A Trust Prediction Approach Capturing Agents' Dynamic Behavior*

Xin Liu and **Anwitaman Datta**
Nanyang Technological University, Singapore
liu_xin@pmail.ntu.edu.sg, anwitaman@ntu.edu.sg

Abstract

Predicting trust among the agents is of great importance to various open distributed settings (e.g., e-market, peer-to-peer networks, etc.) in that dishonest agents can easily join the system and achieve their goals by circumventing agreed rules, or gaining unfair advantages, etc. Most existing trust mechanisms derive trust by statistically investigating the target agent's historical information. However, even if rich historical information is available, it is challenging to model an agent's behavior since an intelligent agent may strategically change its behavior to maximize its profits. We therefore propose a trust prediction approach to capture dynamic behavior of the target agent. Specifically, we first identify features which are capable of describing/representing context of a transaction. Then we use these features to measure similarity between context of the potential transaction and that of previous transactions to estimate trustworthiness of the potential transaction based on previous similar transactions' outcomes. Evaluation using real auction data and synthetic data demonstrates efficacy of our approach in comparison with an existing representative trust mechanism.

1 Introduction

Many internet-scale applications are evolving recently, fueled by and fueling applications open to all, e.g., Web 2.0 and P2P applications, where end-users play active roles in the system, and thus their activities in turn significantly impact the system's behavior and other users' experience with the same. For instance, different from traditional static web sites, Web 2.0 applications allow users to not only retrieve information but also contribute contents and share information with other users (e.g., Wikipedia[1]), or peer-to-peer (P2P) networks [Androutsellis-Theotokis and Spinellis, 2004] enable applications like content distribution (e.g., BitTorrent[2]), Distributed Computation (e.g., Seti@home[3]), etc., by establishing direct connections between the users. Such rich interactions among users enhance the Internet experience significantly. However, due to characteristics (e.g., openness, decentralization, heterogeneity, etc.) of the massively distributed systems, dishonest users can easily join the system and act maliciously or selfishly to achieve their goals. This thus arises serious security issue: how to ensure that the potential interaction partner will not harm the interests of the honest user in a transaction? A common approach to this issue is to build automated trust/reputation system [Jøsang *et al.*, 2007] to derive/maintain trust relationships among the agents[4].

Many existing trust mechanisms derive trust based on an agent's past interactions experience with its counterpart. In the case that direct experience is not available, trustor resorts to indirect experience collected from other agents who have interacted with the target agent. Various techniques (e.g., Bayesian approach [Teacy *et al.*, 2006; Zhang and Cohen, 2008; Liu *et al.*, 2009], eigen matrix [Kamvar *et al.*, 2003], subjective logic [Burnett *et al.*, 2010], etc.) are applied to study target agent's past behavior. However, these works assume a relatively static agent behavior thus are not able to capture an agent's dynamic behavior patterns, which are common in a large scale open system since self-interested agents are very likely to vary their behavior to achieve their goals. For instance, in an online auction site, a malicious seller may act honestly in selling cheap items to gather sufficient reputation and then cheat in selling an expensive item. Traditional trust mechanisms are likely to incorrectly predict the risky transaction as safe since a majority of the seller's past transactions were good.

In this work, we propose a trust prediction approach based on the target agent's historical information, aiming at capturing its dynamic behavior pattern to accurately estimate trustworthiness of the potential transaction with this agent. The main idea is that outcome of the potential transaction can be indicated by that of certain previous transactions, which have the most similar transaction context with the potential transaction. Specifically, given a set of a trustor's past transactions (by chronological order) with the target agent (i.e.,

*The work was partly funded by A-STAR grant numbered 072 134 0055

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

[2] http://www.bittorrent.com/

[3] http://setiathome.berkeley.edu/

[4] We call any system participator (e.g., human, software agent, node, etc.) as an agent

direct experience), we first define context of a transaction as a sequence of its immediate previous transactions (a transaction window), each of which is described by a feature vector. Note that feature selection is application dependent and we will demonstrate in evaluation section which features of a transaction are identified using online auction as an example. We then compare similarity of context of the potential transaction and that of previous transactions. The outcome of a specific previous transaction that has the most similar context with that of the potential transaction is used as the indicator to estimate trustworthiness of the potential transaction. The similarity between two contexts (with the same window size) is calculated by aggregating feature based similarity between a pair of previous transactions in the corresponding contexts.

In case that trustor's direct experience is not available/sufficient, it will contact other agents for indirect experience. The direct experience and indirect experience are merged (by chronological order) to serve as the knowledge repository (i.e., training data) for trust prediction. Note that in this work, we do not discuss how to filter out false indirect experience, which has been thoroughly studied in the community [Teacy *et al.*, 2006; Zhang and Cohen, 2008]. In order to further improve prediction accuracy, we propose to use multiple transaction window sizes and apply Dirichlet distribution to model multiple trust indicators.

The rest of this paper is organized as follows: In Section 2, we present related works regarding historical information based trust management mechanisms. In section 3, we elaborate a basic trust prediction approach, propose further refinements and analyze its computational complexity. Evaluation using real auction dataset and synthetic dataset is conducted to quantify the performance of the proposed approach which we compare with a representative trust mechanism in Section 4. We finally summarize this work with discussion of possible future directions in Section 5.

2 Related work

As the mainstream approach, historical information based trust prediction mechanisms apply various techniques to present, derive and update trust. EigenTrust [Kamvar *et al.*, 2003] is a reputation system developed for P2P environments. It tries to fulfill the goals of self-policing, anonymity, no profit for newcomers, minimal overhead and robust to malicious collectives of peers. EigenTrust uses transitivity of trust and aggregates indirect experience from friends and friends of friends (FoF), etc. to perform a distributed calculation to determine the eigenvector of a "trust matrix" over peers. EigenTrust relies on some pre-trusted peers, which are supposed to be trusted by all peers. In [Ravichandran and Yoon, 2006], a trust system on top of peer group infrastructure is proposed. The groups are formed based on a particular interest. To calculate trust, the authors introduced Eigen Group Trust, which is an aggregative version of EigenTrust [Kamvar *et al.*, 2003]. All the transactions rely on the group leaders, who are assumed to be trusted and resourceful.

There are some works deriving trust based on homophily [McPherson *et al.*, 2001; Kumar *et al.*, 2010] (i.e., people with the similar interest meet and interact often). Differently, our work applies the similar concept to transaction contexts (instead of agents) to study target agent's dynamic behavior.

A few works are proposed to partly address agents' dynamic behavior by applying some intuitive methods. The beta reputation system proposed by Jøsang et al. [Jøsang and Ismail, 2002] estimates reputation of an agents using a probabilistic model (i.e., based on beta probability density function). This model is able to estimate the reputation of an agent by aggregating feedbacks provided by multiple advisors. The authors introduced a forgetting factor in the posterior trust update to get rid of the effect of outdated interaction experience. TRAVOS [Teacy *et al.*, 2006] is a trust and reputation model for agent based virtual organizations. It uses time points when modeling trust and reputation of the target agent, thus is capable of modeling its behavior within a certain period of time.

Zhang et al. [Zhang and Cohen, 2008] is probably closest to our approach. It took into account agent's dynamic behavior by introducing the concept of time window. That is, the ratings of the target agent are partitioned into different elemental time windows. In each time window, the trustor counts the numbers of successful and unsuccessful transactions. The trustworthiness of the target agent is firstly calculated by aggregating numbers of successful and unsuccessful transactions in each time window (taking into account forgetting rate) and then is adjusted according to reputations of the indirect experience providers. We will compare our approach with this work in evaluation section to demonstrate advantages of our context similarity based approach.

3 Our approach

3.1 Notation

We denote by $\mathcal{A}$ the set of all agents in the system. We assume two types of agents, *customers*, which request service from other agents, i.e., *providers*, which provide service. A transaction happens when a customer accepts a provider's service. To indicate quality of a service, the customer can rate the transaction, where rating is a discrete quantitative variable in a certain range, denoted by $\mathcal{L} = \{L_1, L_2, ..., L_l\}$. For instance, the rating could be in the range of [1, 2, 3, 4, 5], where 1 to 5 represents *lowest quality*, *low quality*, *medium*, *high quality* and *highest quality* respectively. Θ_{a_x,a_y} denotes the transaction between provider agent a_y and customer agent a_x. Each transaction Θ is associated with a feature vector $F_\Theta = \{f^1_\Theta, f^2_\Theta, ..., f^d_\Theta\}$. Note that in order to efficiently combine various features, we normalize values (for each feature) to the same range, i.e., [0,1]. The features can be obtained by agent's profile or context of the transaction. For instance, in online auction site, the features can be item's category, price or period of the auction, etc. We will detail such example in evaluation section.

We denote consumer a_x's past transactions with the provider a_y by $\mathcal{T}^D_{x,y} = \{\Theta^1_{a_x,a_y}, \Theta^2_{a_x,a_y}, ...\}$, where D indicates direct experience. The indirect experience, i.e., past transactions obtained from other agents who have interacted with a_y are denoted by $\mathcal{T}^I_{x,y}$. Given a set of a_y's past transactions (ordered by the time they happen), we can study pattern of a_y's behavior by setting transaction context (i.e., transaction window), which is actually sequential sub set of the

transaction set, denoted by W_s, where s represents size of this transaction window.

3.2 Using direct experience

Consider the scenario where customer a_x needs to predict trustworthiness of a potential service provider a_y to decide whether to interact with it or not. When a_x has sufficient past transactions with a_y, a_x can utilize this as the knowledge pool to investigate a_y's behavior pattern to predict trustworthiness of the potential transaction with a_y.

The main idea is that, given the most recent transaction window W_s^r of a_x's sequential past transactions $\mathcal{T}_{x,y}^D$ with a_y (see Fig. 1 as an example), we select the most *similar* earlier transaction window and estimate trustworthiness of the potential transaction Θ based on rating of the transaction which is immediately following the selected earlier transaction window. The similarity between transaction windows reflects how similar is the context/situation of a specific past transaction to that of the potential transaction Θ. So outcome of the past transaction which has the most similar transaction context is a promising indicator of the real quality of Θ.

Algo. 1 illustrates how our approach predicts trustworthiness of a potential transaction in detail. Given trustor a_x's sequential past transactions (by chronological order) $\mathcal{T}_{x,y}^D$ with a_y, based on size of the recent transaction window W_s^r, a_x obtains the earliest transaction window W_s^e, which is actually a sequence of the first s transactions in $\mathcal{T}_{x,y}^D$. Based on features of the transactions, a_x calculates similarity between W_s^r and W_s^e using function $getSimilarity(W_s^r, W_s^e)$ (Line 6 in Algo. 1), which will be described in detail in the next sub section. a_x then slides W_s^e to right by one position to obtain a new W_s^e and calculates the similarity between W_s^r and the new W_s^e. This process continues until W_s^e reaches the last past transaction but one. After similarity calculation, a_x chooses the earlier transaction window which has the largest similarity score and obtains outcome (say $L \in \mathcal{L}$) of the past transaction which is immediately following that transaction window. Then outcome of the potential transaction is estimated as L. a_x finally decides whether to interact with a_y or not based on such prediction. For instance, if the predicted trustworthiness is higher than a predefined threshold, a_x enters the transaction, otherwise, a_x rejects a_y and evaluates other potential interaction partners.

Note that when a_x encounters a new potential transaction (i.e., after current potential transaction), the recent transaction window W_s^r is shifted to right by one position. So our approach is able to adapt to newly encountered transaction. We next discuss how to calculate similarity between two transaction windows in detail.

Similarity calculation

There exist many methods to measure similarity between a pair of objects (e.g., Jaccard index[5], cosine similarity[6], etc.). Given feature vector of each transaction, we apply a heuristic based on Euclidean distance to calculate similarity between the two transaction windows.

[5]http://en.wikipedia.org/wiki/Jaccard_index

[6]http://en.wikipedia.org/wiki/Cosine_similarity

Algorithm 1 Trust prediction algorithm

1: Given a_x's sequential past transactions (by chronological order) $\mathcal{T}_x^D$ with a_y, based on size of the recent transaction window W_s^r, a_x obtains the earliest transaction window W_s^e, which is the first s transactions in $\mathcal{T}_x^D$.
2: S' = 0 //initial similarity score.
3: w = -1 //index of the earlier transaction window
4: **while** TRUE **do**
5: a_x calculates similarity between W_s^r and W_s^e:
6: $S \leftarrow getSimilarity(W_s^r, W_s^e)$.
7: **if** $S > S'$ **then**
8: $S' = S$
9: w = index of current W_s^e.
10: **end if**
11: **if** current W_s^e reaches last past transaction but one of $\mathcal{T}_x^D$ **then**
12: Breaking WHILE loop.
13: **else**
14: Sliding W_s^e to right by one position to obtain a new W_s^e.
15: **end if**
16: **end while**
17: a_x obtains outcome (say $L \in \mathcal{L}$) of the past transaction which is immediately following the wth W_s^e.
18: a_x then predicts trustworthiness of the potential transaction Θ is L.
19: a_x finally decides whether to interact with a_x or not based on the prediction.

For the i^{th} ($i \in [1, s]$) transaction of the two transaction windows $\Theta_i \in W_s^r$ and $\Theta_i' \in W_s^e$, we calculate similarity between these two transactions using their features (Note that all transactions have the same set of features):

$$\sigma_{\Theta_i,\Theta_i'} = \frac{1}{\sqrt{\sum_{j=1}^{|F_\Theta|}(f_{\Theta_i}^j - f_{\Theta_i'}^j)^2}} \tag{1}$$

$\sigma_{\Theta_i,\Theta_i'}$ represents similarity between the two transaction windows at the ith transaction. Following chronological order, a_x is able to measure the overall similarity of the two transaction windows W_s^r and W_s^e by aggregating similarity of all transaction pairs (see Fig. 1 as an illustrative example where size of transaction window is 3):

$$S_{r,e} = \sum_{i=1}^{s} \sigma_{\Theta_i,\Theta_i'} \tag{2}$$

By taking into account multiple features, a_x can comprehensively measure how similar is the earlier transaction window to the recent transaction window. Clearly, the more similar the two transaction windows, outcome of the potential transaction is more accurately predicted based on that of the past transaction which is immediately following that earlier transaction window.

Improving prediction accuracy

We notice that if window size varies, the prediction outcome may also vary accordingly since transaction windows with different sizes may demonstrate different transaction context (i.e., may include or ignore certain behavior patterns). Therefore, in order to improve prediction accuracy, we vary

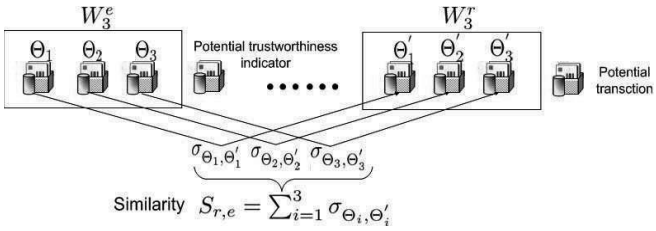


Figure 1: Measuring similarity between recent transaction window and earlier transaction window.

window size (for both recent transaction window and earlier transaction windows) to obtain multiple prediction results for a single prediction. Now the question is how to handle these results to derive the most accurate final prediction result. Given multiple levels of trustworthiness (see Section 3.1 for notations) of a transaction, we model the prediction results using Dirichlet distribution [Jøsang and Haller, 2007], which captures a set of observations that have multiple possible outcomes.

We assume for a single prediction, there are W different window sizes and the corresponding prediction results are denoted by $L' = \{L_k | k = 1, ..., W; L_k \in \mathcal{L}\}$. We then obtain Dirichlet probability density function:

$$f(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} p_i^{\alpha_i - 1}, \quad (3)$$

Where $\vec{p} = \{p_i \geq 0 | 1 \leq i \leq k\}$ denotes the $|\mathcal{L}|$-component random probability variable, and $\vec{\alpha} = \{\alpha_i | 1 \leq i \leq k\}$ denotes the number of predictions corresponding to one of the possible outcomes (derived from L'). The expected value of any of the $|\mathcal{L}|$ random variable (i.e., expected probability of a certain prediction result) is:

$$E(p_i|\vec{\alpha}) = \frac{\alpha_i}{\sum_{j=1}^{k} \alpha_j} \quad (4)$$

An intuitive way to decide the final result from multiple predictions is that the final prediction $L_k \in \mathcal{L}$ is the prediction result which has the highest expected probability.

3.3 Incorporating indirect experience

When service consumer a_x does not have sufficient past transactions with the potential service provider a_y (which is common in large-scale, open systems), it may resort to other agents who have interacted with a_y. However, selfish or dishonest agents may refuse to provide their experience with a_y or report false information. Many works [Xiong and Liu, 2004; Teacy *et al.*, 2006; Zhang and Cohen, 2008] are proposed to address false reports and discussion of such defence mechanisms is beyond the scope of this work. By contacting relevant agents, a_x collects a set of indirect past transactions with a_y, denoted by $\mathcal{T}_{x,y}^I$. Such indirect experience are merged with a_x's direct experience (if any) to form a knowledge base: $\mathcal{T}_{x,y} = \mathcal{T}_{x,y}^D \bigcup \mathcal{T}_{x,y}^I$. Utilizing $\mathcal{T}_{x,y}$, a_x is able to perform our proposed trust prediction algorithm (Algo. 1) to predict trustworthiness of the potential transaction. Note that the mixed direct experience and indirect experience are also sorted by chronological order.

3.4 Computational complexity

We assume that there are N (direct and/or indirect) past transactions. When single transaction window is considered, we denote the size of transaction window by S ($< N$) (for both recent transaction window and earlier transaction window). According to our algorithm, for a single trust prediction, there are $N - S$ transaction window similarity calculations and in each of such calculation, trustor calculates S times of similarity between a pair of past transactions. So the computational complexity of a single trust prediction is $O(NS - S^2)$.

When multiple transaction window sizes are considered for prediction accuracy improvement, the computational complexity changes accordingly. We denote set of possible windows size by $\mathcal{S}$. Note that $S_i (\in \mathcal{S}) < N$. So the computational complexity becomes $O(N \sum_{i=1}^{|\mathcal{S}|} S_i - \sum_{i=1}^{|\mathcal{S}|} S_i^2)$.

Computational complexity of our approach depends on number of past transactions N, transaction window size S_i and number of window sizes $|\mathcal{S}|$. We can manipulate these three variables to control computational complexity. In order to comprehensively study target agent's past behavior, we do not suggest to reduce N (i.e., removing old transactions) except that the number of past transactions is so large that computation speed is seriously slowed or the past behavior is obviously different from recent ones (e.g., a seller only got one or two negative comments when he just joined the system and then got all positive comments for the following 1000 transactions). Alternatively, we suggest to choose appropriate window size and number of these sizes such that the computational overhead is suitably reduced and trust prediction accuracy is not affected greatly. We will evaluate such discussion in the evaluation section.

4 Evaluation

4.1 Simulation settings

We use real dataset collected from an Internet auction site Allegro[7] as well as synthetic data to conduct experiments. The Allegro dataset contains 10,000 sellers, 10,000 buyers, more than 200,000 transactions and over 1.7 million comments. We assume binary outcome of a transaction, i.e., a transaction is considered successful if its feedback is positive, otherwise, it is considered unsuccessful. In order to fully understand how a seller changes its behavior in the transactions, we select a set of (150) sellers which have sufficient historical information (i.e., over 100 past transactions).

According to our approach, each transaction is described by a feature vector. By studying Allegro dataset, we identify five features which can be used to describe context of a transaction: (1) category of the item; (2) price of the item; (3) period of the auction (i.e., difference between auction end time and auction start time); (4) number of items already sold by the seller at the time the transaction happens and (5) outcome of the transaction. Note that in order combine these features, we first normalize their values to the range of [0,1] (for feature (2), (3) and (4)); for item category, if two transactions are within the same category, the difference of this feature is 0, otherwise, the difference is 1 (see Eq. 1); for

[7] http://allegro.pl/

transaction outcome, positive outcome is translated to 1 and negative outcome is translated to 0.

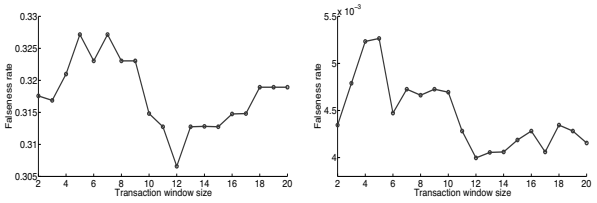
Real allegro dataset provides a real evaluation environment, however, behavior patterns of the sellers in real data are fixed. In order to comprehensively evaluate performance of our approach under different circumstances, and also to more flexibly control agents' behavior, we generate synthetic data derived from real data. Specifically, we generate a synthetic seller with 100 past transactions and simulate three types of its dynamic behavior: (i) for a transaction, the seller cheats or not randomly; (ii) the seller provides good service for the first half of the past transactions, then followed by bad quality services for the remaining half; (iii) the seller conducts several good transactions and then followed by a bad one (simulating the real scenario where a seller behaves honestly in selling cheap items to aggregate sufficient reputation and then cheat in a transaction selling an expensive item). In order to make the simulation more realistic, we further adjust outcomes of the transactions by configuring that a original good/bad transaction is good/bad with the probability of 0.85, otherwise, it is bad/good. Note that all the transactions features (and values) are taken from real Allegro dataset (i.e., a good/bad synthetic transaction is generated from a randomly selected good/bad real transaction). Using synthetic data, we compare our approach with a personalized trust mechanism[8] [Zhang and Cohen, 2008], which is closest to our approach (see related work section for a brief description). We call this compared approach PTE (short for personalized trustworthiness evaluation). Each experiment is repeated 30 times and error bars are added to indicate deviation of each running.

For both real and synthetic dataset, we use false positive (i.e., the transaction is unsuccessful but the algorithm predicted that it would be successful.) rate and false negative (i.e., the transaction is successful but the algorithm predicted that it would be risky.) rate as the metrics to evaluate performance of the trust mechanisms.

4.2 Results

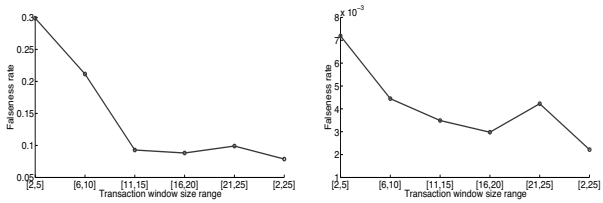
We first demonstrate performance of our approach using real Allegro dataset. Fig. 2(a) and 2(b) show false positive rate and false negative rate respectively when transaction window sizes vary from 2 to 20 (a specific window size is applied to all sellers). We observe that different transaction window sizes result in quite different trust prediction accuracy. The general trends of the falseness rates are: they ascend and then fall to the lowest point, and then ascend again. We show that when window size is 12, we obtain the lowest false positive rate and false negative rate simultaneously. The results indicate that by investigating impact of transaction window size, trustor is able to determine the optimal window size to achieve high trust prediction accuracy while incurring reasonable computational overhead (see Section 3.4)

We then conduct experiments to demonstrate how trust prediction accuracy is improved by applying multiple transaction window sizes. The transaction window size range [2,25] is split into 5 sub ranges, i.e., [2,5], [6,10], [11,15], [16,20] and [21,25]. We show falseness rates by applying different transaction window size sub ranges and compare the results with that when general range [2,25] is applied in Fig. 4. Obviously, by applying the general range, potential transaction's context is comprehensively studied thus generating the most accurate trust prediction. However, this incurs high computational overhead. We also observe that by simply applying certain sub ranges (e.g., [11,15], [16,20]), very high trust prediction accuracy can still be achieved. This again demonstrates that by choosing appropriate transaction window size (range), our approach provides high trust prediction accuracy while keeping computational overhead reasonable.



(a) False positive rate. (b) False negative rate.

Figure 2: Experiments using Allegro dataset (one transaction window size).



(a) False positive rate. (b) False negative rate.

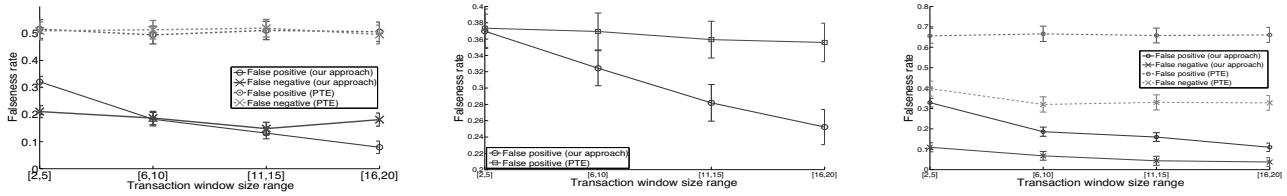
Figure 3: Experiments using Allegro dataset (multiple transaction window sizes).

We finally compare our approach with PTE using synthetic dataset. The comparison is conducted under three types of target seller's dynamic behavior patterns (see simulation setting Section 4.1). From Fig. 4(a) we observe that under behavior pattern (i) (i.e., the seller cheats or not randomly), for PTE, both false positive rate and false negative rate are kept at roughly 0.5. This is because in all time windows, trustor counts the similar numbers of successful and unsuccessful past transactions, which makes PTE approximate random selection. While our approach tries to match context of the potential transaction with that of previous transactions to estimate trustworthiness of the potential transaction, which demonstrates better results.

When behavior pattern (ii) is applied (see Fig. 4(b))[9], PTE performs poorly at the beginning because most of past transactions are successful, which make PTE incorrectly predicts that the potential transactions are more likely to be successful (which are actually risky). Later on, When more unsuccessful transactions are encountered, PTE adjusts the model parameters to detect the risky transactions. So the overall false positive rate is around 0.365. On the other hand, although our approach can not accurately predict potential transac-

[8] Since we assume a centralized simulation environment, i.e., online auction site, we assume that there is no false indirect experience.

[9] Since the seller mainly acts honestly for the first half transactions and acts dishonestly for the rest ones, we only show false positive rate.



(a) Dynamic behavior pattern (i). (b) Dynamic behavior pattern (ii). (c) Dynamic behavior pattern (iii).

Figure 4: Our approach vs PTE (synthetic dataset, multiple transaction window size).

tion's trustworthiness at the very beginning either, it is capable of quickly learning transactions' contexts thus lowering the overall false positive rate (around 0.25 when transaction window size range is [16,20]).

In Fig. 4(c) where behavior pattern (iii) (i.e., the seller behaves well for long, and then suddenly misbehaves for one transaction) is applied, PTE has false positive rate as high as around 0.65. This is because according to this behavior pattern, most of past transactions are successful, which is very likely to generate positive predictions. That is to say, PTE only looks at relative frequency of (un)successful transactions, while our approach is able to discern them since it is based on context rather than just the relative frequencies.

By comparing with PTE, we observe that under different behavior pattern scenarios, our approach outperforms PTE in general by matching transactions' contexts. Moreover, by choosing appropriate transaction window size range (e.g., [16,20] in this case), our approach improves trust prediction accuracy significantly.

5 Conclusion

In this work, we focus on studying impact of the agents' dynamic behavior on trust prediction. Our approach relies on agent's past behavior to compare contexts of the past transactions with that of a potential transaction. Trustworthiness of the potential transaction is then estimated based on outcome of a specific past transaction which has the most similar context. The context of a transaction is actually a transaction window just before that transaction and similarity between two transaction windows is calculated based on features of the corresponding transaction pairs.

Evaluation using real auction dataset and synthetic dataset demonstrates that by carefully choosing transaction window size, our approach is capable of accurately predicting trustworthiness of a potential transaction based on an agent's past transactions (i.e., detecting risky transaction). Moreover, in synthetic data based experiments, under multiple behavior patterns, our approach outperforms existing representative trust mechanism.

The accuracy of the current approach is greatly influenced by the order of the target agent's past transactions, which, however, does not always hold in reality. In the future work, we intent to relax such restriction. The possible solutions including introducing fuzziness in transaction windows matching algorithm or applying hidden markov model (HMM).

References

[Androutsellis-Theotokis and Spinellis, 2004] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36:335–371, December 2004.

[Burnett *et al.*, 2010] Chris Burnett, Timothy J. Norman, and Katia Sycara. Bootstrapping trust evaluations through stereotypes. In *Proceedings of the 9th AAMAS*, 2010.

[Jøsang and Haller, 2007] A. Jøsang and J. Haller. Dirichlet reputation systems. In *Proceedings of the 2nd Inter. Conf. on Availability, Reliability and Security*, 2007.

[Jøsang and Ismail, 2002] Audun Jøsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[Jøsang *et al.*, 2007] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43:618–644, March 2007.

[Kamvar *et al.*, 2003] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th WWW*, 2003.

[Kumar *et al.*, 2010] Udayan Kumar, Gautam Thakur, and Ahmed Helmy. Protect: proximity-based trust-advisor using encounters for mobile societies. In *Proceedings of the 6th IWCMC*, 2010.

[Liu *et al.*, 2009] Xin Liu, Anwitaman Datta, Krzysztof Rzadca, and Ee-Peng Lim. Stereotrust: a group based personalized trust model. In *Proceeding of the 18th CIKM*, 2009.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[Ravichandran and Yoon, 2006] Ajay Ravichandran and Jongpil Yoon. Trust management with delegation in grouped peer-to-peer communities. In *Proceedings of the 11th SACMAT*, 2006.

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

[Xiong and Liu, 2004] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE TKDE*, 16:843–857, 2004.

[Zhang and Cohen, 2008] Jie Zhang and Robin Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electron. Commer. Rec. Appl.*, 7:330–340, November 2008.