qos. In *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pages 641–648. IEEE.

[Liu et al., 2013] Liu, S., Yu, H., Miao, C., and Kot., A. C. (2013). A fuzzy logic based reputation model against unfair ratings. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, pages 821–828. International Foundation for Autonomous Agents and Multiagent Systems.

[Liu et al., 2011] Liu, S., Zhang, J., Miao, C., Theng, Y.-L., and Kot, A. C. (2011). iclub: An integrated clustering-based approach to improve the robustness of reputation systems. In *The 10th International Conference* *on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1151–1152. International Foundation for Autonomous Agents and Multiagent Systems.

[Livingston, 2005] Livingston, J. A. (2005). How valuable is a good reputation? a sample selection model of internet auctions. *Review of Economics and Statistics*, 87(3):453–465.

[Malik et al., 2009] Malik, Z., Akbar, I., and Bouguettaya, A. (2009). Web services reputation assessment using a hidden markov model. In *Service-Oriented Computing*, pages 576–591. Springer Berlin Heidelberg.

[Maybeck, 1979] Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, volume 2 (Mathematics in Science and Engineering). Academic press.

[Menascé, 2002] Menascé, D. A. (2002). Qos issues in web services. *Internet Computing, IEEE*, 6(6):72–75.

[Morris, 1976] Morris, J. M. (1976). The kalman filter: A robust estimator for some classes of linear quadratic problems. *Information Theory, IEEE Transactions on*, 22(5):526–534.

[Mui et al., 2001] Mui, L., Mohtashemi, M., Ang, C., Szolovits, P., and Halberstadt, A. (2001). Ratings in distributed systems: A bayesian approach. In *Proceedings of the Workshop on Information Technologies and Systems (WITS)*, pages 1–7.

[Mui et al., 2002a] Mui, L., Mohtashemi, M., and Halberstadt, A. (2002a). A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, pages 2431–2439. IEEE.

[Mui et al., 2002b] Mui, L., Mohtashemi, M., and Halberstadt, A. (2002b). Notions of reputation in multi-agents systems: a review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 280–287. ACM.

[Nepal et al., 2011] Nepal, S., Malik, Z., and Bouguettaya, A. (2011). Reputation management for composite services in service-oriented systems. *International Journal of Web Services Research (IJWSR)*, 8(2):29–52.

[Nguyen et al., 2010] Nguyen, H. T., Zhao, W., and Yang, J. (2010). A trust and reputation model based on bayesian network for web services. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 251–258. IEEE.

[Pang and Lee, 2008] Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

[Park et al., 2006] Park, S.-T., Pennock, D., Madani, O., Good, N., and DeCoste, D. (2006). Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 699–705. ACM.

[Qiu et al., 2013] Qiu, W., Zheng, Z., Wang, X., Yang, X., and Lyu, M. R. (2013). Reputation-aware qos value prediction of web services. In *Services Computing (SCC), 2013 IEEE International Conference on*, pages 41–48.

[Ramn et al., 2014] Ramn, H., Centeno, R., and Fasli., M. (2014). From blurry numbers to clear preferences: A mechanism to extract reputation in social networks. *Expert Systems with Applications*, 41(5):2269–2285.

[Ran, 2003] Ran, S. (2003). A model for web services discovery with qos. *ACM Sigecom exchanges*, 4(1):1–10.

[Reiser, 1979] Reiser, M. (1979). A queueing network analysis of computer communication networks with window flow control. *Communications, IEEE Transactions on*, 27(8):1199–1209.

[Resnick et al., 2000] Resnick, P., Kuwabara, K., Zeckhauser, R., and Friedman, E. (2000). Reputation systems. *Communications of the ACM*, 43(12):45–48.

[Resnick and Zeckhauser, 2002] Resnick, P. and Zeckhauser, R. (2002). Trust among strangers in internet transactions: Empirical analysis of e-bay's reputation system. In *Advances in applied microeconomics 11*, pages 127–157.

[Resnick et al., 2006] Resnick, P., Zeckhauser, R., Swanson, J., and Lockwood, K. (2006). The value of reputation on ebay: A controlled experiment. *Experimental Economics*, 9(2):79–101.

[Ricci et al., 2011] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook, recommender systems handbook. pages 1–35. Springer, US.

[Rodriguez and Laio, 2014] Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

[Sabater and Sierra., 2001] Sabater, J. and Sierra., C. (2001). Regret: A reputation model for gregarious societies. In *Fourth workshop on deception fraud and trust in agent societies, vol. 70.*

[Schneider et al., 2000] Schneider, J., Kortuem, G., Jager, J., Fickas, S., and Segall., Z. (2000). Disseminating trust information in wearable communities. *Personal and Ubiquitous Computing*, 4(4):245–248.

[Shi et al., 2012] Shi, C., Lin, D., and Ishida, T. (2012). User-centered qos computation for web service selection. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 456–463.

[Srivatsa et al., 2005] Srivatsa, M., Xiong, L., and Liu, L. (2005). Trust-guard: countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th international conference on World Wide Web*, pages 422–431. ACM.

[Teacy et al., 2006] Teacy, W., Patel, J., Jennings, N., and Luck, M. (2006). Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198.

[Teacy et al., 2012] Teacy, W. L., Luck, M., Rogers, A., and Jennings, N. R. (2012). An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artif. Intell.*, 193:149–185.

[Vogiatzis et al., 2010] Vogiatzis, G., MacGillivray, I., and Chli., M. (2010). A probabilistic model for trust and reputation. In *In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 1-1, pages 225 – 232. International Foundation for Autonomous Agents and Multiagent Systems.

[Wang, 1999] Wang, L.-X. (1999). A course in fuzzy systems. In *Prentice-Hall press, USA*.

[Wang et al., 2011] Wang, S., Zheng, Z., Sun, Q., Zou, H., and Yang, F. (2011). Evaluating feedback ratings for measuring reputation of web services. In *Services Computing (SCC), 2011 IEEE International Conference on*, pages 192–199.

[Wang et al., 2012] Wang, X., Liu, L., and Su, J. (2012). Rlm: A general model for trust representation and aggregation. *Services Computing, IEEE Transactions on*, 5(1):131–143.

[Wang et al., 2015] Wang, Y., He, Q., and Yang, Y. (2015). Qos-aware service recommendation for multi-tenant saas on the cloud. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 178–185. IEEE.

[Whitby et al., 2004] Whitby, A., Jøsang, A., and Indulska, J. (2004). Filtering out unfair ratings in bayesian reputation systems. In *Proc. 7th Int. Workshop on Trust in Agent Societies*, volume 6, pages 106–117.

[Wu et al., 2013] Wu, Y., Yan, C., Ding, Z., Liu, G., Wang, P., Jiang, C., and Zhou., M. (2013). A novel method for calculating service reputation. *Automation Science and Engineering, IEEE Transactions on*, 10(3):634–642.

[Xiong and Liu., 2004] Xiong, L. and Liu., L. (2004). Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *Knowledge and Data Engineering, IEEE Transactions on*, 16(7):843–857.

[Xiong and Liu, 2004] Xiong, L. and Liu, L. (2004). Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *Knowledge and Data Engineering, IEEE Transactions on*, 16(7):843–857.

[Xu et al., 2007] Xu, Z., Martin, P., Powley, W., and Zulkernine, F. (2007). Reputation-enhanced qos-based web services discovery. In *Web Services, 2007. ICWS 2007. IEEE International Conference on*, pages 249–256. IEEE.

[Zaki and Bouguettaya, 2009] Zaki, M. and Bouguettaya, A. (2009). Rateweb: Reputation assessment for trust establishment among web services. *The VLDB Journal – The International Journal on Very Large Data Bases*, 18(4):885 – 911.

[Zhang et al., 2012] Zhang, L., Jiang, S., Zhang, J., and Ng, W. K. (2012). Robustness of trust models and combinations for handling unfair ratings. In *Trust Management VI*, pages 36–51. Springer.

[Zheng et al., 2014] Zheng, Z., Zhang, Y., and Lyu, M. R. (2014). Investigating qos of real-world web services. *Services Computing, IEEE Transactions on*, 7(1):32–39.

[Zhou et al., 2015] Zhou, X., Ishida, T., and Murakami, Y. (2015). Dynamic sliding window model for service reputation. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 25–32. IEEE.

[Zhou and Matsubara, 2015] Zhou, X. and Matsubara, S. (2015). Towards robust reputation system based on clustering approach. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 33–40. IEEE.

Publications

Major Publications

Journals

1. **Xin Zhou**, Donghui Lin, Toru Ishida. "A Tournament-based Reputation Model for Web Services under Rating Scarcity". *IEEE Transactions on Services Computing. IEEE TSC.* 2016 (*In review*)

2. **Xin Zhou**, Ishida Toru, Yohei Murakami. "Service Reputation Assessment Using Bayesian Linear Regression Approach". *IEEE Transactions on Services Computing. IEEE TSC.* 2015 (*In review - Major revision*)

International Conferences

1. **Xin Zhou**, Donghui Lin, Toru Ishida. "Evaluating Reputation of Web Services under Rating Scarcity". *2016 IEEE 13th International Conference on Services Computing (SCC 2016)*, San Francisco, USA, June. 2016, pp. 211-218.

2. **Xin Zhou**, Toru Ishida, Yohei Murakami. "Dynamic Sliding Window Model for Service Reputation". *2015 IEEE 12th International Conference on Services Computing (SCC 2015)*, New York, USA, June. 2015, pp. 25-32.

3. **Xin Zhou**, Shigeo Matsubara. "Towards Robust Reputation System

Based on Clustering Approach”. *2015 IEEE 12th International Conference on Services Computing (SCC 2015)*, New York, USA, June. 2015, pp. 33-40.

4. **Xin Zhou**, David Kinny, “Energy-based Particle Swarm Optimization: Collective Energy Homeostasis in Social Autonomous Robots”. *The 2013 IEEE/WIC/ACM international conference on intelligent agent technology, WI-IAT*. Atlanta, Georgia, USA, 2013, pp. 31-37.