Effective Usage of Computational Trust Models in Rational Environments *

Le-Hung Vu, Karl Aberer
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{lehung.vu|karl.aberer}@epfl.ch

Abstract

Reputation-based trust models using statistical learning have been intensively studied for distributed systems where peers behave maliciously. However practical applications of such models in environments with both malicious and rational behaviors are still very little understood. This paper studies the relation between accuracy of a computational trust model and its ability to effectively enforce cooperation among rational agents. We provide theoretical results showing under which conditions cooperation emerges when using a trust learning algorithms with given accuracy and how cooperation can be still sustained while reducing cost and accuracy of those algorithms. We then verify and extend these theoretical results to a variety of settings involving honest, malicious and strategic players through extensive simulation. These results will enable a much more targeted, cost-effective and realistic design for decentralized trust management systems, such as needed for peer-to-peer systems and electronic commerce.

1. Introduction

The problem of managing trust and reputation in open and decentralized systems has attracted substantial research efforts in recent years [3]. A large number of work in this area focus on developing appropriate *computational trust models* to learn behaviors of participants based on their historical performance. Such model uses statistical or heuristic methods to aggregate ratings on past transactions of a target peer and related participants, from which to compute a trust metric as an indication of whether the target cooperates in the next transaction.

Computational trust models have been intensively studied and demonstrated to be robust under various (strategic) attacks by *malicious peers*, i.e., those want to take the system down at any cost by many types of strategic attacks, e.g., by submitting biased ratings to confuse the system and to make it less accurate in learning peers' behaviors. In fact, such models can effectively filter out biased information to obtain a correct picture of the peer's historical quality, as empirical evidences have shown [4]. However, these models strongly assume the probabilistic nature of all participants, ignoring the fact that many of them have economic incentives and behave rationally. In the presence of *rational* participants who behave strategically to maximize their expected life-time utilities, it is still unclear whether and how well a computation trust model can enforce cooperation in the system. As a typical example, a strategic peer can first cooperate to build reputation and then start cheating to increase life-time utilities. Although some simulation [7] suggests that existing computational trust models may also enforce cooperation in presence of both rational and malicious behaviors, no theoretical analysis has been performed to justify this.

In this paper, we prove that a computational trust model with sufficient statistical accuracy can be used effectively to motivate rational peers to cooperate in all but some of their last transactions. In such environments the key to enforcing cooperation is the effectiveness of the identity management scheme to effectively prevent whitewashing behaviors, rather than the computational model being used. This result gives an initial positive answer to the question whether existing trust learning algorithms in the literature produce the same effect: inducing the social optimum point in the system where rational participants cooperate with each other. This also implies that in a heterogeneous environment where peers use different learning algorithms with certain accuracy to learn trustworthiness of their potential partners, cooperation may also emerge. Second, we prove that it is sufficient to use an accurate algorithm with a low probability while still maintaining high cooperation in the system. As a result, we reduce the total implementation cost of the whole selection protocol significantly.

To the best of our knowledge, this is the first work studying the relation between accuracy of a trust learning model and its ability in enforcing cooperation in open environments, as well as analyzing the tradeoff between the cost of a computational trust mechanism and the additional benefits achieved by using it. Other work developing dedicated computational models to learn peer behaviors from historical performance (c.f existing surveys [3, 5]) are complementary to ours, since our results apply to any computational learning models with certain accuracy. The work in this paper is inspired by that of Dellarocas [2]. However our work is more thorough and different in many aspects: we have considered both accuracy and cost of reputation-based computational trust models, proposed a way to use them in an incentive-compatible and cost-efficient way, as well as performed empirical experimental studies of reputation effects on strategic agents in decentralized and dynamic environments. The most related work to our approach is [1], yet it addresses another problem of how to control the behaviors of agents in a centralized sovereign information sharing scenario. Our work proposes an effec-

*This work is partially done under the framework of the EPFL Center of Global Computing as part of the EU project NEPOMUK, contract No. 27705

tive usage of reputation information that is applicable to a wider range of applications with different degrees of centralization.

2. System Model

Consider a P2P application where each participant plays the role of a seller (provider) or a buyer (client) of certain resources (a sellable good item or a service) with certain prices and quality. Let u be the "legal" payoff of a seller in a transaction if he or she behaves honestly and yields a good transaction outcome. Example honest behaviors are to always ship the item or provide good service to the buyer after receiving payment. We assume that u lies within a range of minimal price u_* and maximal price u^*. If the seller cheats, i.e., does not ship the good or provides the low-quality services, it gains a further "illegitimate" amount v, where $0 \leq v < \infty$. In this case, the transaction is considered having a bad outcome. We assume that peers know the lower bound of prices u_* and that a buyer can estimate the illegitimate gain v of a seller in each transaction. Such an assumption is realistic: peers can learn these values by looking at trading history of other peers. In centralized systems the minimally accepted price u_* for each category of services/items can be defined. The illegitimate gain v can also be estimated easily, e.g., as the shipping cost plus the item value. The above environment is an abstraction of many practical P2P applications with different degrees of centralization, where participants are rational to a certain extent. Such a scenario represents, for example, a centralized eBay-like trading site on top of a social network, or a decentralized market of services. Consequently, our proposed solution can be used in all these applications.

We consider a computational trust model $\mathcal{R}$ as a dishonesty detector to evaluate the trustworthiness (reliability) of a rating with binary outcome, similar to a conventional spam detector in machine learning literature. Formal models and illustrating examples are given in the extended version of this paper [8].

The *accuracy* of a computational trust model is defined by two *misclassification errors*, α and β, where $0 \leq \alpha \leq 1$ is the probability that $\mathcal{R}$ misclassifies an unreliable rating as reliable. Inversely, $0 \leq \beta \leq 1$ is the probability that a reliable rating wrongly classified as an unreliable one. Such accuracy implies the ability of the computational trust model in eliminating effects of possible malicious attacks by biased raters. Practically, a bound on accuracy of a computational model can be measured either experimentally or via theoretical analysis. Definition 1 proposes an approach for a rational buyer to select a seller among candidates using such a dishonest detector.

Definition 1 *A buyer uses the following* seller-selection protocol $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ *for its transactions:*

1. *buyer gets the most recent binary rating r on the seller, considering the absence of a rating as the presence of a positive one*
2. *the binary reliability t of r is evaluated using the computational trust model $\mathcal{R}$*
3. *if $t = + \wedge r = -$ or $t = - \wedge r = +$, buyer publishes this cheating detection to a shared space*
4. *the seller is included for selection if there are less than $k \geq 1$ published cheating detections on it, otherwise the buyer ignores it.*

3. Main Theoretical Results

Theorem 1 shows the relation between the errors α, β of a computational trust model and its *incentive-compatibility*, i.e., its effectiveness in enforcing cooperation of a rational seller.

Theorem 1 *The selection protocol $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ makes it optimal for a rational seller, to cooperate in all but its last Δ transactions, where $\Delta = max\{1, \lfloor v^*/(u_*((1-\epsilon)^k - \epsilon^k)) \rfloor\}$. This holds even in presence of malicious and strategic manipulation of ratings by sellers, provided that the trust model $\mathcal{R}$ has misclassification errors α, β upper-bounded by some $\epsilon < 0.5$.*

The proofs of this theorem and related ones are available in the extended version of this paper [8]. As extensive simulations show later on, the above result also holds for the case peers use different computational trust models with different inputs and personalized settings, and the probability of detecting a bad rater is different for each peer. In those situations where there are certain (usually small) probabilities that an intrinsically honest seller appears as cheating to a buyer, e.g., it fails to ship the item, and a cheating seller satisfies the buyer, e.g., it sends a low-quality item yet still pleases the buyer. The inclusion of such probabilities in the analysis is also straightforward.

Theorem 1 implies that if the temporary illegitimate gain v^* is very high, e.g., trading of expensive items, the parameter $\Delta \to \infty$. Thus enforcing cooperation of a rational seller in such a transaction is impossible, which is an intuitive result. If sellers are long-term players staying in the system infinitely, the number of last Δ transactions plays no roles and thus any trust model with reasonably good accuracy ($\alpha, \beta \leq \epsilon < 0.5$) can be used as an effective sanctioning tool to motivate sellers' cooperation (Corollary 1). Otherwise, if sellers only participate in a limited number of transactions, or in case of high k values, very high levels of accuracy ($\epsilon < 0.05$) are required to reduce Δ, or to ensure cooperation of sellers in most transactions.

Corollary 1 *It is possible to use any computational trust model with misclassification errors upper-bounded by some $\epsilon < 0.5$ to effectively enforce cooperation of rational sellers who participate infinitely or in a very large number of transactions, even in presence of strategic rating manipulation by participants.*

Given rationality of peers, Corollary 1 implies that the key to ensure cooperation is not the accuracy of the trust learning algorithm but on an *identity management scheme* that is effective in preventing white-washing behaviors. For example, the establishing of a new identity can be made costly so that peers want to stay for many transactions rather than change their identities and start over. For centralized e-trading systems where trusted parties are present, a simple yet effective solution to ensure cooperation of all rational participants even in the last Δ transactions is to require each seller to deposit an approximate amount Δv^* to a trusted third party before being able to join the system.

This deposited sum will only be returned to the sellers if it intends to quit the system and only if it has not been detected as cheating by k or more others peers.

While higher accuracy of a computational trust model is generally desirable, it usually comes with higher cost. Example cost includes *communication* and *computation* cost to retrieve relevant ratings and to estimate rating reliability, or *opportunity cost* caused by missed trading opportunities by wrong and late decisions. Consider the case a buyer can choose either a computational trust model $\mathcal{R}_1$ or another model $\mathcal{R}_2$ to evaluate the trustworthiness of a rating. Suppose that $\mathcal{R}_1$ is an accurate detector with misclassification errors $\alpha_1 = \alpha, \beta_1 = \beta$, both upper-bounded by some $\epsilon < 0.5$, and with an expected cost $\mathcal{C}_1$. Let $\mathcal{R}_2$ be the simple computational trust model $\mathcal{N}$ (always trusts all ratings) with errors $\alpha_2 = 1, \beta_2 = 0$, and negligible cost $\mathcal{C}_2 \ll \mathcal{C}_1$. Since α, β are less than $\epsilon < 0.5$, the computational model $\mathcal{R}_1$ can be used to motivate the cooperation of a seller in most transactions (Theorem 1). However, this approach is costly to deploy. On the other hand, it is impossible to use only the naive trust model $\mathcal{R}_2 = \mathcal{N}$ for the same purpose since the seller can strategically manipulate ratings easily, e.g., by colluding with others to submit biased ratings. This use of the trust model $\mathcal{N}$ is less costly and thus more preferable. Theorem 2 proposes a way to optimize the cost of using expensive computational model $\mathcal{R}_1$ while still ensuring cooperation in the system (see [8] for the proof).

Theorem 2 *Consider the selection protocol $\mathcal{S}_1 = \langle \mathcal{R}, 1 \rangle$, in which the dishonesty detector $\mathcal{R}$ is implemented by using the trust model $\mathcal{R}_1$ with probability c and the naive model $\mathcal{N}$ with probability $1 - c$. The seller finds it optimal to cooperate in all but its last Δ transactions, where $\Delta = max\{1, \lfloor \frac{v^*}{u_*(1-2\epsilon)} \rfloor\}$, under the condition that $c \geq c_* = \frac{v^*}{\delta u_*(1-2\epsilon)}$, where $\delta \geq \Delta$ is the number of remaining transactions of the rational seller. This result holds in presence of strategic manipulations of ratings.*

If most sellers staying in the system infinitely or long enough, the mix of two computational trust models in Theorem 2, has an expected accumulative implementation cost $O(\mathcal{C}_1 log(N))$, where a seller stays in the system for N transactions [8]. Since the cost of using only one expensive trust model in the same case is $O(\mathcal{C}_1 N)$, mixing two computational trust models help to reduce the total implementation cost significantly. Hence, given the rationality of participants, the accurate trust learning algorithm $\mathcal{R}_*$ mostly plays the role of a sanctioning tool rather than the role of learning trustworthiness of potential partners.

4. Experimental Analysis

We use our generic trust prototyping and simulation framework for all experiments. The details on the modeling, implementation, and configuration files for all experiments in this paper are available online[1]. Various realistic conditions are simulated: peers can leave and join dynamically, buyers use computational trust models with different personalized inputs and settings, and peers exhibit several types of rating and serving behaviors. We then computed the number of transactions a seller participates from its up time by using a scaling factor K. K is set such that those peers with up-time approximating the mean of the overall uptime distributions participate in μ_{trans} transactions, where μ_{trans} is a parameter of the simulation. Experiment settings and corresponding results are summarized in Table 1, whose details are given in the extended version of this paper [8]. Seller types consist of: $s\%$ strategic, $g\%$ good, and $b\%$ malicious (bad). There are five rater types modeled: $h\%$ honest, $sr\%$ strategic, $a\%$ advertising, $b\%$ badmouthing, the rest are those peers leaving no reports after a transaction.

Three computational trust models were implemented and used in our simulation to evaluate the reliability of a rating. We then combined each of them with the naive algorithm $\mathcal{N}$ to estimate the total saved cost as proposed by Theorem 2. The first model $\mathcal{L}$ is the PeerTrust PSM/DTC algorithm [9], where a peer i estimates the trustworthiness of another rater j based on the similarity between i's and j's ratings on sellers both i and j have contacted with. The second model $\mathcal{X}$ uses the maximum likelihood estimation-based learning techniques [4]. A peer i estimates the probability that a rater j is trustworthy so as to maximize the likelihood of getting the current set of ratings from i and j on those sellers they both experienced. We also implemented a dishonesty detector $\mathcal{A}$ with reasonably good misclassification errors: both α, β are less than $\epsilon = 0.1$, and with a high cost of each time being used, which simulates a global trust learning algorithm and is used to verify the relation between the learning accuracy and the cooperation level in the system with peers having the same input when learning the reliability of raters. The two models $\mathcal{L}$ and $\mathcal{X}$ were used to test the efficiency of the seller-selection protocol in environments where peers use different algorithms with personalized inputs and settings to estimate rating behaviors. For the sake of readability, in the following experiments we only show the results for algorithm $\mathcal{L}$. Results for algorithm $\mathcal{X}$ are close to those of $\mathcal{L}$, and results of $\mathcal{A}$ are even better. The use of other global trust learning models like EigenTrust [6] in place of the algorithm $\mathcal{A}$ is subject to our future work.

As a base for other experiments, we first estimated the overall misclassification errors α, β of those implemented trust learning algorithms $\mathcal{L}$ and $\mathcal{X}$ under a variety of scenarios, depending on the fraction of honest reporting users h in the system. The most representative scenarios C_1, C_2, and C_3 for this experiment type are given in Table 1. The results in Fig. 1(a) with the mean number of transactions $\mu_{trans} = 50$ show that the accuracy of the computational model $\mathcal{L}$ is highest in less malicious environments (higher levels of honest reporters and good sellers), as we expected. These $max\{\alpha, \beta\}$ statistics are then used in our later experiments as global knowledge of all strategic peers, where we also observe the same trend of accuracy statistics. We tested with different values of μ_{trans} and observed that when most peers only participate in few transactions ($\mu_{trans} < 10$), the computational model did not have enough sample data for accurate learning, resulting in high errors $\alpha, \beta > 0.5$ and thus such a computational trust model was ineffective in enforcing cooperation.

The cooperation level in different scenarios with various fractions of strategic sellers and raters, which is defined as the fraction of good transactions in the system, are given in Fig. 1(b) for $\mu_{trans} = 50$ (cases C_4, C_5, C_6 of Ta-

[1] http://lsirpeople.epfl.ch/lhvu/download/repsim/

Table 1. Experimental settings of representative simulation scenarios.

Scenario	s	g	b	h	sr	a	b	Result
C_1. No strategic sellers, most seller bad	0	15	85	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1a
C_2. No strategic sellers, half seller good	0	50	50	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1a
C_3. No strategic sellers, most seller good	0	85	15	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1a
C_4. Few sellers strategic, most bad	10	5	85	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1b and c
C_5. Most sellers strategic	85	5	10	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1b and c
C_6. Different seller types	33	34	33	0 to 100	$\frac{100-h}{5}$	$\frac{2(100-h)}{5}$	$\frac{100-h}{5}$	Fig. 1b and c

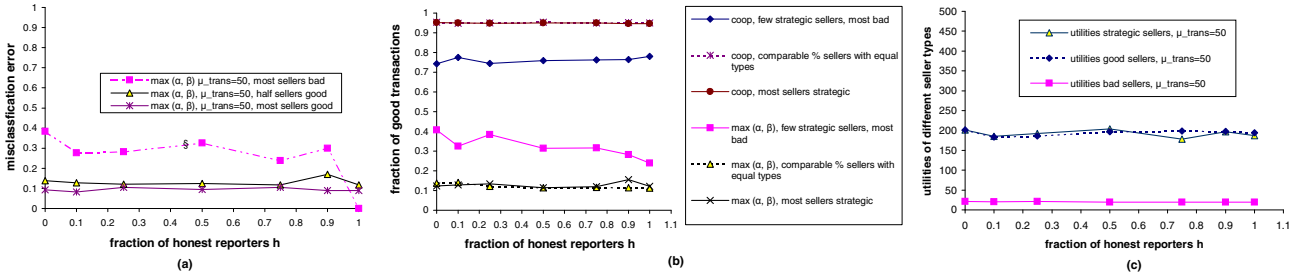

Figure 1. a) the misclassification errors of algorithm $\mathcal{L}$; b) Accuracy of a computational trust model and cooperation level; c) Utilities of different seller types.

ble 1). The case of all strategic sellers showed even a better trend. We also performed experiments with other extreme cases, e.g., all sellers, buyers, and raters strategic, yet the results were similar and thus are not given here. For small $\mu_{trans} < 10$ the cooperation dropped significantly, as the learning had no values at all.

In presence of various rating and serving behaviors (cases C_4, C_5, C_6 of Table 1), accumulated utilities of strategic peers who cooperate in all transactions except the last Δ ones were generally the highest, as Fig. 1(c) shows. Utilities of strategic sellers are close to those of good ones since strategic peers are enforced to cooperate most of their life. Our proposed approach works with dynamic joins and leaves of peers in the system, given that most peers stay in the system long enough ($\mu_{trans} > 10$ in our simulation). Experiments of using a combination of algorithms to minimize the total learning cost (Theorem 2) also gave promising results [8]: (1) the implementation cost was reduced significantly while a high cooperation level was still maintained; and (2) such an approach learned peer behaviors faster and during the same simulation duration, the total number of good transactions in the whole system was much higher.

5. Conclusion

Our study provides a starting point to use reputation information effectively by exploiting both its *sanctioning* and *signaling* roles [2] in decentralized and self-organized systems. The tradeoff between accuracy and cost of such models has also been exploited to minimize the implementation cost of a reputation system in such environments. As part of our future work, we plan to implement and simulate a variety of bounded-rational behaviors, e.g., peers may use various reinforcement learning algorithms to derive their best strategies to follow. Such empirical simulations may give us more insights to the effectiveness of different trust learning algorithms in boosting trust and enforcing cooperation in presence of bounded-rational behaviors. Our ultimate goal is to provide a "cookbook" for applications of different trust learning algorithms depending on the level of rationality present in open and decentralized systems.

References

[1] R. Agrawal and E. Terzi. On honesty in sovereign information sharing. In *Proc. of EDBT'06*, volume 3896, pages 240–256. Springer, 2006.

[2] C. Dellarocas. Reputation mechanism design in online trading environments with pure moral hazard. *Information Systems Research*, 16(2):209–230, 2005.

[3] C. Dellarocas. Reputation Mechanisms. *Handbook on Economics and Information Systems (T. Hendershott, ed.), Elsevier Publishing*, 2005.

[4] Z. Despotovic. *Building trust-aware P2P systems*. PhD thesis, Swiss Federal Institute of Technology Lausanne, Switzerland, 2005.

[5] J. Golbeck. Trust on the world wide web: A survey. *Foundations and Trends in Web Science*, 1(2):131–197, 2006.

[6] S. D. Kamvar, M. T. Schlosser, and H. G. Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of WWW'03*, 2003.

[7] A. Schlosser, M. Voss, and L. Brckner. On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 10, 2005.

[8] L.-H. Vu and K. Aberer. Effective usage of computational trust models in rational environments. Technical Report LSIR-REPORT-2008-007, 2008, available at http://infoscience.epfl.ch/search?recid=125277&of=hd.

[9] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, 16(7):843–857, 2004.