**Technical report LSIR-REPORT-2008-007**

# Effective Usage of Computational Trust Models in Rational Environments

LE-HUNG VU
and
KARL ABERER
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

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 such computational learning models and their ability to effectively enforce cooperation among rational agents as a result of their game-theoretic properties. We provide theoretical results that show under which conditions cooperation emerges when using trust learning algorithms with given accuracy and how cooperation can be still sustained while reducing cost and accuracy of those algorithms.

Specifically, we use a computational trust model as a dishonesty detector to filter out unfair ratings and proved that such model with reasonable false positives and false negatives can effectively build trust and cooperation in the system, assuming the rationality of participants. These results reveal two interesting observations: first, the key to the success of a reputation system in rational environment is not a particular sophisticated learning mechanism but an effective identity management scheme to prevent whitewashing behaviors. Second, in heterogeneous environment where peers use different learning algorithms with certain accuracy to learn trustworthiness of their potential partners, cooperation may also emerge. In other words, different computation trust models produce the same effects on rational participants in enforcing cooperation and building trust among them.

We 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, electronic commerce or community systems.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents; Multiagent systems; I.6.3 [**Simulation and Modeling**]: Applications

General Terms: Algorithms, Economics, Design, Experimentation

Additional Key Words and Phrases: trust, reputation, learning, rationality, incentive-compatibility

## 1. INTRODUCTION

Reputation information has long been shown as an effective tool to enforce cooperation and build trust among participants in a variety of e-commerce systems and online forums such as eBay[1], Yahoo Auction[2], or Slashdot[3].

Researches on the effectiveness and application of reputation information in open and

[1] http://www.ebay.com
[2] http://auctions.yahoo.com/
[3] http://www.slashdot.com/

decentralized environments have also attracted substantial efforts recent years. Depending on the way reputation information is used, current work in the area can be classified into two solution classes [Dellarocas 2005b]: the first includes those *computational trust models* that learn peers' behavior based on their historical performance, e.g., whether and to which extent a peer provide high quality services and gives reliable recommendation on the others. Work in this class mainly exploits the signaling role of reputation: peer learns participants' behavior types from reputation information as a form of social knowledge. The second category of solution comprises those *game-theoretical approaches* that use reputation as a sanctioning tool to enforce cooperation and establish trust among participants, e.g., by penalizing bad peers and praising good ones. The system is designed as a game whose structure ensures that being cooperative is the dominant strategy of all rational peers who want to maximize their life-time expected utilities. Interested readers can refer to existing surveys [Golbeck 2006; Jøsang et al. 2005; Despotovic and Aberer 2006; Dellarocas 2005b] for systematic reviews on the area based on various viewpoints.

Computational trust models propose appropriate statistical or ad hoc heuristic methods to aggregate ratings on past transactions of a peer and related participants, from which to compute a trust metric as indication of a target's trustworthiness. The main principle of such models is to learn and predict the future behaviors of peers by examining their past behavioral patterns, assuming such behaviors follow certain probabilistic models, e.g., peers behaving good in the past are likely to continue doing so in next transactions. For example, a learning peer can estimate the reliability of another as proportional to the similarity between its own experience with ratings reported by the latter [Xiong and Liu 2004]. The EigenTrust model [Kamvar et al. 2003] assigns a unique global trust value to each peer as the PageRank-like value of that peer in a network of individual agents giving recommendations among one another. Alternatively, a peer may assume behaviors of a target to follow a certain probabilistic model, thus it evaluates trustworthiness of the target as the posterior probability the latter provides good services, using ratings from the others as evidences to update the original probabilities [Whitby et al. 2005; Patel et al. 2005]. Such computational methods are designed to be resilient against malicious peers who 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.

On one hand, computational trust models have been intensively studied and demonstrated to be robust under various (strategic) attacks by malicious peers, e.g., ballot-stuffing, badmouthing, or collusion among many participants. In fact, statistical learning models and appropriate heuristics can effectively filter out biased information to obtain a correct picture of the peer's historical quality, as empirical evidences have shown [Despotovic 2005; Anceaume and Ravoaja 2006; Sun et al. 2006]. On the other hand, current computational trust 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 rationally opportunistic (or selfish) peers, who are neither dishonest nor cooperative but change their behaviors strategically to maximize expected life-time utilities, it is still unclear how well a computational trust model can enforce cooperation in the system. As a typical example, strategic peers can first cooperate to build reputation and then start cheating to increase their life-time utilities. Although some simulations [Liang and Shi 2007; Schlosser et al. 2005] suggest that these algorithms may also enforce cooperation in presence of both rational and malicious behaviors, no theoretical analysis has

been performed to justify this.

Game-theoretical trust management approaches of the second solution class mainly deal with rationally opportunistic participants, who have full knowledge of the solution model being deployed and behave strategically to maximize their life-time utilities. Promising solutions include giving monetary incentives to peers to motivate their good service provisioning and truthfully reporting behaviors [Jurca and Faltings 2006; Miller et al. 2005], or by using reputation as a effective sanctioning tool: peers with lower reputation are less likely to be selected, as in [Dellarocas 2005a]. However, these solutions usually rely on many assumptions: peers having specific utility functions, being fully rational with unlimited computational power to find complex equilibriums of the designed mechanism. More importantly, such solutions do not consider malicious behaviors of attackers.

Since both rationally opportunistic and malicious behaviors can be present in real environments, an effective reputation management approach to minimize influences of these unwanted behaviors is of paramount importance. A natural question is *how we can exploit well-tested and accurate computational trust models to effectively establish trust among partners in environments where peers may exhibit various behaviors*, including honest, malicious, and rationally opportunistic. We want to know whether and under which conditions, computational trust models that are resilient against malicious attackers can also be used in a incentive-compatible way to motivate cooperation of rationally opportunistic players. As the main performance criteria of such models are their statistical accuracy in detecting malicious behaviors, it is our interest to study the relation between such statistical accuracy measures and their ability to enforce cooperation among peers and discourage their selfish behaviors.

Another question is related to the optimized usage of computational trust models, as such methods are costly to deploy and maintain. To effectively learn of bad behaviors and minimize their influences, various cost is incurred to retrieve recommendations, filter out biased information, evaluate and update reputation scores of peers [Liang and Shi 2007], etc. In fact, given the rationality of peers, it may be unnecessary to always use accurate yet costly learning algorithms to encourage truthful behaviors. The reason is rationally selfish peers make use of their knowledge about the deployed algorithm(s) to avoid having bad reputation and maintain their high benefits from the system. Being aware of the existence of such learning algorithms to reliably detect bad behaviors, peers have little incentives to cheat and thus expensive learning can be avoided. Consequently, accurate yet costly trust learning mechanisms should only be used as an inspection tool to detect past cheating behaviors, in order to punish bad agents appropriately, e.g., by not selecting them for later transactions. Such punishment meted out to peers found cheating can be used to provide sufficient incentives for cooperation of peers. Exploiting this fact, we want *to minimize the cost of using expensive computational trust models* in environments where most peers are rational. A solution to this question implies many benefits for various applications, e.g., where peers have limited resources and quick decisions are generally preferred to avoid missing opportunities, especially in competitive scenarios.

In this paper, we propose and analyze a simple but effective way of using a computational trust model to minimize influence of malicious peers and at the same time, keep rational peers being cooperative during most of their transactions. Specifically, we propose a peer selection mechanism that consists of two steps. First, we use a computational trust model to estimate the reliability of the most recent rating on a peer, based on the

rating's credibility of the rater and the trustworthiness of the peer being evaluated. Well-experimented computational trust models, e.g., [Xiong and Liu 2004; Patel et al. 2005] can be used to identify malicious rating behaviors with high accuracy in this step. The result of the learning is then used to decide whether to include the target peer for selection or to ignore/blacklist the peer being rated. Such selection approach can be applied easily in various open e-markets or online communities with different degree of centralization. Our theoretical analysis and extensive simulation to analyze the efficiency of this peer selection protocol under various scenarios provide the following results and contributions:

- **sufficient conditions on statistical accuracies of a trust learning algorithm to enforce cooperation**: we prove that if statistical accuracy measures of the chosen learning algorithm in the first step are good enough, rational peers find it most beneficial to cooperate in all but some of their last transactions. These conditions also show that in rational environments, even simple and naive algorithms may lead to a good social outcome with high cooperation level in the system. According to extensive simulations, such results are still applicable if peers join and leave dynamically, provided most of them stay long enough. 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 trust learning algorithm being used. By showing that there exist certain sufficient conditions so that a computational trust model can be used effective in boosting cooperation, we provide 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 fully 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.
- **a cost-efficient reputation management approach**: we propose a way to best combine existing trust learning algorithms to maximize cooperation in the system, while minimizing cost, if there are alternative computational trust models with cost/accuracy trade-off among them. Inspired by a inspection game-theoretic approach [Avenhaus et al. 2002], we prove that during the step of evaluating rating reliability, it is sufficient to use an accurate (and expensive) 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.

In a larger extent, our presented selection protocol establishes an umbrella framework to use reputation information effectively in decentralized and self-organized systems by exploiting both its *signaling* and *sanctioning* roles [Dellarocas 2005a].

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. The most related work is by Dellarocas [Dellarocas 2005a] that studies some design parameters of a reputation system and its effectiveness. However, this work does not include the analysis of influences of malicious behaviors and the optimization of the cost of the reputation mechanism being deployed. Our work analyzes the impacts of these factors, comprises extensive simulations on a variety of scenarios, and is also applicable for any open systems with different degree of centralization.

Section 2 of this paper gives a formal definition of our system model. In Section 3 we study the relation between accuracy of a computational trust model and its incentive-compatibility with a detailed theoretical analysis. We then propose a way to combine different trust models to minimize their usage cost while retaining their efficiency in enforcing cooperation among rational participants in Section 4. Section 5 proposes an approximate approach to use computational trust models in scenarios with peers joining and leaving dynamically overtime. Our simulation and experimental results are presented in Section 6. We summarize related work in Section 7 before giving some discussions and concluding the paper in Section 8.

## 2. SYSTEM MODEL

### 2.1 Scenario and Assumptions

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. Denote $P$ the set of all peers and $W : P \times P \rightarrow D_f$ the social relationship among them, where $D_f$ is the domain to represent the relationship values between two peers. For example, $D_f$ may represent the relationship type between two friends in a social network (famlily/close/normal/stranger) or the weight of an edge between two nodes in a trust network[4].

The set of outcomes of a transaction between peers is $\mathcal{O}$. In this work, we consider $\mathcal{O}$ as a binary set corresponding to a bad or a good outcome, yet this can be generalized to any set with a total ordering among the elements.

A peer provides a resource with value $u$ for its clients, where $u$ lies within a range of minimal price $u_*$ and maximal price $u^*$. Thus we name $u$ the "legal" payoff of a peer in a transaction if he or she behaves honestly in providing a resource and yields a good transaction outcome. Example honest behavior is to ship the item to the buyer after receiving payment, or to provide a good service to the client. If the provider peer cheats, e.g., doesn't ship the good or provides the low-quality services, it gains a further "illegitimate" amount $v$, where $0 \leq v \leq v^* < \infty$. For example, in an e-trading system, this $v$ can be the shipping cost plus the value of the item as evaluated by the seller. In this case, the transaction is considered having a bad outcome.

Denote $T(x, y, t)$ a transaction between two peers $x \in P, y \in P$, where $x$ is the buyer (a client) of the resource, $y$ is the seller (the server or the provider), and $t \in \Omega$ is the timestamp of the transaction. For convenience, let $T_s(y) = \{T(x, y, t) \mid x \in P, t \in \Omega\}$ be the set of all transactions with $y$ as the seller of the resource (or server). Similarly, $T_c(x) = \{T(x, y, t) \mid y \in P, t \in \Omega\}$ is the set of all transactions where the peer $x$ plays the role of the buyer.

We use $r_s(x, y, tr)$ to represent a rating from a peer $x$ on another peer $y$ on a transaction $tr \in T(y)$. Generally, peers may also need to rate the reliability of a rating, e.g., a seller reports that a certain rating on one of the transaction is unfair. We use another notation $r_r(x, y, r)$ for represent the rating of a peer $x$ on a specific rating $r$ made by $y$. The value of a rating $r_s(x, y, tr)$ or $r_r(x, y, r)$ belongs to the outcome domain $\mathcal{O}$.

Our goal is to maintain a high level of cooperation in our decentralized application environment: all providers/sellers mostly behave honestly, and clients/buyers are motivated

[4]http://trust.mindswap.org/

to leave truthful (reliable) ratings after their transactions.

Whereas the problem of trust and cooperation is ubiquitous and may be present cross all layers of the system, in this paper we focus on the cooperation among participants in the application layer. We assume the existence of a (decentralized) storage system that supports efficient search for resources, ratings, and related information on participants. Such storage system should ensure that advertisements of provided services, published ratings, and transaction information are authentic and can not be tampered with. That is, peers can neither fake transactions nor modify feedback data submitted by others. These goals can be met by using a DHT-based storage [Aberer et al. 2003] and cryptographic tools such as digital signatures based on a decentralized PKI infrastructure [Datta et al. 2003].

We also assume that peers typically know the lower bound of prices of a service (or a resource) $u_*$ and they can estimate the gain $v$ of a bad provider peer after each transaction, for example, in eBay-like environment, is the shipping cost plus the item value. Such an assumption is realistic: a centralized system can define those values $u_*$ for each category of services or items, or peers can learn these values by looking at trading history of other peers in the system.

The final assumptions are that peers are generally long-term players and stay in the system infinitely, or at least, long-enough. This assumption can be relaxed under certain circumstances, e.g., in centralized environments where it is possible to impose an entrant fee for a newly joined peer.

The above environment is an abstraction of many practical P2P application scenarios with different degrees of centralization, where participants are rational to a certain extent. Such scenario can represent, for example, a centralized eBay-like trading environment, a social-network based system for selling and buying goods, or a decentralized markets of services [Papazoglou and Georgakopoulos 2003; Buyya et al. 2001]. Consequently, our proposed solution can be used in all these applications. Similarly, the assumptions above to simplify our analysis are also well-accepted ones in the trust-related research literature for analyzing peer's behaviors [Despotovic 2005].

### 2.2 Example Application

Given the above general notations, we introduce the following running example to illustrate different concepts and results of our work. We consider a decentralized (also called a C2C or peer-to-peer) market of goods, where each participant can be seller and/or buyer of certain good items. As a concrete example, any person in the Internet can advertise and sell their items in a peer-to-peer system and/or an online social network like Facebook[5] or MySpace[6]. Another realistic showcase of this is the recent launch of Neighborhoods[7] that enables eBay users to do shopping via their social networks.

In such $C2C$ trading scenario, buyers can search and buy available items that match their needs and interests. As a common rule, a buyer has to pay for the item first. Only after receiving the payment, the seller decides to ship or not to ship the item to the buyer. If the seller cheats (doesn't ship the good or provides the low-quality item), it gains a further amount $v$, where $0 \leq v \leq v^* < \infty$. This value $v$ can be the shipping cost plus the value of

[5]http://www.facebook.com
[6]http://www.myspace.com
[7]http://neighborhoods.ebay.com

the item as evaluated by the seller.

The traditional way to enforce the cooperation of the seller via reputation mechanism is to allow a buyer to rate a seller as *honest* or *cheating* after finishing a transaction with him [Resnick et al. 2000]. Other buyers can search for available recommendations or ratings on the seller and decide whether they should go into transaction with this seller. In this case, a computational mechanism to effectively eliminate unreliable ratings in order to minimize influences of malicious rating users are strongly required. Other incentive mechanism to elicit sufficient truthful reports from buyers for such computational mechanism or trust learning algorithm are also necessary. Note that we do not consider certain auction schemes that can also be deployed for selection of an appropriate buyer, yet this issue is orthogonal to our concerns and thus outside the scope of the paper.

For brevity and readability reason, we use the notion of a buyer, a seller, or a good item to illustrate our concepts. These notations also mean, respective, to a service provider, a client, or a service offered by a peer in other application scenarios.

### 2.3 Computational Trust Models as Dishonesty Detectors

A computational trust model relies on various statistical or heuristic methods to learn behavior of a peer (the target) based on various information sources. The first is the performance statistics of the target in past transactions, both via recommendations/ratings from previous partners and via personal experience of the learning peer on the target. Other information includes intrinsic features of the target itself, e.g., frequencies of posted ratings and involved transactions, location of the raters [Cornelli et al. 2002], relationships with other peers in the systems [Ashri et al. 2005]. The behaviors that can be learned from such information include both serving and rating behavior of any peer in the system. The former represents the likelihood that the peer offers a high quality resource to its clients, e.g., whether a seller ships the item to buyer after receiving the payment. The latter means the trustworthiness of the peer in rating a transaction, that is, whether the peer truthfully reports its experience.

In this paper, we use a computational trust model as a dishonesty detector to effectively evaluate the trustworthiness (or reliability) of a rating, similar to a conventional spam detector in machine learning literature. From this perspective, we can formally model such mechanism as in Definition 2.1 (see Section 2.1 for defined concepts).

**Definition** 2.1. A *reputation-based* and possibly *personalized computational trust model* used by a peer $i$ to estimate the trustworthiness of the rating by another peer $j$ is a 5-tuple $\mathcal{R} = \langle P_i, V_i, \mathcal{F}_j, A, D \rangle$ where:

- $P_i \subseteq P$ is a set of peers that $i$ considered relevant to the evaluation of $j$'s rating reliability.
- $V_i \subseteq \{r_s(x, y, v) \mid x \in P_i \wedge y \in P_i\} \cup \{r_r(x, y, v) \mid x \in P_i \wedge y \in P_i\}$ is subset of all ratings related to peers in $P_i$.
- $\mathcal{F}_j$ is a set of properties of the target peer $j$, for instance, its location, frequencies of posted ratings, number of involved transactions, etc.
- $A$ is a function that operates on $P_i, W, V_i, \mathcal{F}_j$ and outputs a trust value $T_{ij}$. The value $T_{ij}$ is understood as the trustworthiness of $j$ in rating. $T_{ij}$ may take binary, real, or discrete values.
- $D$ is a set of decision rules with binary outcome stating whether $i$ should trust $j$ given the (personalized) trust metric $T_{ij}$.

Definition 2.1 is the formal description of several (if not most) well-known heuristic or statistical trust evaluation algorithms in the literature. This formal model includes all approaches studying trust on peer to peer systems [Despotovic and Aberer 2006], on social networks [Golbeck 2006], in e-commerce systems, and on other online communities [Dellarocas 2005b; Jøsang et al. 2005].

The set of relevant peers and related ratings $P_i$ and $V_i$ constitutes the actual feedback retrieval mechanism of the computational trust model and contributes the main cost of the mechanism, as we will explain later on. To illustrate the expressiveness of the formal model in Definition 2.1, Example 2.2 specifies the well-known heuristic trust learning proposed by [Xiong and Liu 2004].

**Example** 2.2. The personalized trust model PeerTrust [Xiong and Liu 2004] can be seen as a tool to evaluate rating reliability via the following formal modeling:

- $P_i = \{i, j\} \cup \{k \mid k \in T_s(i) \wedge T_s(j)\}$. In other words, the set of relevant peers $P_i$ includes are $i, j$ and those peers having interactions with both of them.
- $V_i = \{r_s(x, y, v) \mid x \in \{i, j\}, y \in P_i\}$, that is, $V_i$ consists of those ratings by $i$ and $j$ on serving behaviors of other peers $k$ in the relevant peer set $P_i$.
- the relationship $W$ among peers and the features of the target $\mathcal{F}_j$ are not considered.
- $T_{ij} = 1 - \sqrt{\frac{\sum_{k \in P_i} d^2[r_s(i,k,v), r_s(j,k,v)]}{\|P_i\|}}$, where $d[u, v]$ is the difference between two ratings by $i$ and $j$, which can be measured as the Euclidean distance between their respective rating values $r_s(i, k, v)$ and $r_s(j, k, v)$.
- The ratings by $j$ are considered as reliable if $T_{ij} > T_{min}$ and as unreliable otherwise, where $T_{min}$ is a system design parameter.

**Example** 2.3. Another approach to estimate peer's trustworthiness is to assume peers to behave according to a probabilistic model [Despotovic and Aberer 2004]. From the viewpoint of the learning peer $i$, each target peer $j$ has a probability $T_{ij}$ of reporting truthfully what its observes. This model is specified similar to Example 2.2, except the definition of $T_{ij}$ and the decision rule $D_i$ to estimate rating reliability. We formally define:

$$T_{ij} = \frac{\sum_{k \in P_i} I(r_s(i, k, v) = r_s(j, k, v))}{\|\{r_s(i, k, v) \mid k \in P_i\}\|} \tag{1}$$

where the function $I(c)$ evaluates to 1 if the boolean condition $c$ is true. Thus $T_{ij}$ is defined by the fraction of ratings by $j$ having the same values as ratings by $i$. This $T_{ij}$ is an estimate of the probability the peer $j$ being honest when rating, and such estimate maximize the likelihood of having the observation set $V_i$ by the set of relevant peers $P_i$. In this case, the decision rule $D_i$ to estimate rating reliability is to trust the rating with probability $T_{ij}$.

Other computational trust models can be represented similarly. Global trust metrics like EigenTrust [Kamvar et al. 2003] and complaint-based algorithm [Aberer and Despotovic 2001] consider all peers in the networks, thus $P_i = P$, and consider the recommendation/ratings among each pair of them.

Considering a computational trust model as a dishonesty detector, we define its statistical accuracy measures similar to those of a conventional spam detector in machine learning literature (Definition 2.4).

**Definition** 2.4. The *accuracy of a computational trust model* $\mathcal{R}$ formulated in Definition 2.1 in estimating the reliability of a rating is defined by its two *misclassification errors*, $\alpha$ and $\beta$, 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. We refer to $\alpha$ and $\beta$ respectively as false positive and false negative errors henceforth.

The accuracy of a computational model also implies its resilience to possible malicious attacks that manipulate ratings, by estimating the trustworthiness of a rating using the combination of performance statistics of both the rater and the seller being considered to estimate whether a rating is trustworthy.

## 2.4 Strategic Seller-Selection Protocol

We propose the following approach for a a rational buyer to select a seller among candidates (Definition 2.5). Concrete implementation of this mechanism will be presented later on in Section 5. Table I summarizes most frequently used notations in this section.

**Definition** 2.5. A buyer uses the following *seller-selection protocol* $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ to evaluate the eligibility of a seller 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 with 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.

The above seller-selection protocol $\mathcal{S}_k$ is tough for bad players, including malicious and rationally opportunistic sellers. It assures that a globally blacklisted seller has no further chance to do transactions with other peers. The evaluation of rating reliability by a computational trust model in step (2) aims to reduce influences of strategic manipulation of ratings by rational or malicious peers. The goal here is to eliminate as much malicious sellers as possible when they start cheating and incentivize rationally opportunistic sellers to cooperate. Actually, the use of the above seller-selection protocol with a global computational trust model mimics the behavior of a centralized reputation system in practice.

The parameter $k \geq 1$ represents the cautiousness of a buyer in trusting cheating detections published by others and maybe different for each buyer. The incentives for sharing the learning results are not much an issue: in case all buyers do not share the learning results, a buyer can still do the learning by itself and our approach still holds if the evaluation of rating reliability is deterministic, e.g., $\mathcal{R}$ is a deterministic global computational trust model. Generally the learning at step (2) is verifiable, e.g., in case of global computational trust models such as EigenTrust [Kamvar et al. 2003] or complaint-based [Aberer and Despotovic 2001], writing wrong learning results is detectable and not an issue.

Thus the main problem is the badmouthing attack, when many peers collude to badmouth a certain seller to eliminate it. To reduce the effect of this attack, a robust algorithm $\mathcal{R}$ should consider using trustworthiness of both rater and the seller being rated to estimates

Table I. Commonly used notations

| Notation | Definition |
|---|---|
| $u_*$ | minimal price of offered items/resources/services, $u_* > 0$ |
| $u^*$ | maximal price of offered items/resources/services $u^* \leq u_*$ |
| $u$ | legal gain of seller when cooperating, $u_* \leq u \leq u^*$ |
| $v$ | additional gain of seller when cheating in a transaction, $v > 0$ |
| $v^*$ | maximal additional gain of seller by cheating, $0 \leq v_* < \infty$ |
| $\alpha$ | Pr(rating estimated as reliable \| unreliable rating) |
| $\beta$ | Pr(rating estimated as unreliable \| reliable rating) |
| $k$ | # of negative learning results posted on a seller before globally blacklisting it |
| $\epsilon$ | upper-bound of $\alpha$ and $\beta$, $0 \leq \epsilon \leq 1$ |
| $\Delta$ | # of remaining transactions for which a rational seller does not have incentives to cooperate, $\Delta \in \mathbb{N}$ |
| $\mathcal{R}$ | a computational trust model to estimate reliability of a rating, explained in details in Definition 2.1 |
| $\mathcal{N}$ | the naïve computational model evaluating all ratings as reliable |
| $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ | a seller-selection protocol specified in Definition 2.5 |

whether a rating is reliable. Anyway, the accidental blacklisting of a good peer is unharm to a buyer and can be reduced by increasing $k$.

Note that in the designed seller-selection protocol, the actual computational trust models being used and its settings may be private information of the learning. Only the theoretically or experimentally-proved bound of the misclassification errors of these models are shown to the participants in the system.

We consider the absence of a rating after a transaction as a positive one to reduce the effect of having less ratings for making decisions. The incentives to leave truthful feedback after a transaction in the system is another issue we do not consider in details in this work. In fact, it is possible to integrate existing incentive mechanisms, e.g., via side-payment [Miller et al. 2005] into our system without any major efforts. Furthermore, buyers have other indirect incentives to leave reliable ratings after transactions. First, such behaviors help to eliminate bad sellers. Second, future sellers are motivated to cooperate with honest reporting buyers as shown in a later analysis (Corollary 3.7), which also give additional motivation for peers to leave truthful feed back.

## 3. INCENTIVE-COMPATIBILITY OF COMPUTATIONAL TRUST MODELS

In this section, we study the relation between the accuracy of a computational trust model and its incentive-compatibility, that is, the possibility to effectively use a computational model with reasonably good accuracy for boosting trust and cooperation in decentralized environments.

### 3.1 Quantifying the Learning Accuracy

Misclassification errors of a given computational model depend on several factors, both endogenous and exogenous. Endogenous factors come from the computational model itself, most importantly are input-related factors (personal experience of the learning peer, collected ratings) and algorithmic issues (underlying probabilistic model, feedback retrieving

and aggregation strategies, etc.). Exogenous factors include behaviors of participants (malicious, rationally opportunistic, or voluntarily cooperative) and system dynamics (such as the volumes of transactions and participating levels over time).

Practically, a bound on computational model accuracy can be measured either experimentally or via theoretical analysis. This question has already been extensively studied in several previous work, namely [Levien 2002; Xiong and Liu 2004; Kamvar et al. 2003], most of which having been shown to have low $\alpha, \beta$ under various attack scenarios. Such studies confirm that the main factor influencing the errors $\alpha, \beta$ of a computational trust model is the overall fraction of honest raters in the system.

**Example** 3.1. Consider the simple computational model $\mathcal{N}$ that uses a naively optimistic algorithm: trust all ratings and consider the absence of a rating as a positive one. We claim that misclassification errors of $\mathcal{N}$ are $\alpha = \alpha_0 = 1$ and $\beta = \beta_0 = 0$. In fact, let $0 \leq \delta_h, \delta_l, \delta_i \leq 1$, where $\delta_h + \delta_l + \delta_i = 1$ be respectively the probabilities that the rating peer provides a reliable rating, an unreliable one, and no rating after a transaction with a specific seller at the time of estimate. There are two possibilities:

- the seller did cooperate in the last transaction, thus the absence of a rating can be considered a reliable positive rating. So $\alpha_0 = Pr(est+, real-)/Pr(real-) = \delta_l/\delta_l = 1$ and $\beta_0 = Pr(est-, real+)/Pr(real+) = 0/(\delta_h + \delta_i) = 0$.
- the seller did not cooperate in last transaction, thus the absence of a rating is an unreliable positive rating. Still, we have $\alpha_0 = Pr(est+, real-)/Pr(real-) = (\delta_l + \delta_i)/(\delta_l + \delta_i) = 1$ and $\beta_0 = Pr(est-, real+)/Pr(real+) = 0/\delta_h = 0$.

In our work, we generally do not care about actual values of those errors $\alpha, \beta$ but only their upper-bound $\epsilon$, and we are concerned with those computational models with accuracy better than random guess, i.e., $\epsilon < 0.5$. Several measurements on $\alpha, \beta$ of the computational trust models mentioned in Examples 2.2 and 2.3 under different conditions also confirm the accuracy of such mechanisms in learning the reliability of ratings (c.f. Section 6.4).

Our approach supposes that such bound of $\alpha, \beta$ of trust models being used is provided to all peers in the system. Alternatively, these errors can also be learnt by participants themselves. Although a proof for the existence of a trust model with low $\alpha, \beta$ is out of the scope of the paper, an initial positive answer is available [Anceaume and Ravoaja 2006].

In practice, accurate learning with $\alpha, \beta$ upper-bounded by some $\epsilon < 0.5$ is generally possible in most application scenarios. For instance, an accurate yet expensive method to estimate reliability of ratings by peer on a seller is to perform full monitoring on performance of the seller to learn its real *past* behavior. Such monitoring can be implemented in many ways: in an e-commerce system, monitoring is done via legal investigations on suspicious transactions conducted. In a market of web services, one can deploy monitoring agents to periodically probe and test the service being offered by a provider to estimate the real offered quality level offered during last period by that provider.

## 3.2 Learning Accuracy and Incentive-compatibility

Suppose that sellers have a minimal legal gain $u_*$, maximal illegitimate cheating gain $v^*$, and their possible strategies at each transaction are either cooperate or cheat. Given a bound $\epsilon$ of $\alpha, \beta$ of computational trust model(s) being used by buyers in the system, Theorem 3.2 shows the relation between the errors $\alpha, \beta$ of a computational trust model and its effectiveness in enforcing cooperation of a seller during its life-time.

THEOREM 3.2. *The seller-selection protocol $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ makes it optimal for a rational seller to cooperate in all but its last $\Delta$ transactions, where $\Delta = max\{1, \lfloor v^*/(u_*((1-\epsilon)^k - \epsilon^k))\rfloor\}$. This hold even in presence of malicious and strategic manipulation of ratings by sellers, provided that the trust model $\mathcal{R}$ has misclassification errors $\alpha, \beta$ upper-bounded by some $\epsilon < 0.5$.*

PROOF. Rational sellers apparently do not find incentives to cooperate in a last transaction. Let us consider those rational sellers staying in the system for $\Delta > 1$ transactions more after the current one.

Denote $U$ the current (accumulative) utilities of a rational seller and $u_h$ its best (maximized) expected utilities for remaining time in the system if it is not globally blacklisted after the current transaction. Since the fully cooperative strategy during $\Delta \geq 1$ transactions leads to a total utility of at least $\Delta u_*$, it follows that $u_h \geq \Delta u_*$ for any seller who wants to be involved in at least $\Delta$ further transactions.

Let $0 \leq h, l, s, i \leq 1$ respectively be the probabilities that the current buyer exhibits one of the following rating behaviors after the transaction: honest (provides reliable ratings), advertising (posts positive ratings on the seller), badmouthing (rates the seller negatively), and a non-participating user (does not leave any rating), where $h + l + s + i = 1$.

The probabilities that an honest seller obtains a positive or a negative rating after a transaction are correspondingly $h^+ = h + s + i = 1 - l$ and $1 - h^+$. Thus the probability that the seller will be blacklisted by a next buyer is: $x_b = h^+\beta + (1 - h^+)\alpha = (1 - l)\beta + l\alpha \leq \epsilon$, since $0 \leq l \leq 1$ and $0 \leq \alpha \leq \epsilon, 0 \leq \beta \leq \epsilon$. The probability the seller is globally blacklisted after the current transaction is then $x_b^k \leq \epsilon^k$. This inequality holds even in presence of malicious or strategic manipulation of ratings by sellers, making $h, l, s, i$ different for each learning, provided that misclassification errors $\alpha, \beta$ of $\mathcal{R}$ are less than $\epsilon$.

Similarly reasoning, if the seller is cheating in this transaction, the probability it obtains a positive rating is $l^+ = s + i = 1 - h - l$, with probability $1 - l^+$ such a seller receives a negative rating. In this case, the seller will be blacklisted by a future buyer with probability $y_b = l^+(1-\alpha) + (1-l^+)(1-\beta) = (1-h-l)(1-\alpha) + (h+l)(1-\beta) \geq 1 - \epsilon$. Thus the probability the seller is globally blacklisted is $y_b^k \geq (1-\epsilon)^k$.

Let $U_{honest}$ (and $U_{cheat}$) be the best (maximal) expected life-time utilities of the seller if it is honest (respectively cheating) in the current transaction, it follows that:

$$\begin{aligned} U_{honest} &= U + u + u_h(1 - x_b^k) \\ U_{cheat} &= U + (u + v) + u_h(1 - y_b^k) \\ \delta_{hc} &= U_{honest} - U_{cheat} = -v + u_h(y_b^k - x_b^k) \\ &\geq -v^* + u_h((1-\epsilon)^k - \epsilon^k) \end{aligned}$$

Since $u_h \geq \Delta u_* > 0$ and $0 \leq \epsilon < 0.5$, the condition $\Delta \geq v^*/(u_*((1-\epsilon)^k - \epsilon^k)) > 0$ implies $\delta_{hc} \geq 0$. Therefore, a rational seller considers cooperation as the dominant strategy in any transaction except in its last $max\{1, \lfloor v^*/(u_*((1-\epsilon)^k - \epsilon^k))\rfloor\}$ ones. □

We can make use of the following observation in case we only know the illegitimate gain $v$ of a seller in a transaction, but not the maximal value $v^*$ (Proposition 3.3).

PROPOSITION 3.3. *The seller-selection protocol $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ makes it optimal for a rational seller to cooperate in the current transaction if the seller stays for further $\Delta$ transactions, where $\Delta = max\{1, \lfloor v/(u_*((1-\epsilon)^k - \epsilon^k))\rfloor\}$.*

The above theoretical analysis also holds for the case peers use different computational trust models with different inputs and personalized settings to evaluate the trustworthiness of the last rater, 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 above analysis is straightforward.

According to Theorem 3.2, if the temporary gain $v^*$ is very high, such as the case of selling of expensive items, the parameter $\Delta \to \infty$, meaning that enforcing the cooperation of a rational seller in such a transaction is impossible, which is an intuitive result. In other cases, a relation between $\Delta$ and the error upper-bound $\epsilon$ can be drawn. For example, Fig. 1 shows the relation between $\Delta$ and $\epsilon$ for different values of $k$ and $v^* = u_*$, e.g., peers sell and buy items of comparable prices. We have the following observations: if sellers are long-term players staying in the system infinitely, the number of last $\Delta$ 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. 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 the parameter $\Delta$, i.e., to ensure cooperation of sellers in most transactions.

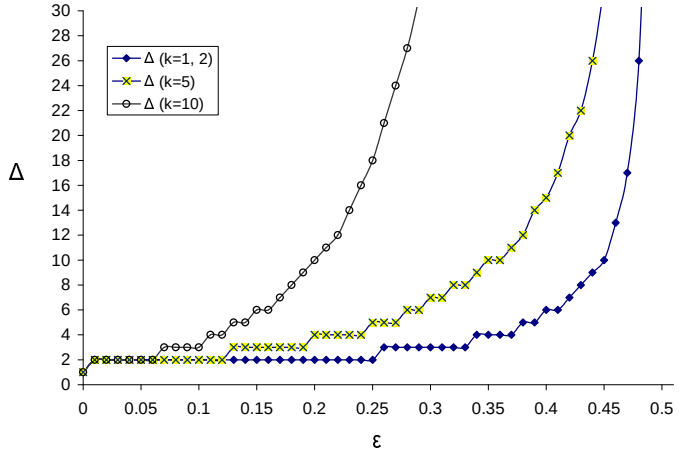


Fig. 1. The relation between the upper-bound $\epsilon$ of misclassification errors of a computational trust model and incentives of rational sellers to cooperate for different values of $k$. Rational sellers find it most beneficial to cooperate during all but their last $\Delta$ transactions.

Consequently, the study of the selection protocol $\mathcal{S}_k = \langle \mathcal{R}, k \rangle$ gives us the following observation on those sufficient conditions to use a computational trust model $\mathcal{R}$ to effectively boost cooperations among rational entities (Corollary 3.4).

COROLLARY 3.4. *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.*

Taking the rationality of peers into account, Corollary 3.4 also reveals 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 $\Delta$ transactions is to require each seller to deposit an approximate amount $\Delta v^*$ to the third trusted 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.

Corollary 3.5 shows the possibility of enforcing cooperation of rational sellers in another case of the seller-selection protocol $\mathcal{S}_k$ with $k = 1$ in case the computational trust model being used has errors $\alpha, \beta$ depending on the fraction of honest reporting users $h$.

COROLLARY 3.5. *The seller-selection protocol $\mathcal{S}_1 = \langle \mathcal{R}, 1 \rangle$ makes it optimal for a rational seller to cooperate in all but its last $\Delta$ transactions, where $\Delta = max\{1, \lfloor v^*/(u_*(1-\alpha-\beta-(\beta-\alpha)h))\rfloor\}$, $h$ is the overall fraction of honest raters in the system, and the trust model $\mathcal{R}$ has corresponding misclassification errors $\alpha, \beta$ such that $\alpha+\beta+(\beta-\alpha)h < 1$.*

PROOF. We proceed as in the proof of Theorem 3.2 with $k = 1$ and note that $y_b - x_b = 1 - \alpha - \beta - (\beta - \alpha)h$. □

We can also make use of another observation (Proposition 3.6).

PROPOSITION 3.6. *The seller-selection protocol $\mathcal{S}_1 = \langle \mathcal{R}, 1 \rangle$ makes it optimal for a rational seller to cooperate in the current transaction if it stays for $\Delta$ further transactions more, where $\Delta = max\{1, \lfloor v/(u_*(1-\alpha-\beta-(\beta-\alpha)h))\rfloor\}$ and $v$ is the illegitimate gain of the seller in current transaction.*

Fig. 2 shows the number of last $\Delta$ transactions in which a rational seller might not find the incentive to cooperate versus the probability it will be blacklisted given misclassification errors $\alpha, \beta$ of a learning algorithm (with different values of $v^*, u_*$).

### 3.3 Incentive-compatibility of the Naive Computational Model

We want to further consider one very simple case of the naively optimistic trust model $\mathcal{N}$: a seller always trusts any rater and considers the absence of a rating as the presence of a reliable positive one. Corollary 3.7 shows the relation between the incentive-compatibility of such naively optimistic algorithm $\mathcal{N}$ and the buyer's truthfully reporting probability $h$, in case there is no strategic manipulation of ratings in the system. The seller-selection protocol $\langle \mathcal{N}, 1 \rangle$ for this special case is actually equivalent to the reputation system considering only the last rating studied by Dellarocas [Dellarocas 2003]

COROLLARY 3.7. *With the exception of strategic manipulation of ratings by a seller, the seller-selection protocol $\langle \mathcal{N}, 1 \rangle$ makes it optimal for a rational seller to cooperate in all transactions but its last $\Delta = max\{1, \lfloor v^*/(hu_*)\rfloor\}$ ones, where $h$ is the probability that a buyer leaves an honest rating after a transaction. Such selection mechanism also gives direct incentives for long-term buyers to leave truthful ratings after their transactions.*

PROOF. Proceed as in the analysis of Theorem 3.2, knowing that the simple computational model $\mathcal{N}$ has misclassification errors $\alpha = 1$ and $\beta = 0$, we have $\delta_{hc} \geq$

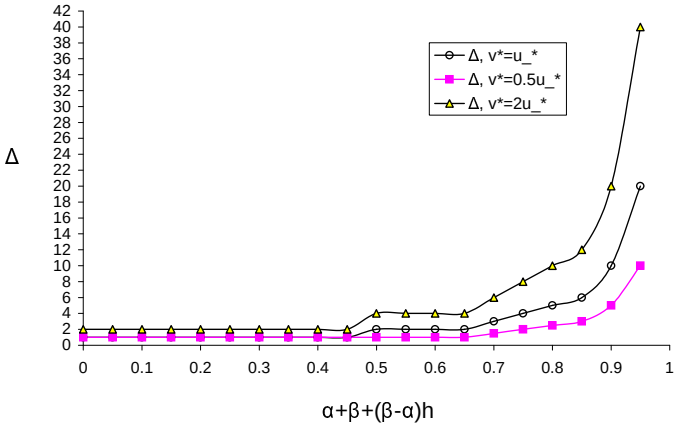


Fig. 2. The relation between the accuracy of the algorithms and the incentives of a rational seller to cooperate. Rational sellers find it most beneficial to cooperate during its whole life-time, except the last $\Delta$ transactions.

$-v^* + \Delta u_*(1-(1-h)) = -v^* + \Delta u_* h$. Thus seller would cooperate with a buyer if $\delta_{hc} \geq 0$, or $\Delta \geq v^*/(hu_*)$. Naturally, the above condition also gives direct incentives for a long-staying buyer to leave a correct rating after its transaction to increase his probability value $h$ as observed by subsequent sellers, which maximizes his chance of having successful transactions in the future (decrease $\Delta$ of other sellers). □

The seller-selection protocol $\langle \mathcal{N}, 1 \rangle$ implements a mechanism for which buyers and sellers follow a tit-for-tat strategy: who behave honestly (reports reliably or cooperates when selling) will be rewarded accordingly. The disadvantage of such mechanism is its exclusion of the case a seller manipulates his ratings. We expect the truthfulness reporting probability $h$ to be affected by preferences of a rating user and noises in his observations: whether he wants to leave a rating and according to his erroneous evaluation of the seller behavior, e.g., due to possible shipping delays or loss.

In the presence of (strategic) manipulation of the reports, a buyer must use a sophisticated trust learning algorithm to evaluate the reliability of the rater, as presented in the previous Section 3.2. Otherwise a selection protocol based on only the latest rating can be attacked easily. For example, a cheating seller can collude with another buyer to immediately stuff a positive rating with newer time-stamp in the system.

Fig. 3 shows the relation between the overall probability $h$ that a peer is a honest reporting peer and the number of $\Delta$ last transactions for which sellers may not find incentives to cooperate, with different settings of $v^*$ and $u_*$. For example, let us consider the previous trading scenario where peers sell and buy items of comparable prices ($u_* \simeq v^*$). If seller is a long-term player, and the truthfully reporting probability $h \simeq .6$ (a highly vulnerable environment), even the naive algorithm makes it optimal for a seller to behave honestly for most of his transactions (except the last one).

It is important to note that Theorem 3.2 and its corollaries (Corollary 3.5 and 3.7) state those *sufficient*, not *necessary* conditions for a trust learning algorithm to be incentive-compatible. In general, there maybe other designs better than the proposed selection mechanism $\mathcal{S}_k$.

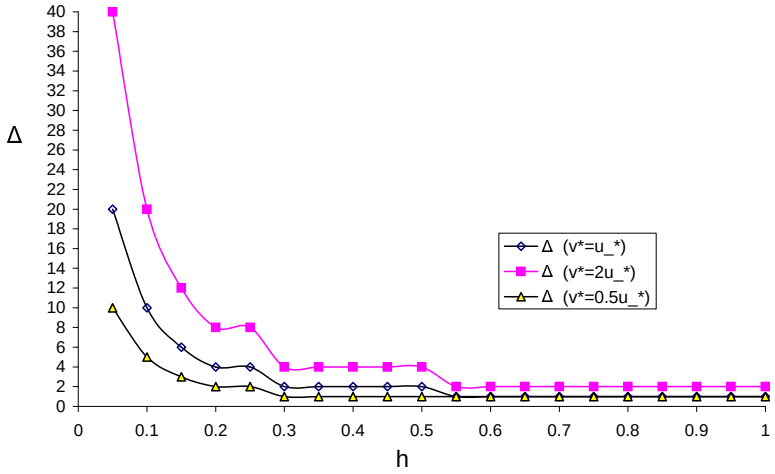


Fig. 3. The relation between the truthful reporting rate $h$ and the number of last transactions during which sellers have no incentive to cooperate in case buyers use the selection algorithm $\langle \mathcal{N}, 1 \rangle$.

## 4. COST-EFFICIENT REPUTATION MANAGEMENT

### 4.1 Implementation Cost of Computational Trust Models

Accurate computational trust models are costly to build and to maintain. Several sources may contribute to the cost of such computational mechanism, as summarized in Definition 4.1.

**Definition** 4.1. The implementation cost of a computational trust model $\langle P_i, V_i, \mathcal{F}_j, A, D \rangle$ (Definition 2.1) consists of the following component cost:

- communication cost $\tau_c$ to explore the peers $P_i$ and retrieve relevant ratings $V_i$.
- computation cost $\tau_e$ of the algorithm $A$ and decision making algorithm $\mathcal{D}$ to aggregate, analyze ratings and make appropriate decisions.
- possible monetary cost $\tau_m$: since reliable ratings can be considered as sellable goods offered by third-party monitoring agents in the system, buying them incurs certain cost. This cost may also include the payment for participants in system with any side-payment mechanism to elicit truthful reporting behaviors [Miller et al. 2005].
- storage cost $\tau_s$ to store and to maintain rating information and historical performance of potential partners/raters.
- hidden/opportunity cost $\tau_o$: time-consuming learning algorithms may lead to late decisions and hence is considered as more costly than fast algorithms, especially in competitive scenarios.

A detailed study of the cost model of each computational model is much dependent on the system and application on which we apply the algorithm. For example, the estimation of opportunity cost $\tau_o$ during the lifetime of a peer is non-trivial, though it maybe significant. Such a thorough study of all cost types of existing algorithms is out of the scope of the paper. In this work we only focus on the most obvious and measurable cost of communication $\tau_c$. Evaluations of such communication cost $\tau_c$ of well-known trust learning algorithms will be presented in Section 6.

### 4.2 Cost-effective Usage of a Computational Trust Model

This section studies the cost and benefits of using different trust learning algorithms and investigates the possibilities of minimizing the cost while still maintaining a high level of cooperation in the system. While an algorithm with better accuracy is generally preferable (see Theorem 3.2), it usually comes with higher cost. For instance, an algorithm relying on the retrieval/buying reliable information from trusted third-party agents may give us more accurate information about the sellers, yet it is apparently more expensive than a simple averaging algorithm (Example 2.3).

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 algorithm with misclassification errors $\alpha_1 = \alpha, \beta_1 = \beta$, both upper-bounded by some $\epsilon < 0.5$, and with an expected cost $\mathcal{C}_1$. A typical example of $\mathcal{R}_1$ is to buy information from some third-party monitoring agents to get correct estimate of the behavior of the last buyer/rater. 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 $\mathcal{C}_2 \ll \mathcal{C}_1$. The cost $\mathcal{C}_1, \mathcal{C}_2$ may compose of certain component cost in Definition 4.1.

On one hand, since $\alpha, \beta$ are less than $\epsilon \leq 0.5$, the computational model $\mathcal{R}_1$ can be used to motivate the cooperation of a seller in most transactions (Theorem 3.2), yet 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. However, this use of the trust model $\mathcal{N}$ is, less costly, and thus more preferable.

In this scenario, our approach is to let a buyer use the computational model $\mathcal{R}_1$ with probability $c$ and model $\mathcal{R}_2$ with probability $1-c$. Theorem 4.2 proposes a way to optimize the cost of using expensive computational model $\mathcal{R}_1$ while still ensuring cooperation in the system.

THEOREM 4.2. *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 naively model $\mathcal{N}$ with probability $1 - c$. The seller finds it optimal to cooperate in all but its last $\Delta$ 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.*

PROOF. According to Theorem 3.2 (k=1), using only the algorithm $\mathcal{R}_1$ can ensure cooperations of a rational seller in all transactions but its last $\Delta > max\{1, \lfloor \frac{v^*}{u_*(1-2\epsilon)} \rfloor\}$ ones.

Let $\delta \geq \Delta$ be the number of remaining transactions of the seller at the current step. Proceed as in Theorem 3.2 with $k = 1$, $\alpha' = c\alpha_1 + (1-c)\alpha_2 = 1 - c + c\alpha$, and $\beta' = c\beta_1 + (1-c)\beta_2 = c\beta$, we get $\delta_{hc} \geq -v^* + \delta u_*(h(1-c) + c(1-\alpha-\beta-(\beta-\alpha)h))$.

Since $0 \leq h \leq 1$, it follows that $\alpha + \beta + (\beta - \alpha)h \leq \alpha + \beta + \max\{\beta - \alpha, 0\} \leq 2\max\{\beta, \alpha\} \leq 2\epsilon$. Thus, $h(1-c) + c(1-\alpha-\beta-(\beta-\alpha)h) \geq c(1-2\epsilon)$ for $c \in [0,1]$. This makes $\delta_{hc} \geq -v^* + \delta u_* c(1-2\epsilon)$.

Therefore, $\delta_{hc} \geq 0$, or being cooperation is a dominant strategy for the provider if and only if $c \geq c_* = \frac{v^*}{\delta u_*(1-2\epsilon)}$ □

Fig. 4 shows the minimal probability $c_*$ of using the expensive trust model $\mathcal{R}_1$ depending on the estimated remaining transactions $\delta$ of the seller, with $u_* = v^*$ and $\epsilon = 0.35$. In

case most sellers staying in the system infinitely or long enough, the mix of two computational trust models help to reduce the total implementation cost significantly, as shown in Corollary 4.3. The cost of using only one expensive trust model is, on the other hand, much higher $O(\mathcal{C}_1 N)$.

COROLLARY 4.3. *In case most sellers staying in the system infinitely or long enough, the mix of two computational trust models as in Theorem 4.2 has an expected accumulative implementation cost $O(\mathcal{C}_1 log(N))$, where a seller stays in the system for $N$ transactions.*

PROOF. Since the cost of the naive model $\mathcal{N}$ is negligible, we can estimate the total accumulative implementation cost of the mixed approach in Theorem 4.2 as $\mathcal{C}(N) = \mathcal{C}_1 \frac{v^*}{u_*(1-2\epsilon)} \sum_{\delta=\Delta}^{N} \frac{1}{\delta}$, where $N$ is the number of transactions of rational sellers.

Since $\lim_{N\to\infty} \sum_{\delta=1}^{N} \frac{1}{\delta} - logN = \gamma$, where $\gamma = 0.577...$ is the Euler-Mascheroni constant, it follows that $\mathcal{C}(N)$ is $O(\mathcal{C}_1 logN)$, which is significantly lower than the implementation cost $O(\mathcal{C}_1 N)$ of using only the expensive trust model $\mathcal{R}_1$. □

Hence, given the rationality of participants, the accurate computational trust algorithm $\mathcal{R}_1$ mostly plays the role of a sanctioning tool rather than the role of learning trustworthiness of potential partners.

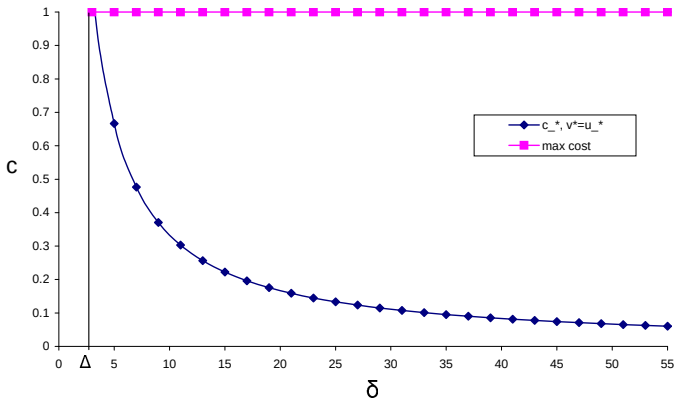


Fig. 4. The minimal necessary probability $c_*$ to use the expensive algorithm $\mathcal{R}_1$ depending on the number of remaining transactions $\delta$ of a rational seller for $\epsilon = 0.35, u_* = v^*$.

## 5. USING REPUTATION INFORMATION IN DYNAMIC SCENARIOS

The above analysis shows that if rational peers staying in the system infinitely, cooperation is ensured relatively easily. Based on the above results, this section proposes a seller-selection protocol for rational buyers that ensures cooperation of rational sellers for most transactions in the dynamic scenarios, where peers can join and leave dynamically after a number of transactions.

We assume in such dynamic scenarios, the following information is global knowledge of all peers: first, the distribution of life-time of sellers, which is observable and learnable to all peers. Second, peers know misclassification errors $\alpha, \beta$ of the learning algorithm $\mathcal{R}$

being used. These errors may depend on the overall rate of honest reporting users $h$. In fact, $\alpha, \beta$ and $h$ can be learnt from the own experience of each peer.

Rational peers can only use all available information and behave strategically to maximize their long-term benefits. Thus, a rational buyer has to make decision based on the expected number of remaining transactions of each candidate seller.

We approximate the algorithm for a rational (strategic) buyer to select a seller before each transaction as in Algorithm 1.

**Algorithm 1** selectSeller(candidate sellers set $S$, algorithm $\mathcal{R}$, globalBlacklist $L$): seller $s$

1: $EligibleSellers = \varnothing$;
2: **for** each $s \in S$ **do**
3: **if** $s \in L$ **then**
4: continue;
5: **end if**
6: Measure the overall rate of honest reporting peers $h$;
7: Get misclassification errors $\alpha, \beta$ of algorithm $\mathcal{R}$ for current $h$;
8: Estimate benefit $u_*$ and current cheating gain $v$ of $s$;
9: $\Delta_{min} = max\{1, \lfloor \frac{v}{u_*(1-\alpha-\beta-(\beta-\alpha)h)} \rfloor\}$;
10: $p[s]$ = Pr[seller $s$ stays at least $\Delta_{min}$ further transactions];
11: Get binary rating $r_i$ by peer $i$ on the latest transaction of $s$;
12: Run $\mathcal{R}$ to evaluate the reliability (binary trust) $t_i$ of $r_i$.
13: **if** $r_i == t_i$ **then**
14: $EligibleSellers = EligibleSellers \cup \{s, p[s]\}$;
15: **else**
16: Report to add $s$ to the global blacklist $L$;
17: **end if**
18: **end for**
19: Select a seller $s$ from $EligibleSellers$ with probability $p(s)/\sum_s p(s)$;

Algorithm 1 follows description of the seller-selection protocol $\mathcal{S}_1 = \langle \mathcal{R}, 1 \rangle$ in Definition 2.5 with some modifications. Practically, cautious buyers can simply use measured values of $\alpha, \beta$ in worst case scenarios (their upper-bound $\epsilon$) instead of estimating the honest reporting rate $h$ and measure those errors $\alpha$, $\beta$ (lines 6 and 7). Other quantities can be obtained easily. For example, in an e-trading system, the minimal legal benefit $u_*$ of a seller is the minimal value of an item accepted for trading in the system. The gain $v$ is the value of the current resource plus the shipping cost announced by the seller. $\Delta_{min}$ is the minimal number of remaining transactions that a rational seller finds incentives to cooperate for the current level of $\alpha, \beta$, and $h$ (see Proposition 3.6). Since we assume that the distribution of number of transactions by sellers can be learnt from their trading history in the system, it is trivial to estimate the probability $p[s]$ that a seller stays in $\Delta_{min}$ further transactions. This probability can be seen approximately the probability that this strategic seller cooperates in the current transaction compared and thus is used as a selection criteria (line 19).

According to the above analysis (Proposition 3.6), the best strategies of rational sellers are described as in Algorithm 2. That is, a strategic seller will cooperate in all transaction except its last $\Delta$ ones, also considering current rate of honest reporting $h$ in the system. Similar to a buyer, the seller estimates its $\Delta$ parameter based on the global knowledge of misclassification errors $\alpha, \beta$ of the learning algorithm being used, its current temporal

gain $v$ and minimal legal benefit $u_*$ at each step. Note that the values of $\Delta$ and $\Delta_*$ in Algorithm 2 are more accurate than the client-estimated $\Delta_{min}$ in Algorithm 1 since a seller generally knows all of its $v, u$ and its total number of transactions.

**Algorithm 2** bestServingStrategy(alg $\mathcal{R}$): servingStrategy

1: Measure the overall rate of honest reporting $h$;
2: Get misclassification errors $\alpha, \beta$ of alg. $\mathcal{R}$ for current $h$ level;
3: Estimate the minimal own benefit $u_*$ and cheating gain $v$;
4: $\Delta_* = max\{1, \lfloor \frac{v}{u_*(1-\alpha-\beta-(\beta-\alpha)h)} \rfloor\}$; /* estimate what the buyer estimates*/
5: Estimate my remaining number of transactions $\Delta$;
6: **if** $\Delta \geq \Delta_*$ **then**
7: Return good;
8: **else**
9: Return bad;
10: **end if**

Using the above seller-selection protocol, the number of successful transactions, or cooperation level, in the system generally depends on how accurate buyers can estimate the number of remaining transactions of a seller in order to select the right one still having incentives to cooperate. More concretely, a strategic seller is motivated to cooperate if $\Delta_{min} \geq \Delta_*$.

The algorithm for a strategic buyer to use a mix of two algorithms $\mathcal{R}$ and $\mathcal{N}$ to select a seller (Theorem 4.2) is then implemented as follows. A buyer estimates the number of remaining transaction $\delta_{buyer}$ of a seller from expected number of transactions of all sellers in the system. Next, it computes the minimal probability $c_{buyer} = \frac{v}{\delta_{buyer} u_*(1-2\epsilon)}$ it needs to use $\mathcal{R}$ in the current transaction. On the other hand, the best thing a strategic seller can do is to also estimate those values $c_{buyer}, \delta_{buyer}$, and computes the minimal probability $c_{seller} = \frac{v}{\Delta u_*(1-2\epsilon)}$ for its remaining number of transactions $\Delta$. The seller cooperates only if it stays more than $\Delta > \Delta_*$ and $c_{buyer} \geq c_{seller}$.

Since it is very difficult to perform a rigorous analysis on the peer cooperation in such dynamic scenarios with various possible parameters of the join and leave processes of peers, we defer the measurement and analysis of cooperations later on (Section 6), where extensive simulations with various input parameters obtained from real case studies are performed for such analysis purposes.

## 6. EXPERIMENTAL ANALYSIS

### 6.1 Simulation Framework

We use our generic trust prototyping and simulation framework[8] as the simulation tool for all experiments. This tool enables the rapid development and testing of many trust computational models under a variety of environment settings. Particularly, new algorithms for peers and appropriate performance metrics can be easily defined and integrated into the framework without major efforts.

Fig. 5 shows an overall architectural view of our simulation and prototyping framework, which is composed of the following modular components. PeerApplication layer is an

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

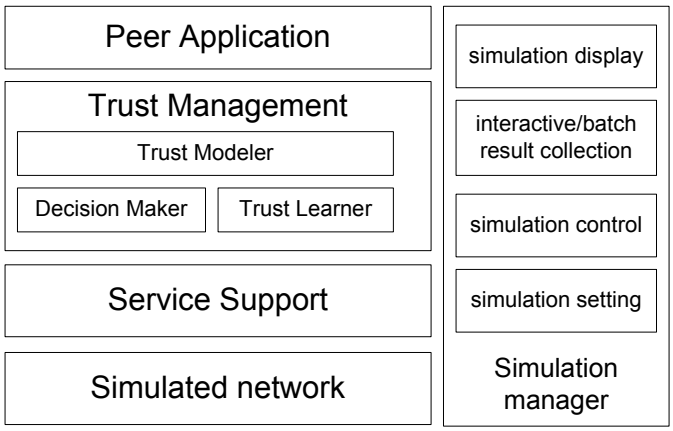


Fig. 5. Architecture of the trust prototyping and simulation framework

abstraction of the P2P application to be simulated. Such application is defined as an interaction scenario where peers provide or obtain resources of different prices and quality to and from the others. This abstraction makes it possible to tweak our system to simulate a range of similar applications, e.g., a C2C trading scenario, or a market of services, with the least efforts.

The TrustManagement layer serves as a trust management subsystem supporting peers in modeling and learning trustworthiness of other targets, as well as supporting them in their trust-based decision making processes. We provide standard APIs and many basic building blocks for users to develop and plug-in many computational trust (learning) models and decision making algorithms into the simulation without major efforts.

Another immediate layer, the ServiceSupport layer, encapsulates the distributed information storage, retrieval, and routing mechanisms being used. Particularly, this layer provides upper-layers with capabilities of sending requests and receiving responses among the peers, as well as searching and retrieving information on resources and related ratings on them for the trust computation. This layer helps separating the trust management subsystem from the underlying identity handling, searching, retrieving of information mechanisms, and thus facilitating the integrating and testing of a certain trust management approach on top of different distributed information management schemes.

The SimulatedNetwork layer represents the underlying communication network, which can be implemented as a centralized or decentralized system depending on the system being simulated. Since we only emphasize on the simulation and study of social and strategic behavior of peers in the application layer for a certain trust management approach, we only provide a basic implementation of this layer. Specifically, the layer is implemented as a network of nodes with latencies among them following the King latency data set [Gummadi et al. 2002], where of peers join and leave the system according to a realistic churn model [Stutzbach and Rejaie 2006]. In fact, our system can be configured to use other message passing protocols or underlying communication networks with any latency models as needed, besides the default implementation.

The SimulationManager takes care of all back-end supports for the simulation, e.g., capturing, inspecting, controlling, and collection of measured data in both interactive and

batch modes. We use RePastJ[9] as the base to develop this component, as this library provides several useful tools to help in setting up of the environments, dynamic controlling peer's parameters, performing visualization, and collecting simulation results, etc. Again, the modular designed and well-specified APIs of the SimulationManager layer makes it possible to use the system with other simulation libraries. In fact, we can provide a decentralized implementation of this layer to support larger-scale simulations or even emulate an application behavior on a real network via the help of real users. Fig. 6 shows some screenshots of our prototyping and simulation framework in actions, whose further details can be found online[10].

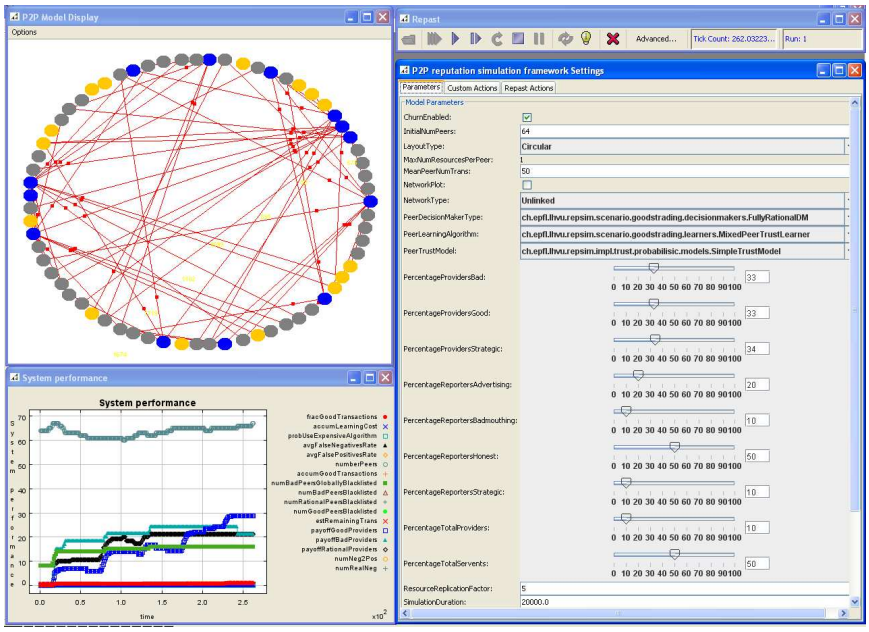


Fig. 6. Screenshots of the trust prototyping and simulation framework in GUI (interactive) mode. The system can be used to simulate interactions and cooperation among peers in different application scenarios by implementing a peer to provide appropriate services. Users can configure various settings of the environments (right panel) such as fraction of peers with different behaviors, trust computational models to be used. The trust relationships among peers are shown in the network of the top-left panel, where one can inspect and change behavior of certain peers during simulation time interactively, e.g., the screenshot honest participants are blue peers with many trust relationships. System performance over time is shown in the bottom-left panel. Most simulation settings such as distribution of different peer types with various behaviors and definitions of (new) system performance metrics can be done programmatically via provided APIs to facilitate the collection and detailed analysis of experimental results.

## 6.2 Simulation Setting

We verify our theoretical analysis and its practical application in the example C2C trading scenario with the above prototyping and simulation framework. The details on the

[9]http://repast.sourceforge.net
[10]http://lsirpeople.epfl.ch/lhvu/download/repsim/

modeling, implementation, and configuration files for all experiments in this paper are available[11].

Peers are modeled as sellers and/or buyers of many items and exhibit different behaviors. Cooperation level and accumulated utilities of different peer types who sell and buy goods of different prices are then measured and analyzed under various simulations settings: first, peers can leave and join dynamically, thus having different number of transactions during their lifetime. Second, buyers use different computational trust models with different personalized inputs and settings to evaluate the rating trustworthiness of others. Third, peers exhibit several behaviors, both irrationally malicious and/or strategic, details to be presented in coming sections. To more accurately model the delay of information propagation in the network, we implement the routing of messages so that the search cost in the system similar to that of a logarithmic P2P routing overlay, i.e., it takes time $O(logn)$ to lookup a data item in system with $n$ peers. The latencies among nodes in the network are set up with the King-latency data set provided by [Gummadi et al. 2002], which measured the real latencies between nodes in the Internet. The dynamic joins and leaves of peers in the system are simulated according to statistics from real-life case studies on many live peer-to-peer systems [Stutzbach and Rejaie 2006]. Specifically, we schedule the inter-arrival time of peers to follow a Weibull distribution with shape parameter $0.53$, and generate the up-time of peers to follow a Weibull distribution with shape parameter $0.34$ [Stutzbach and Rejaie 2006]. The scaling parameters of these Weibull distributions are set so that all peers live long enough to send and receive many messages during their up-time. We then compute 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 $\mu_{trans}$ transactions, where $\mu_{trans}$ is a parameter of the simulation.

### 6.3 Implementation of Peer Behaviors

Since it is impossible to implement full rationality, in our simulation rational peers are approximated as strategic ones who use all available information to find a most beneficial strategy for them at each step. A (strategic) buyer first searches for the available goods satisfying its requirements and then selects one according to Algorithm 1 and its variant introduced previously in Section 5.

A good seller ships the good after receiving the payment with a very high probability ($\gamma^+ = 0.99$ ) of satisfying the buyer and get a good rating. A bad seller either doesn't ship the good or ship a low quality item to the buyer, resulting in a very low probability ($\gamma^- \approx 0.05$) of meeting buyer's anticipation and thus likely to get a bad rating afterwards. Other sellers are strategic and use all available information to find the best strategy to follow, e.g., whether it should ship the item or not, so as to maximize its long-term utilities, following the best strategies described in Algorithm 2 (Section 5).

Regarding rating behaviors, peers consist of the following types: honest (always reports correctly about what it has observed), badmouthing (always reports negatively), advertising (always reports positively), ignoring (does not report), and strategic. Strategic raters can provide correct, incorrect ratings or does not leave any rating depending on each situation.To reduce the complexity of the simulation, in this paper we have to limit rating behavior of strategic peers to the following *safe* strategy. When asked by an unknown peer

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

Table II. Experimental settings of representative simulation scenarios with different types of sellers and buyers. Seller types consist of: $s\%$ strategic, $g\%$ good, and $b\%$ 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. We set $a > b$ since advertising behaviors are generally popular than badmouthing behaviors in most case studies of current reputation-enabled systems.

| **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. 7 |
| $C_2$. 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. 7 |
| $C_3$. 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. 7 |
| $C_4$. 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. 8 |
| $C_5$. Most sellers strategic | 85 | 5 | 10 | 0 to 100 | $\frac{100-h}{5}$ | $\frac{2(100-h)}{5}$ | $\frac{100-h}{5}$ | Fig. 8 |
| $C_6$. Different seller types | 33 | 34 | 33 | 0 to 100 | $\frac{100-h}{5}$ | $\frac{2(100-h)}{5}$ | $\frac{100-h}{5}$ | Fig. 8, 9 |
| $C_7$. All sellers strategic | 100 | 0 | 0 | 0 to 100 | $\frac{100-h}{5}$ | $\frac{2(100-h)}{5}$ | $\frac{100-h}{5}$ | Fig. 10 |

on the trustworthiness of a trusted, a blacklisted, or an unknown partner, a strategic peer respectively reports positively, negatively, or honestly. When asked by a trusted peer about trustworthiness of another peer, the strategic peer always reports honestly. Requests from blacklisted peers are ignored. This reporting strategy is chosen for many reasons. First, the cost of reporting after a transaction in our current application is negligible. Second, this strategy helps peers to rapidly build up and extend its trust relationships with many other peers in the system, thus *intuitively most beneficial* to them.

Settings of each experiment and corresponding result of the experiment are summarized in Table II. We simulate the system in most cases with dynamically leaves and joins of peers, starting with $n = 256$ peers. In convergence the number of peers approximately double this initial number with our churn model. We were able to run our simulation with up to $1024$ initial peers, and the results are similar. Each simulation is run until convergence and the measured outputs are averaged over $10$ different runs.

### 6.4 Accuracy of Example Computational Trust Models

We implemented three computational trust models and used each of them in our simulation to evaluate the reliability of a rater. We then combine each of them with the naive algorithm $\mathcal{N}$ to estimate the total saved cost as proposed by Theorem 4.2. The first computation model $\mathcal{L}$ is the PeerTrust PSM/DTC algorithm proposed by Xiong and Liu [Xiong and Liu 2004], presented earlier in Example 2.2. With this algorithm, a peer $i$ estimates the trustworthiness of another rater $j$ based on the similarity between $i$'s and $j$'s ratings on some other sellers which both $i$ and $j$ have contacted with. The second algorithm $\mathcal{X}$ is a the maximum likelihood estimation-based learning algorithm as in Example 2.3. 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 both have experienced with. We also implemented a dishonesty detector $\mathcal{A}$ with reasonably good misclassification errors: both $\alpha$, $\beta$ are less than $\epsilon = 0.1$, and with a high cost of each time being used. In practice, a peer can implement this detector by asking for information from the third

party monitoring or consulting agents before buying things. The dishonesty detector $\mathcal{A}$ 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 algorithms $\mathcal{L}$ and $\mathcal{X}$ are used to test the efficiency of the seller-selection protocol in a more relaxed environment 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 $\mathcal{L}$. The results for algorithm $\mathcal{X}$ are close to those of $\mathcal{L}$, where as the results of $\mathcal{A}$ are even better. The used of other global trust learning models like complain-based trust [Aberer and Despotovic 2001] or EigenTrust [Kamvar et al. 2003] in place of the algorithm $\mathcal{A}$ is subject to our future work.

As a base for other experiments, we first estimate the overall misclassification errors $\alpha, \beta$ 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 experimentation identified in Table II are based on two observations. First, it is not necessary to measure precisely the accuracy of a learning algorithm but only the overall trend and worst case misclassification errors $\alpha, \beta$ depending on the percentage of honest reporting peers $h$ in the system. Secondly, strategic selling behaviors can be excluded when estimating these misclassification errors for the reason that learning uses only historical data, and outcome of these strategic behaviors can be assumed to follow one of the overall trend discovered by three above extreme cases. We measure these statistics under three cases where there are different fraction of good and bad sellers in the systems (scenarios $C_1$, $C_2$, and $C_3$ in Table II). To increase readability, we show only the highest misclassification error values of $\alpha, \beta$ for each algorithm in our experiments. Note that such robustness or accuracy of these algorithms have already been extensively studied in related work [Despotovic and Aberer 2004; Xiong and Liu 2004], most of which having been shown to have low $\alpha, \beta$ under various attack scenarios. Therefore, the goal of our simulation here is just to confirm these expectations and to get statistics of these errors for later experiments.

Fig. 7 shows estimated maximal values of $\alpha, \beta$ of the learning algorithm $\mathcal{L}$ based on PeerTrust PSM/DTC approach proposed by Xiong and Liu [Xiong and Liu 2004] for the mean number of transactions $\mu_{trans} = 50$ in three extreme cases $C_1, C_2, C_3$ of Table II. Generally, the accuracy of $\mathcal{L}$ is highest where most peers staying in the system longer ($\mu_{trans} = 50$) and 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 these accuracy statistics.

We tested with different values of $\mu_{trans}$ and observed that when most peers only participate in very few transactions ($\mu_{trans} < 10$), the trust learning algorithm can not collect enough historical data for accurate learning, resulting in high errors $\alpha, \beta > 0.5$ and thus our approach is not applicable.

### 6.5 Cooperation in Various Environments

The levels of cooperation in the system with different fractions of strategic sellers and raters are given in Fig. 8 for $\mu_{trans} = 50$ (cases $C_4, C_5, C_6$ of Table II). The case of all strategic sellers shows even a better trend. We also perform experiments with other extreme cases, e.g., all sellers, buyers, and raters strategic, yet the results are similar and

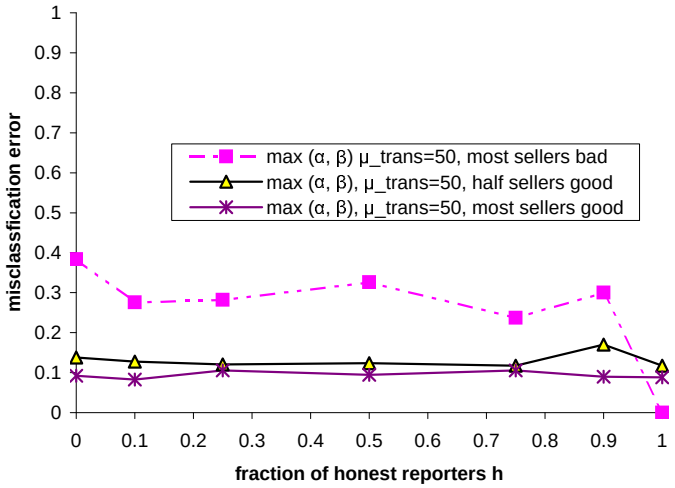


Fig. 7. The misclassification errors of algorithm $\mathcal{L}$ in three cases with different fraction of bad sellers and malicious raters, i.e., scenarios $C_1, C_2, C_3$ of Table II with $\mu_{trans} = 50$.

thus are not given here. For small $\mu_{trans} < 10$ the cooperation drops significantly, as the learning has no values at all.

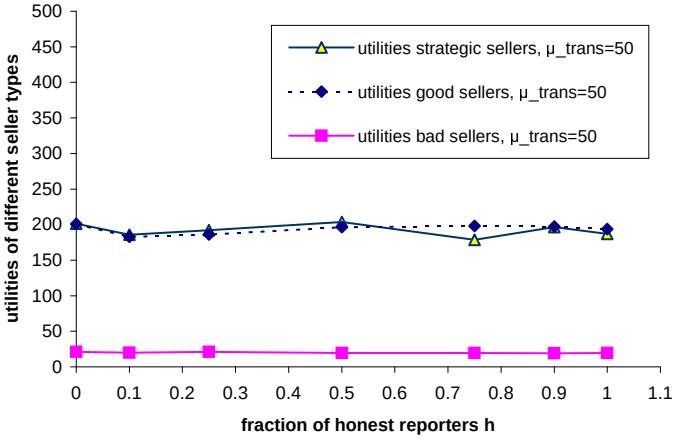


Fig. 8. Relation between cooperation (fraction of good transactions) and accuracy of a trust learning model in environments with mixed behaviors: honest, malicious, and strategic for $\mu_{trans} = 50$. Cooperation is very high since the learning accuracy is good.

In case there are many types of rating and serving behaviors (cases $C_4, C_5, C_6$ of Table II), the accumulated utilities of rational peers, following the serving strategies designed by the theoretical analysis (cooperate in all transactions except the last $\Delta$ ones), are generally the highest among different types. Representative result is shown in Fig. 9 with comparable number of peers with different types. Herein utilities are measured in terms of

accumulate gains (the price of an item is approx 10 units). In fact, utilities of strategic sellers are approximate to those of good ones, since strategic peers are enforced to cooperate most of their life. Our proposed approach still works even 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).

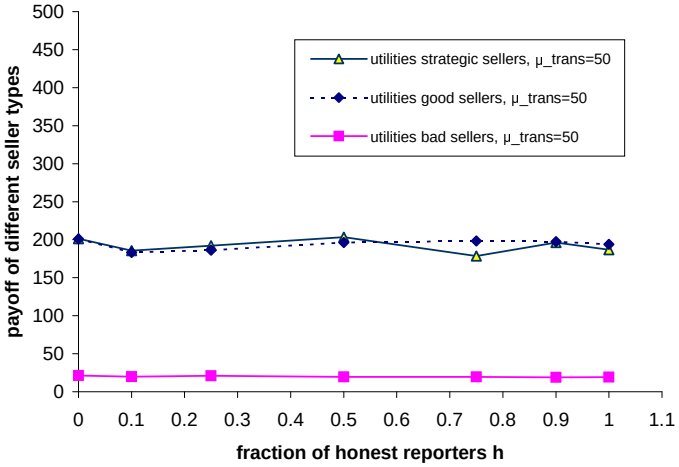


Fig. 9. Utilities of different types of sellers in a scenario with comparable number of peers with different types: malicious peers are quickly detected and blacklisted, thus bad sellers have very low life-time payoff. Utilities of strategic sellers are highest and approximate to that of good sellers, since they are enforced to cooperate in most of their transactions.

In case of all strategic peers, we can also use a combination of algorithms to minimize the total learning cost (as in Theorem 4.2). Fig. 10 compares the cost of two approaches: the first one uses only the algorithm $\mathcal{L}$, the second uses $\mathcal{L}$ with a low probability. It is interesting to see that the learning cost can be reduced significantly by using the expensive algorithm a very low probability while still maintaining a high level of cooperation in the system. Furthermore, the approach $M$ using the combination of two algorithms learns faster and thus during the same simulation time, the total number of good transactions in the whole system is also much higher.

## 7. RELATED WORK

A large number of work in trust research literature focus on developing appropriate computational models for learning peer behaviors based on their historical performance information, for example [Levien 2002; Kamvar et al. 2003; Srivatsa et al. 2005; Xiong and Liu 2004]. Comparative surveys of research literature on this subject can be found elsewhere [Golbeck 2006; Jøsang et al. 2005; Dellarocas 2005b; Despotovic and Aberer 2006]. Such computational models have been shown to be robust under various attack scenarios. These work are complementary to what we are doing, since any robust and accurate computational learning models can be used in our reputation-based peer selection protocol (c.f. Section 4).

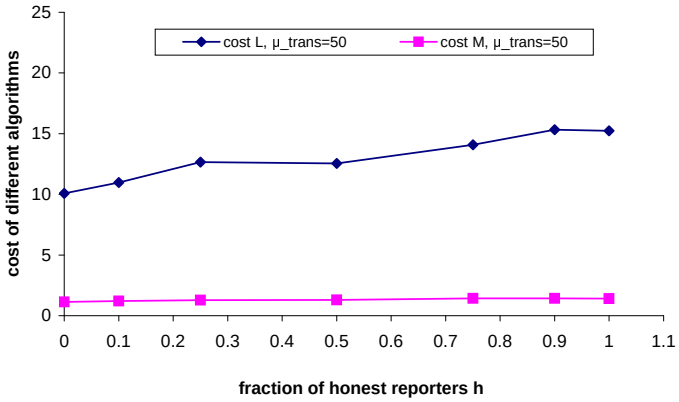


Fig. 10. Comparison the learning cost between two approaches: (L) is the use of a single algorithm $\mathcal{L}$, and (M) the mix of algorithm $\mathcal{L}$ and the naive algorithm $\mathcal{N}$. Herein, we measure the cost of the learning algorithm as the number of sent and received messages for a good transaction. Though both approaches have very high level cooperation (same as in Fig. 8), the second one has a significantly lower cost.

Our work is inspired by that of Dellarocas [Dellarocas 2003], which studies various design parameters of reputation mechanisms. Some conclusions in his paper coincide with the analysis in our work, e.g., the robustness of the naively optimistic algorithm (Section 3.3). However, in this paper we have considered many more aspects of a reputation-based computational trust model, namely its accuracy and cost, from which we have obtained other promising results. We have proposed a way to use such a computational model 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. Our cost-efficient reputation management solution in Section 4 is an application of inspection game theory [Avenhaus et al. 2002], wherein an accurate yet expensive trust learning algorithm plays the role of an inspector detecting dishonest behaviors of sellers who take the role of an inspectee of the game. To the best of our knowledge, this work is the first work to apply inspection game in reputation systems. The most related work to our cost-effective reputation management approach is [Agrawal and Terzi 2006], yet it addresses another problem of how to control the behaviors of agents in a centralized sovereign information sharing scenario. They propose to use an auditing device as a trusted centralized agent and decide the appropriate frequency of auditing. The notion of punishment amount and frequency are similar to our notion of probability to use the accurate and expensive algorithm in our selection algorithm. In fact, our work proposes an effective usage of reputation information that is applicable to a wider range of applications with different degrees of centralization.

## 8. DISCUSSION AND CONCLUSIONS

Many interesting observations can be derived from our previous analysis. First of all, by showing that there exist certain sufficient conditions for a computational trust model to be effective in enforcing cooperation, we provide an initial positive answer to the question whether existing trust learning algorithms in the literature produce the same effects in inducing the social optimum point of the system, where rational participants fully cooperate

with each other. It also implies that in a heterogeneous environment where peers use different learning algorithms with certain accuracy to learn trustworthiness of their potential partners, cooperation also emerges.

Secondly, depending on the presence of rationality from participant of the scenario being studied, either simple or sophisticated trust learning algorithms is appropriate: in environments with fully rational peers caring about their utilities, very simple algorithms are sufficient to stimulate cooperations in the system. In other scenarios where bad peers have the only goal of bringing the system down at any cost, sophisticated learning algorithms are still necessary to filter out malicious agents.

Our seller-selection protocol is based on the following ideas: to use a trust learning algorithm to evaluate the reliability of a most recent report on a seller. Such algorithm play the role of an inspector to detect only *past* malicious behaviors of raters and sellers. By introducing a simple selection protocol that punishes bad peers, we create a social-controlling mechanism to stimulate cooperations of rational peers in next transactions. This is different from other systems that mainly use sophisticated learning algorithms to predict future peer performance from the past. The inclusion of well-tested learning algorithms with high accuracy to filter out malicious partners in our solution helps it to perform well even in mixed environments with various malicious and rational behaviors. To a larger extent, our presented protocol establishes an umbrella framework to use reputation information effectively by exploiting both its sanctioning and signaling roles [Dellarocas 2005a] in decentralized and self-organized systems.

However, our analysis also has limitations. First, the results from the previous sections are the best outcome obtained from the equilibrium state of the system. In reality, behaviors of peers (buyers/sellers) may deviate from that outcome during the operation of the system, e.g., due to bounded rationality and limited availability of information. The second point is that in realistic peer-to-peer networks, peers can join and leave dynamically. Such dynamics of the system may encourage peers to milk their gained reputation, leave, and then rejoin the system under new identities. Our analysis shows that sellers may still be tempted to cheat in their last transactions or if the temporary gain $v$ is so high compared to normal benefits they obtain for being cooperative. Under this circumstance the approach proposed by this paper is only an approximate solution that introduce incentives for most players (buyers and sellers) who stay in the system long enough to be cooperative. In this context the key to ensure cooperation is not the accuracy/errors of the trust learning algorithm being used, but on the management of identities: the establishing of a new identity must be costly so that peers want to stay in the system for many transactions rather than changing their identities and start over. Such whitewashing behaviors can be prevented by using many existing techniques, for example, to impose an entrant fee on newly joined peers. For those applications where peers may get very large temporary gains for cheating, it is very hard to design any totally decentralized incentive-based mechanism to motivate short-term peers to be fully cooperative, and reputation information may not be an effective tool anyway.

As of our future work, we want to derive some theoretical bounds on the cooperation level in the system of our reputation management approach, given arrival and up-time distributions of peers. We are also implementing various serving strategies of bounded rational peers in our simulation framework: we assume sellers use reinforcement learning algorithms to explore many possibilities before deriving their best serving strategies to fol-

low. We expect that such empirical simulations give us more insights to the effectiveness of different trust learning algorithms in enforcing cooperation and building trust in presence of bounded-rational peers with limited available information. Our ultimate goal is to provide a "cookbook" for applications of different trust learning algorithms in open and decentralized systems.

## REFERENCES


ABERER, K., CUDRÉ-MAUROUX, P., DATTA, A., DESPOTOVIC, Z., HAUSWIRTH, M., PUNCEVA, M., AND SCHMIDT, R. 2003. P-Grid: a self-organizing structured P2P system. *SIGMOD Rec. 32,* 3, 29–33.

ABERER, K. AND DESPOTOVIC, Z. 2001. Managing trust in a peer-2-peer information system. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*. ACM Press, New York, NY, USA, 310–317.

AGRAWAL, R. AND TERZI, E. 2006. On honesty in sovereign information sharing. In *EDBT* (2006-03-15), Y. E. I. et al., Ed. Lecture Notes in Computer Science, vol. 3896. Springer, 240–256.

ANCEAUME, E. AND RAVOAJA, A. 2006. Incentive-based robust reputation mechanism for p2p services. In *OPODIS*. 305–319.

ASHRI, R., RAMCHURN, S. D., SABATER, J., LUCK, M., AND JENNINGS, N. R. 2005. Trust evaluation through relationship analysis. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 1005–1011.

AVENHAUS, R., STENGEL, B. V., AND ZAMIR, S. 2002. Inspection games. *Handbook of Game Theory with Economic Applications 3*, 1947–1987. available at http://ideas.repec.org/h/eee/gamchp/3-51.html.

BUYYA, R., STOCKINGER, H., GIDDY, J., AND ABRAMSON, D. 2001. Economic models for management of resources in peer-to-peer and grid computing. In *Proceedings of the SPIE International Conference on Commercial Applications for High-Performance Computing*. Denver, USA.

CORNELLI, F., DAMIANI, E., VIMERCATI, S. C., PARABOSCHI, S., AND SAMARATI, P. 2002. Choosing reputable servents in a P2P network. In *WWW '02: Proceedings of the 11th International Conference on World Wide Web*. ACM Press, New York, NY, USA, 376–386.

DATTA, A., HAUSWIRTH, M., AND ABERER, K. 2003. Beyond "web of trust": enabling p2p e-commerce. In E-Commerce, 2003. CEC 2003. IEEE International Conference on. *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, 303–312.

DELLAROCAS, C. 2003. Efficiency and robustness of binary feedback mechanisms in trading environments with moral hazard. 4297-03 (Apr.). available at http://ideas.repec.org/p/mit/sloanp/1852.html.

DELLAROCAS, C. 2005a. Reputation mechanism design in online trading environments with pure moral hazard. *Information Systems Research 16,* 2, 209–230.

DELLAROCAS, C. 2005b. Reputation Mechanisms. *Handbook on Economics and Information Systems (T. Hendershott, ed.), Elsevier Publishing*.

DESPOTOVIC, Z. 2005. Building trust-aware P2P systems. Ph.D. thesis, Swiss Federal Institute of Technology Lausanne, Switzerland.

DESPOTOVIC, Z. AND ABERER, K. 2004. A probabilistic approach to predict peers' performance in P2P networks. In *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004*, M. Klusch, S. Ossowski, V. Kashyap, and R. Unland, Eds. Vol. 3191 / 2004. 62–76.

DESPOTOVIC, Z. AND ABERER, K. 2006. P2P reputation management: Probabilistic estimation vs. social networks. *Journal of Computer Networks, Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security 50,* 4 (March), 485–500.

GOLBECK, J. 2006. Trust on the world wide web: A survey. *Foundations and Trends in Web Science 1,* 2, 131–197.

GUMMADI, K. P., SAROIU, S., AND GRIBBLE, S. D. 2002. King: estimating latency between arbitrary internet end hosts. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM Press, New York, NY, USA, 5–18.

JØSANG, A., ISMAIL, R., AND BOYD, C. 2005. A survey of trust and reputation systems for online service provision. *Decision Support Systems*.

JURCA, R. AND FALTINGS, B. 2006. Minimum payments that reward honest reputation feedback. In *Proceedings of the ACM Conference on Electronic Commerce*. Ann Arbor, Michigan, USA, 190–199.

KAMVAR, S. D., SCHLOSSER, M. T., AND MOLINA, H. G. 2003. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the Twelfth International World Wide Web Conference*.

LEVIEN, R. 2002. Attack-resistant trust metrics. Ph.D. thesis, University of California at Berkeley.

LIANG, Z. AND SHI, W. 2007. Analysis of ratings on trust inference in open environments. *Elsevier Performance Evaluation (forthcoming*.

MILLER, N., RESNICK, P., AND ZECKHAUSER, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science 51,* 9, 1359–1373.

PAPAZOGLOU, M. P. AND GEORGAKOPOULOS, D. 2003. Service-oriented computing. *Commun. ACM 46,* 10, 24–28.

PATEL, J., TEACY, W. T. L., JENNINGS, N. R., AND LUCK, M. 2005. A probabilistic trust model for handling inaccurate reputation sources. In *Proceedings of iTrust'05, France*, P. Herrmann, V. Issarny, and S. Shiu, Eds. Springer Berlin Heidelberg, 193–209.

RESNICK, P., KUWABARA, K., ZECKHAUSER, R., AND FRIEDMAN, E. 2000. Reputation systems. *Commun. ACM 43,* 12, 45–48.

SCHLOSSER, A., VOSS, M., AND BRCKNER, L. 2005. On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation 10.*

SRIVATSA, M., XIONG, L., AND LIU, L. 2005. Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In *WWW'05: Proceedings of the 14th international conference on World Wide Web*. ACM Press, New York, NY, USA, 422–431.

STUTZBACH, D. AND REJAIE, R. 2006. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*. ACM Press, New York, NY, USA, 189–202.

SUN, Y. L., HAN, Z., YU, W., AND LIU, K. J. R. 2006. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *INFOCOM'06*.

WHITBY, A., JØSANG, A., AND INDULSKA, J. 2005. Filtering out unfair ratings in Bayesian reputation systems. *The Icfain Journal of Management Research 4,* 2, 48–64.

XIONG, L. AND LIU, L. 2004. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng. 16,* 7, 843–857.