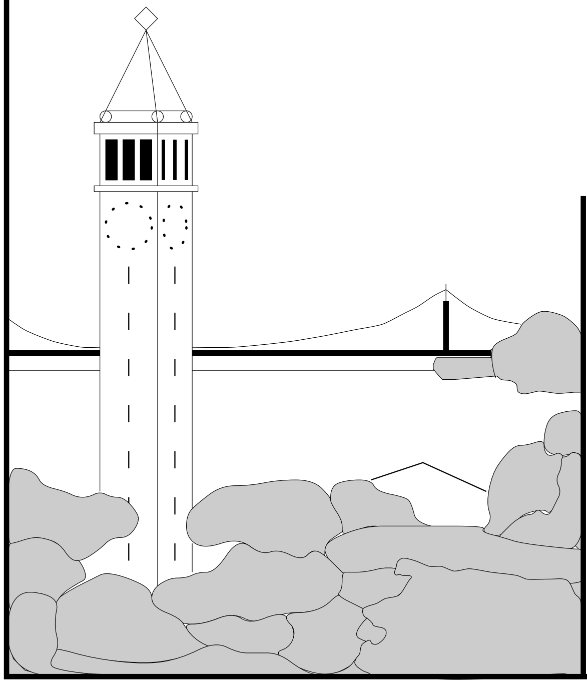
EM-trust: A Robust Reputation Algorithm for Peer-to-peer Marketplaces

Jonathan Traupman and Robert Wilensky



Report No. UCB/CSD-05-1400

July 2005

Computer Science Division (EECS)
University of California
Berkeley, California 94720

EM-trust: A Robust Reputation Algorithm for Peer-to-peer Marketplaces

Jonathan Traupman and Robert Wilensky[*]

July 7, 2005

Abstract

We have developed EM-trust, a robust algorithm for evaluating reputations in peer-to-peer marketplaces. EM-trust is robust in the sense that it is far less susceptible than are previous algorithms to errors due to inaccurate feedback. Moreover, a Bayesian version of EM-trust seems ideally suited to real peer-to-peer marketplaces, in which participants are unlikely to have had previous interactions. The properties of both EM-trust variants as well as eBay's "percent positive feedback" algorithm have been evaluated in a marketplace simulator designed to model a real peer-to-peer marketplace.

1 Introduction

In any marketplace, conducting a transaction entails risk. Will the seller deliver the promised goods on time and in good condition? Will the buyer provide payment? Traditionally, the reputations of the two parties involved helped to reduce this risk to the point that conducting business was possible. A store that has sold quality merchandise in the past will probably be a good place to shop in the future. One's history of paying debts on time will determine whether banks or retailers will extend credit.

However, these simple methods for evaluating reputations of buyers and sellers break down in large scale peer-to-peer marketplaces online. Many of these markets have so many participants that it is highly unlikely that a user has had past interactions with any given other person in these systems. In addition, users can easily change identities in these systems, allowing untrustworthy individuals a way to escape a bad reputation.

Reputation systems aid users of online peer-to-peer marketplaces in making trust decisions under these difficult conditions. Instead of relying on traditional word-of-mouth mechanisms for discovering the reputation potential trading partners, users of large online markets query the reputation system. Reputations are computed by aggregating the opinions of the other users in the system,

[*]This research supported in part by the Digital Libraries Initiative under grant NSF CA98-17353

effectively allowing users to evaluate the trustworthiness of total strangers by using the opinions of other total strangers.

Perhaps the most well known and widely used reputation system is the Feedback Forum on eBay, a major online auction marketplace. In this reputation system, users are asked to rate the people with whom they interact by giving them either positive, neutral, or negative feedback. eBay then takes these ratings and generates two reputation scores: the first, percent positive feedback, is simply the percentage of feedback that is positive. The second score is the number of positive feedbacks minus the number of negative feedbacks.

While the Feedback Forum has its documented problems [12, 4], it does seem to work well in practice. One of its weakest points, however, is that it naïvely assumes that the feedback users give is accurate. While most users are probably honest in their assessment of others, feedback does not always reliably indicate how well a user performed in a transaction.

Of particular concern is the problem of retaliatory negative feedback. In the typical retaliatory negative scenario, one of the participants in a transaction is unhappy with the other's performance and leaves a negative feedback. The recipient of the negative feedback responds with a negative feedback for the first user, even though he did nothing wrong aside from complain. It now becomes much more difficult to ascertain which negative is accurate, and should thus count against the user's reputation, and which is merely retaliation and should be ignored. The Feedback Forum currently makes no attempt to determine the accuracy of ratings and thus counts each feedback equally.

A further effect of retaliatory negatives is the overall chilling effect they have on participation in the feedback process. A single negative feedback will have a large effect on the reputation of a user who has participated in only a few transactions, while it will have scarcely any effect on users with long histories. Experienced users — typically large sellers — exploit this asymmetry and rarely leave feedback first. If they receive a negative, even one that is justified, they often respond with a retaliatory negative that can badly damage the reputation of a trading partner with only a few transactions. Small-time users are often aware of this tactic and thus won't leave negative feedback except in the case of outright fraud. The overall result is that negative feedback is discouraged and underreported [12]. Some researchers believe that eBay turns a blind eye to this underreporting since a marketplace with abundant positive feedback appears safer and more inviting to new customers [4].

In this paper, we present a new reputation algorithm, which we call EM-trust, that mitigates the effects of retaliatory negative feedback. It uses an Expectation-Maximization (EM) approach to calculate the maximum likelihood estimates for agents' probabilities of completing a transaction successfully.

Using a marketplace simulator, we demonstrate that EM-trust estimates agents' true honesty more accurately than eBay's percent positive feedback, even in the absence of retaliatory negative feedbacks. The relative performance of EM-trust improves further at high rates of retaliation. In addition, we show that EM-trust's ability to prevent failed transactions is comparable to eBay's, and it is significantly less likely to give low reputations unfairly to honest users.

Finally, we present a Bayesian version of EM-trust that yields even better results in our tests.

2 Related Work

Reputation systems grew out of earlier work on so-called "soft security," the problem of determining whether online services should be trusted, as contrasted with "hard security," which concerns itself with cryptographic solutions to the problem of communicating reliably and privately [10]. Marsh [11] is among the early work that combines notions of trust and reputation from philosophy, psychology, and economics and applies them to multi-agent systems. While he gives a roadmap of issues in the field, Marsh does not propose a reputation system.

The concept of reputation has been studied in some depth by the economics community, though their conclusions are often not applicable to practical reputation systems. For example, Kennes and Schiff [9] present a theoretical analysis of the value of a reputation system. However, their model assumes a two epoch time horizon that does not effectively capture user behavior in real markets (and which allows them to reach the dubious conclusion that reputation systems have negative value for buyers). Tadelis [14] models reputation as an asset that can be traded and finds that a good reputation is not always an indication of a good agent. However, most online reputation systems do not allow users to transfer their reputation to others.

Many existing proposals for reputation systems assume a pure peer-to-peer market where there is no central reputation system to query. These systems can be broadly divided into categories. On one hand are those that primarily use one's previous experience with an agent to estimate its reliability [8]. On the other are systems that use small-world phenomena to build chains of acquaintance to find other agents who can vouch for the reputation of a potential trading partner [7, 15]. Some systems, like Kasbah's Histos and Sporas system use a combination of both mechanisms [16]. Relying on past experiences to judge reputations is unlikely to work with eBay-like marketplaces because their size means that most interactions will be with strangers. It is also far from clear that the small-world model is applicable to these large scale peer-to-peer markets.

The problem of fraud in online markets and in the feedback system has recently attracted considerable attention both from researchers and the mainstream press. Dellarocas [5] discusses the problem of making robust evaluations of reputation despite unreliable feedback but does not propose an actual reputation system. The Pinocchio system [6] tries to detect and discourage inaccurate feedback using a combination of economic incentives and fraud detection. Unlike EM-trust, it does not try to compensate for any inaccurate feedback that is left.

Online marketplaces, and particularly eBay, have been widely studied by the economics and business communities. For example, Lucking-Reiley et al. look at online auctions of rare coins to determine what features drive pricing differences among similar items on eBay [3]. Bajari and Hortacsu [1] also look at pricing and compare eBay users' behavior to theoretically ideal auction behavior. Calkins [4] analyzes the eBay reputation system from a legal standpoint and finds it lacking.

Resnick and Zeckhauser performed a major empirical study [12] of eBay user behavior, including participation rates, bidding behavior and feedback. We have used their results as the foundation for our marketplace simulator. They conclude their study with several important lessons regarding the design of reputation systems. Resnick, Zeckhauser, and other collaborators also authored a comparison of various reputation systems [13], which analyzed their strengths and weakness, and proposed several requirements for successful reputation systems.

3 Robust Reputation Systems: Our Approach

Our goal is to create a reputation system that is robust in the face of the missing and inaccurate feedback that will inevitably confront real systems. We assume that we are operating in a peer-to-peer market like today's eBay. These markets are large enough that most agents have no interaction history with each other. There are also no clear strongly connected communities that would allow us to exploit network structure. Therefore, the only information source the reputation system uses is the potentially unreliable feedback left by other, unknown users.

We begin by defining an agent's reputation as the probability that the agent will perform acceptably in a transaction. Acceptable performance for a seller means selling only accurately described, functional products and sending the goods in a timely fashion through a reliable shipper. For a buyer, it involves remitting payment in an approved form on time. As suggested by [2], we currently do not try to assess motivation — poor performance caused by dishonesty or malice is indistinguishable from mere incompetence. Since the end result is the same, we do not think it is necessary to treat the sources of unacceptable performance separately.

The job of the reputation system, then, is to accurate estimate this probability from users' (possibly unreliable) feedback. We propose to do so by using an expectation-maximization approach, leading to an algorithm we call EM-trust.

Because most agents only interact with a tiny fraction of the other agents in a system, the matrix of feedbacks is typically very sparse. As with many such applications in which data are sparse, such approaches can be made more accurate, or can converge more quickly, if something is known about likely distribution of agents' behaviors. Taking the prior distribution of agent success rates into account leads to Bayesian EM-trust. This Bayesian approach may improve the estimation accuracy in the face of this sparsity, but introduces the additional problem of prior selection.

Testing such algorithms in a real market is not feasible, so we developed a simulator to evaluate our algorithms. This simulator was designed using the results of [12] to model eBay as closely as possible. We describe the simulator in detail further below.

3.1 The EM-trust Algorithm

When agent i interacts with agent j in a transaction, we observe two feedback variables, F_{ij} and F_{ji}, indicating the feedback left by agent i for agent j and vice versa. These variables are multinomial distributed and can take values in the set $\{-1, 1, 0\}$ indicating negative, positive, and no feedback respectively. We do not currently model neutral feedback, since it is both infrequently given and is considered by many to be merely a weak negative.

Also associated with each transaction are two latent random variables, P_{ij} and P_{ji}, indicating whether agent i and j respectively performed acceptably in the transaction. We assume independence between transactions, so the distribution of these Bernoulli random variables is characterized by their parameters, λ_i and λ_j, which we call the "honesty" of the agents.[1]

In general, the feedback variables can depend on the individual performance variables as well as on each other. These dependencies are complex and difficult to quantify, so we do not attempt to model them explicitly. However, we do make some assumptions about the way rational agents leave feedback. Our first assumption is that a positive feedback always indicates that the recipient behaved acceptably in the transactions. While there are reasons why an agent may withhold giving negative feedback to an underperforming partner (e.g., because of fear of retaliation), there is no compelling scenario under which it makes sense to leave a false positive feedback. Second, we assume that a negative feedback by itself does not indicate poor performance, unless the recipient of the negative feedback has left a positive for her partner. We know that the process of retaliation creates false negatives, and we also hypothesize that a nefarious user may leave pre-emptive false negatives to try to disguise her bad behavior.

The goal of EM-trust is to estimate the values of λ_i for all agents in the system. Since we do not observe the P_{ij} variables, we proceed by using an Expectation-Maximization algorithm to iteratively refine our estimates of these parameters.

We start with initial estimates $\lambda_i^{(0)}$ for each of the i agents' parameters. These starting values do not have a great effect in our application, so we start with $\lambda_i^{(0)} = 0$ for all i.

[1] The term "honesty" is merely a shorthand for "probability of acceptable performance." As noted above, these parameters encompass all possible causes of unacceptable behavior, not just deliberate dishonesty.

3.1.1 Expectation Step

For each transaction involving agent i, we calculate the conditional expectation that i performed acceptably given the observed feedback:

$$\begin{aligned}
\mathbb{E}[P_{ij}|F_{ij}=1,F_{ji}=1] &= 1\\
\mathbb{E}[P_{ij}|F_{ij}=0,F_{ji}=1] &= 1\\
\mathbb{E}[P_{ij}|F_{ij}=-1,F_{ji}=1] &= 1\\
\mathbb{E}[P_{ij}|F_{ij}=1,F_{ji}=-1] &= 0\\
\mathbb{E}[P_{ij}|F_{ij}=0,F_{ji}=-1] &= \mathbb{E}[P_{ij}|P_{ij}P_{ji}=0]\\
\mathbb{E}[P_{ij}|F_{ij}=-1,F_{ji}=0] &= \mathbb{E}[P_{ij}|P_{ij}P_{ji}=0]\\
\mathbb{E}[P_{ij}|F_{ij}=-1,F_{ji}=-1] &= \mathbb{E}[P_{ij}|P_{ij}P_{ji}=0]
\end{aligned}$$

We do not compute the expectation for the the two unlisted cases ($F_{ij} = 1, F_{ji} = 0$ and $F_{ij} = F_{ji} = 0$) and instead treat these transactions as missing data.

When at least one negative and no positive feedback is given, all we know is that someone behaved unacceptably. We cannot rely on the feedback being accurate in these cases, so we use the more fundamental expectation $\mathbb{E}[P_{ij}|P_{ij}P_{ji} = 0]$.

To compute this expectation, we simply look at the joint distribution, $\mathbb{P}\{P_{ij}P_{ji}, P_{ij}, P_{ji}\}$, given in the table[2]:

$P_{ij}P_{ji}$	P_{ij}	P_{ji}	$\mathbb{P}\{P_{ij}P_{ji}, P_{ij}, P_{ji}\}$
0	0	0	$(1-\lambda_i)(1-\lambda_j)$
0	0	1	$(1-\lambda_i)\lambda_j$
0	1	0	$\lambda_i(1-\lambda_j)$
1	1	1	$\lambda_i\lambda_j$

Simply marginalizing over P_{ji}, conditioning on $P_{ij}P_{ji} = 0$, and taking the expectation yields the expression:

$$\mathbb{E}[P_{ij}|P_{ij}P_{ji}=0] = \frac{\lambda_i^{(t)} - \lambda_i^{(t)}\lambda_j^{(t)}}{1-\lambda_i^{(t)}\lambda_j^{(t)}}$$

which we use for the expected performance in the previous estimation.

This estimation process is the key to EM-trust algorithm. We assume that negative feedback is mostly unreliable and so we penalize both parties in such a transaction. The amount of “blame” given to each party is based on the current reputations of the two parties in a transaction. The aim of this technique is to render retaliatory feedback irrelevant. If an agent has received a negative

[2] Missing entries have probability zero.

feedback, it does not matter whether it leaves a retaliatory negative or not: the reputations of both parties will be computed in the same fashion regardless of whether the agent retaliates.

The only way in which the recipient of a negative feedback can change the outcome of the reputation process is by leaving a positive for its partner, which will have the effect of shifting *all* blame for the transaction failure onto itself, lowering its reputation even further. On its face, it is not desirable for the reputation system to discourage agents from leaving honest feedback. However, agents rarely leave positive feedback for others that gave them negative feedback, even in eBay where such behavior will not result in a lower reputation. Resnick and Zeckhauser [12] report that none of the buyers and only 13% of sellers who received negative feedback respond with a positive feedback. Therefore we feel that making a pre-existing strategy, not praising those who criticize you, slightly more optimal is a worthwhile tradeoff for eliminating most of the effect of the far more damaging tactic of retaliation.

One minor problem is that this formulation of the conditional expectation can cause a division by zero if the estimate for $\lambda_i^{(t)}$ and $\lambda_j^{(t)}$ are both 1 on some iteration t. We solve this problem by setting any estimates of 1 to 0.999999999 or some other constant arbitrarily close to 1.

3.1.2 Maximization Step

For the maximization step, we use the conditional expectations computed above as if they were the observed values of the P_{ij} and use the standard maximum likelihood estimation formula for a Bernoulli random variable to compute updated estimates of the parameters λ_i.

Let $\langle P_{ij} \rangle^{(t)} = \mathbb{E}[P_{ij} | F_{ij}, F_{ji}]$ be the conditional expectation of agent i's behavior in a transaction with agent j computed in the expectation step of iteration t. Let A be the set of agents with which agent i has interacted and for whom we have computed $\langle P_{ij} \rangle^{(t)}$. We estimate the parameter $\lambda_i^{(t+1)}$, the updated value of agent i's honesty, by

$$\lambda_i^{(t+1)} = \frac{1}{|A|} \sum_{j \in A} \langle P_{ij} \rangle^{(t)}$$

We then use these updated parameter estimates in the next estimation step and repeat the whole process until convergence. While we have no bounds on the number of steps until convergence, EM algorithms generally converge quickly, which appears to be true in practice with EM-Trust.

Currently, if a pair of agents have had multiple transactions together, EM-trust behaves like eBay and only counts the most recent transaction. This approach makes it more difficult for malicious users to create bogus reputations by creating fake buyer accounts and leaving multiple positive feedbacks for sales of non-existent items. If it should become necessary to include all transactions between two users, the modifications to EM-trust would be trivial.

3.2 Bayesian EM-Trust

Because most agents will only interact with a handful of others during the course of their participation in the market, the data sets that EM-Trust will use will likely be highly sparse. This data sparseness may cause inaccuracies in the estimates, since maximum likelihood methods do not incorporate anything other than the observed data. For example, the estimated probability of success for a user with a single successful transaction and no unsuccessful ones will be one. Yet with only one datapoint, it is hard to have much confidence in this estimate.

To help manage this sparseness, we developed a Bayesian version of EM-Trust that uses a prior distribution of agent behavior as well as the observed data to estimate the agent's actual behavior distribution. For a prior we use a mixture of Beta distributions:

$$\gamma \mathrm{Beta}(\alpha_1, \beta_1) + (1-\gamma)\mathrm{Beta}(\alpha_2, \beta_2)$$

where the α_1 and β_1 parameters describe the probability distribution of acceptable performance among agents that are mostly honest and competent (i.e., the "good" agents) and α_2 and β_2 describe the distribution of acceptable performance among mostly dishonest or incompetent (i.e. "bad") agents. The γ parameter describes the proportion of good agents in the market. As with all Bayesian estimators, these parameters need to be calculated through some method other than using the actual data set.

The estimation step remains the same as in the standard EM-Trust algorithm, but the maximization step replaces the simple maximum likelihood estimate with the Bayesian estimate:

$$\lambda_i^{(t+1)} = \pi \frac{\alpha_1 + \sum_{j \in A} \langle P_{ij} \rangle^{(t)}}{\alpha_1 + \beta_1 + |A|} + (1-\pi) \frac{\alpha_2 + \sum_{j \in A} \langle P_{ij} \rangle^{(t)}}{\alpha_2 + \beta_2 + |A|}$$

where

$$\begin{aligned}
\pi &= \frac{1}{1 + \frac{1-\gamma}{\gamma} \frac{B(\alpha_2', \beta_2')}{B(\alpha_1', \beta_1')} \frac{B(\alpha_1, \beta_1)}{B(\alpha_2, \beta_2)}} \\
\alpha_1' &= \sum_{j \in A} \langle X_i \rangle_j^{(t)} + \alpha_1 \\
\beta_1' &= |A| - \sum_{j \in A} \langle X_i \rangle_j^{(t)} + \beta_1 \\
\alpha_2' &= \sum_{j \in A} \langle X_i \rangle_j^{(t)} + \alpha_2 \\
\beta_2' &= |A| - \sum_{j \in A} \langle X_i \rangle_j^{(t)} + \beta_2
\end{aligned}$$

and $B(\alpha, \beta)$ is the Beta function, $\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx$. The posterior distribution of the λ_i is also a mixture of Betas with parameters α_1', β_1', α_2', β_2', and mixture parameter π.

4 Marketplace Simulator

Ideally, we would test algorithms such as EM-Trust in a real peer-to-peer marketplace with agents of known competence. However, since such a marketplace is unavailable, we run our experiments in a simulated one. The design of the simulator is our attempt to achieve a balance between having enough complexity to accurately model how real agents interact and keeping the number of simulation parameters manageable.

We believe that the simulator achieves our desiderata: Its assumptions appear to us to be realistic, at least to a first approximation; it seems to model observed behavior well enough, and robustly enough, so that the results are plausible proxies for what would happen in real marketplaces. Moreover, as far as we can tell, it does not incorporate any biases designed to favor (or disfavor) our models. The robustness of the simulation results suggests that further complications are unwarranted, at least for our purposes. Of course, various improvements, such as more accurately and automatically determining model parameters, are possible. We discuss such further work below.

The following sections describe the simulator design in depth and discuss its many configurable parameters. As the simulator is a complex piece of software, the descriptions are necessarily fairly long. Full knowledge of the simulator design is not required for understanding the subsequent results.

4.1 Overall Design

For simplicity, our market assumes that all agents are trading in a single commodity at a fixed price. Since none of the algorithms we test currently use price, commodity type, or bidding behavior when calculating reputations, it was unnecessary to simulate the full auction process.

The simulator runs for a specified number epochs, each of which consist of a number of individual transactions. Between epochs, the simulator recalculates the reputations of all the agents in the system. We do not recalculate reputations after each transaction both for efficiency reasons and because feedback is not usually left immediately after completion of a transaction. All the simulations in this report were run for 200 epochs, each with a size of 1000 transactions.

To simulate a single transaction, the simulator first chooses a seller agent. Next, it chooses a buyer agent. The seller and buyer then decide whether they want to interact with each other based on their reputations. The simulator continues to choose buyers until it finds a pair that agrees to interact or until a fixed period of time elapses. If a buyer is never found, the seller fails to sell its goods and has to try again later.

Once a buyer/seller pair agrees to interact, the simulator determines their performance in the transaction. Each agent has an honesty parameter (λ_i) indicating their probability of performing acceptably, which is used to randomly generate their performance on each transaction.

Finally, the agents are allowed to leave feedback for each other based on their performance in the transaction. The agents' performance and feedback as well as the transaction time, ID, and other statistics are recorded and the simulator begins the process again for the next transaction.

At the end of an epoch, the simulator runs the reputation system to update agent reputations. It also records some snapshot information about the state of the market to be used for later analysis. Once the specified number of epochs has completed, it records further performance information and exits.

The simulator is controlled by setting the various parameters prior to running a simulation. These parameters and their default values are given in Table 1.

The simulator is written in Java and allows the user to configure the marketplace parameters, generate sets of agents, and run simulations using either an interactive command line or a batch scripting interface. Most importantly, it is possible to run multiple simulations with different reputation systems on the same initial set of agents.

4.2 Creating Agents

Before the simulator can begin to process transactions, it must create the pool of agents that will participate in the market. Agents are divided into four classes according to two orthogonal characteristics: their disposition can be either "good" or "bad," and their type is either "buyer" or "seller."

An agent's disposition determines how its honesty parameter is generated. The simulator first determines whether the agent will be good or bad randomly according to a user supplied parameter. Once the agent's disposition is chosen, the simulator generates the agent's honesty parameter randomly from a Beta distribution associated with the disposition. The honesty distribution for good agents usually has a high mean, while the distribution for bad agents has a low mean. Across all agents, the distribution of honesties generated by this process follow a mixture of these two Beta distributions, with the proportion of good agents in the system as the mixture parameter.

The number of buyer and seller agents is specified by the user. While both agent types can both buy and sell, sellers tend to sell more often than buy and buyers do the opposite. Associated with each agent is two Poisson rate parameters, one for buying and the other selling. Each of these parameters are randomly generated from a Gamma distribution that varies according to the agent type.

4.3 Generating Transactions

An agent's participation in the market is modeled by a pair of Poisson processes, one for buying and one for selling. The rate of these processes is controlled by the agent's buying and selling rate parameters. These processes generate a series of times when an agent wants to buy or sell in the market.

The simulator maintains two priority queues each containing all active agents sorted by their next buy or sell time. When conducting a transaction, the simulator chooses a seller from the front of the seller queue, then pulls potential buyers off the buyer queue until an agreeable match is found. If no buyer with

Simulator function	Parameter name	Default Value
General	Transactions per epoch	1000
	Number of epochs	200
Agent creation	Number of buyers	4000
	Number of sellers	1350
Agent honesty	Proportion of good agents	0.98
	Good agent sub-distribution α	18.0
	Good agent sub-distribution β	2.0
	Bad agent sub-distribution α	2.0
	Bad agent sub-distribution β	18.0
Buyer participation	Mean buy rate	0.2
	Variance of buy rate	0.08
	Mean sell rate	008
	Variance of sell rate	008
Seller participation	Mean buy rate	0.08
	Variance of buy rate	0.0128
	Mean sell rate	64
	Variance of sell rate	1.024
Interaction	Interaction threshold	0.884
	Threshold width	0.2
Respawning	New agent creation rate	25.0
	Respawn rate	0.6
Feedback	First feedback probability for good agents	0.3
	Second feedback probability for good agents	0.6
	First feedback probability for bad agents	0.1
	Second feedback probability for bad agents	0.5
Retaliation	Good agent retaliation rate	0.25
	Bad agent retaliation rate	0.75
Bayesian prior	γ (Mixture ratio)	0.98
	α_1	18.0
	β_1	2.0
	α_2	2.0
	β_2	18.0

Table 1: Description of simulator parameters and their default values.

a buy time less than four time units greater than the seller's seller time is available, the sale expires. Likewise, the buy offers of buyers who cannot find a seller within four units of their initial buy time also expire.

After either a completed transaction or when a buy or sell offer expires, the simulator generates a new buy or sell time as appropriate and inserts the agents back in the priority queues.

4.4 Agent Interactivity

Once a potential buyer/seller pair are generated using the buy/sell Poisson processes, the agents are given the choice of whether they want to interact with each other.

Two global parameters determine how agents choose whether to interact or not. The first parameter, the interaction threshold, determines the reputation value of its partner above which the agent is likely to want to interact. The second parameter, the transition width, is the width of the transition region between always wanting to interact and always declining to interact. The probability of interaction is given by a modified logistic function:

$$I(r) = \frac{1}{1 + e^{[-\frac{2 \ln 99}{w}(r-t)]}}$$

where r is the potential partner's reputation, t is the interaction threshold and w is the transition width.

In this function, the interaction threshold controls the value of r for which the probability of interaction is 0.5. The transition width controls the steepness of the slope of the logistic function and is defined as the length of the interval of r values where $I(r)$ lies between 0.01 and 0.99.

To determine whether to interact with a potential partner, an agent feeds the partner's reputation to this function to obtain a probability of interaction. It then generates a Bernoulli distributed random variable with this probability to determine whether to interact or not. If the transition width is set to 0, the interaction is no longer stochastic and the agent always interacts if the partner's reputation is greater than the interaction threshold and declines to interact otherwise.

Agents start with zero reputations in the eBay and standard EM-Trust systems — Bayesian EM-Trust starts users at the mean of the reputation prior — so agents will never start to interact unless we handle interaction with new users specially. Therefore, if an agent's potential trading partner has received no feedback yet, the simulator substitutes the mean honesty for the new user's reputation when calculating the probability of interaction.

The simulator allow the user to specify the interaction threshold and transition width globally, but currently does not permit individual agents to have their own parameter values.

4.5 Leaving Feedback

Three parameters control the way in which agents leave feedback: the probability of leaving the first feedback, the probability of leaving a second feedback, and the probability of leaving a retaliatory negative. Agents have different feedback strategies depending on their disposition but currently, all agents of a given disposition share common parameters.

After the agent's behavior in a transaction has been recorded, the simulator polls each agent to see if they wish to leave the first feedback. If neither agent wants to leave the first feedback, neither leave feedback for the transaction. If both agree to leave the first feedback, the simulator picks one or the other randomly with equal probability.

The agent that leaves feedback first must determine the type of feedback to leave solely on the basis of its own behavior and its partner's behavior. Since the other agent has not yet left a feedback, it cannot base its decision on the other's feedback.

Good agents always leave accurate first feedback: if the other agent behaved correctly, it will leave a positive, otherwise it will leave a negative. A bad agent will not leave a positive feedback if it behaved dishonestly in the transaction because leaving a positive will allow its partner to leave a negative without fear of retribution. Instead, it will leave a pre-emptive negative feedback to try to disguise its responsibility for a bad transaction. If it behaved honestly, it will leave an accurate feedback.

While it is a tunable parameter, all of our tests were run with bad agents having a low probability of leaving the first feedback. We believe the optimal strategy for someone running a scam in a peer-to-peer market is to leave no feedback unless it receives a negative, in which case it retaliates. Using this strategy, in the best case no feedback is left, so there is no evidence of the scam.

After the first agent leaves feedback, the simulator queries the other agent if it wants to leave the second feedback. If the agent wants to leave a second feedback, it can base its decision on both the behavior of both agents in a transaction as well as the feedback left by the first agent. If either a good or bad agent receives a positive feedback, and it decides to leave any feedback, it will leave an accurate second feedback.

If a good agent receives a negative feedback it will leave a retaliatory negative feedback according to its probability of retaliation, regardless of its own behavior or that of its partner. If it does not choose to retaliate, the good agent will leave an accurate feedback. Except for retaliation, good agents do not generally try to game the feedback system because we assume their failures are due more to mistakes or lack of competence than to malice.

Bad agents will also retaliate for negative feedback according to their probability of retaliation. If they do not retaliate, they will always leave a negative feedback if the other agent did not behave correctly. However, a bad agent will leave a positive feedback for a correctly behaving partner only if the agent itself also behaved correctly. Otherwise, it will leave no feedback. We assume bad agent's behave badly chiefly out of dishonesty so they try to use the feedback system to cover their tracks as much as possible.

4.6 Agent Respawning

In real marketplaces, agents whose reputations fall too low will likely discard their identity and re-enter the system as a new user. This non-persistence of identities led Zacharia et al [16] to suggest that reputation systems should never allow reputation to fall below the level of a new user. While eBay (at least the percent positive feedback metric) and EM-Trust systems follow this guideline, the fact remains that a new user with a zero reputation will likely receive more trust than a user that has zero reputation after a number of transactions.

For this reason, any user that has received at least one feedback but whose reputation is below the mean prior honesty will throw away its old identity, since its chance of interacting is greater as a new user than as one with a poor reputation. The agent then either creates a new identity or leaves the market permanently. The distribution of these two events is controlled by a simulator parameter.

We simulate creating a new identity by adding a new agent with the same parameter values to the simulated marketplace. The old identity is marked inactive and is removed from the buying and selling queues. The simulator tracks these identity changes so that we can analyze the frequency with which individual agents change identities.

In addition to respawning agents, the simulator also adds new agents to the market according to a Poisson process whose rate is set by user-specified parameter. These new agents are generated in the same fashion and with the same honesty and buy/sell rate distributions as the original set of agents in the marketplace.

5 Experimental Results

In order to test EM-trust, we ran a series of simulations and used several metrics to compare the reputations returned by EM-trust, Bayesian EM-trust, and eBay percent positive feedback. The first test measures how accurately the three reputation systems can estimate the true underlying performance parameters for the agents in a marketplace. In the second test, we evaluate how well the algorithms detect and eliminate low performance agents. Finally, we briefly explore the sensitivity of Bayesian EM-trust to the choice of prior distribution.

5.1 Predicting Honesty

Our first experiment measures the three reputation systems' abilities to learn the actual honesty of a set of agents. We tested EM-trust, Bayesian EM-trust, and eBay percent positive feedback in the marketplace described by the parameters in Table 1. The ratio of buyers to sellers, their buying and selling rates, and the number of low-performance agents were chosen to produce interaction statistics that are close to the real world results reported by [12]. The interaction threshold was set to 0.884, the mean honesty value of agents in the system, implying that a new agent has a 50% chance of being allowed to interact in a transaction.

For each reputation system, we ran simulations at four different levels of retaliation. The 0% level simulates a marketplace where all feedback is completely accurate: no one leaves false negatives in retaliation for a received negative feedback. At the 50% level, agents leave retaliatory negative feedback for about half of the negative feedbacks they receive. At the 100% level, agents always retaliate for negative feedback. Finally, we simulated a market where good agents leave retaliatory negatives 25% of the time, while bad agents retaliate 75% of the time. While we were unable to find concrete figures for the actual retaliation rates in real markets, anecdotally it is fairly high [12].

After each epoch of 1000 transactions, we computed the mean absolute error between the reputations returned by the three systems tested and the known ground truth honesties of the agents in the system. Each simulation was run 48 times and the results averaged in order to smooth out random fluctuations. The results of this test are shown in Figure 1.

In all four simulations, EM-trust outperformed eBay percent positive feedback, while Bayesian EM-trust did better than either of the other two. As the level of retaliation increases, both EM-trust's and eBay's performance degrades, though eBay percent positive feedback is affected considerably more than EM-trust. The level of retaliation has almost no effect on the performance of Bayesian EM-trust.

These graphs demonstrate that both EM-trust variants accomplish the goal we set for them: both perform at least as well as eBay percent positive feedback in the zero retaliation case, and are much less influenced by inaccurate feedback data introduced by retaliatory negatives. While these charts show results for only one set of simulation parameters, our testing indicates that they are qualitatively representative of a wide range of parameter values.

5.2 Classification Performance

While more accurate evaluations of agents' performance is certainly a desirable feature in a reputation system, the fundamental problem they aim to solve is one of classification. A participant in a peer-to-peer marketplace hopes to use the information returned by the reputation system to make a choice about whether to interact or not with a potential trading partner.

To test the three algorithms' ability to distinguish good users from bad ones, we looked at two statistics: the transaction success rate and the precision of agent deactivations. The transaction success rate is simply the percentage of transactions where both parties behaved correctly. The deactivation precision is the percentage of deactivated agents — agents that are removed from the system because of a low reputation — whose true honesty is less than the overall mean honesty. While different from the classical definitions, the former statistic can

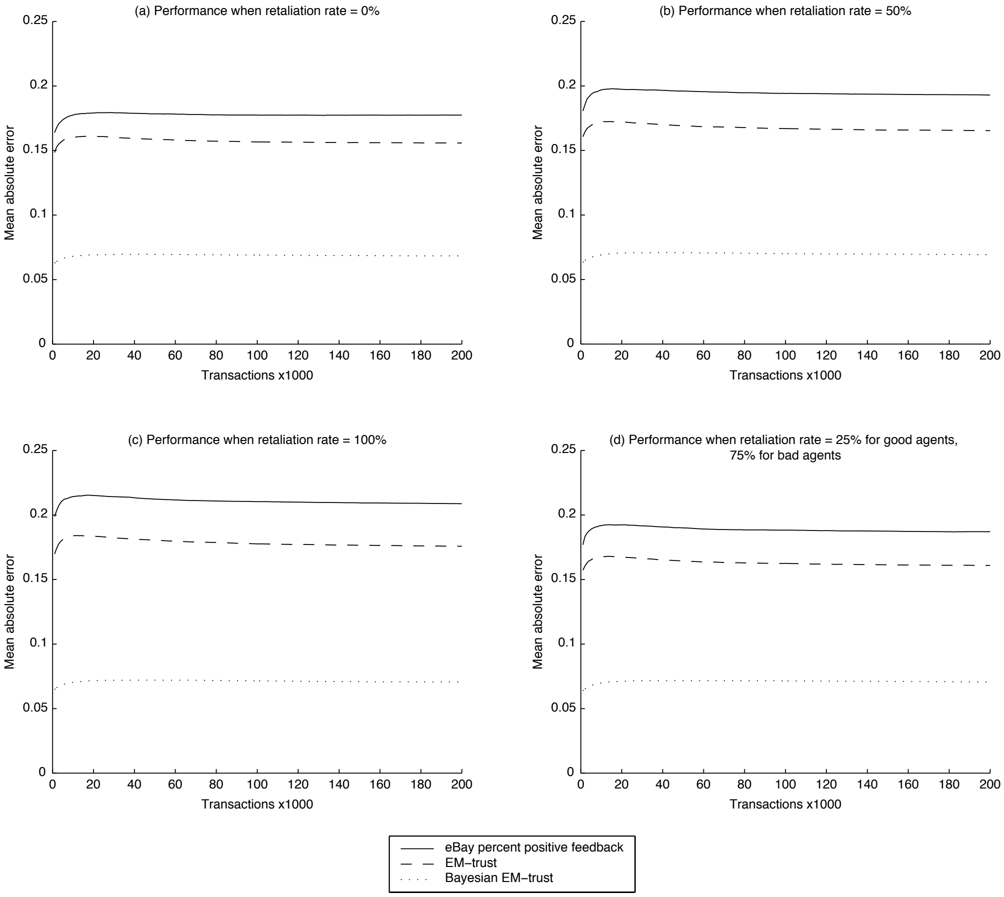


Figure 1: Reputation system accuracy with (a) retaliation rate = 0%, (b) retaliation rate = 50%, (c) retaliation rate = 100%, and (d) retaliation rate = 25% for good agents and 75% for bad agents. All graphs show the mean absolute error between reputations and true agent performance.

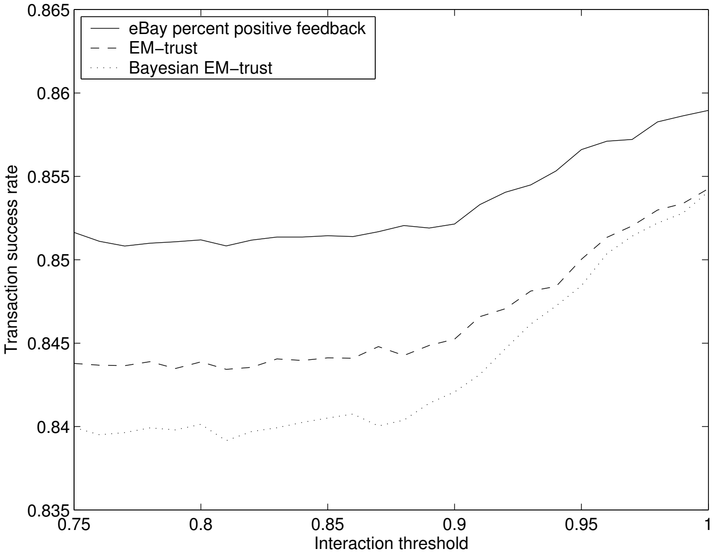


Figure 2: Effect of the interaction threshold on the transaction success rate when using eBay percent positive feedback, EM-trust, and Bayesian EM-trust reputation systems.

be interpreted as a form of recall and the latter as type of precision. An ideal reputation system would receive a score of 1.0 on both metrics.

We evaluated the algorithms over a range of interaction thresholds to show how these two statistics change with the selectivity of the marketplace's participants. The results are shown in Figures 2 and 3. As with the MAE results, each simulation was run 48 times and averaged to smooth out any random fluctuations introduced by the simulation process. All tests were conducted with the retaliation set to 25% for good users and 75% for bad users.

Of the three algorithms, eBay percent positive feedback yields the highest transaction success rate for a given interaction threshold. At low interaction thresholds, standard EM-trust outperforms its Bayesian variant, but the gap closes at higher thresholds. The performance of all three systems is very close, though, particularly at high interaction thresholds, where eBay has a transaction success rate of 85.9% compared to 85.4% for the two EM-trust variants.

The differences among reputation systems is much greater when looking at the deactivation precision. In this test, we see that the performance of the three algorithms is reversed: about 56.4% of agents deactivated by eBay are actually bad, while 61.2% of those deactivated by EM-trust are bad agents. Bayesian EM-trust does even better: 64.8% of its deactivations are for bad agents.

Interestingly, the deactivation precision does not appear to be strongly affected by the interaction threshold. This fact suggests that the safest interaction strategy is to trade only with the very best agents. Of course, such a strategy

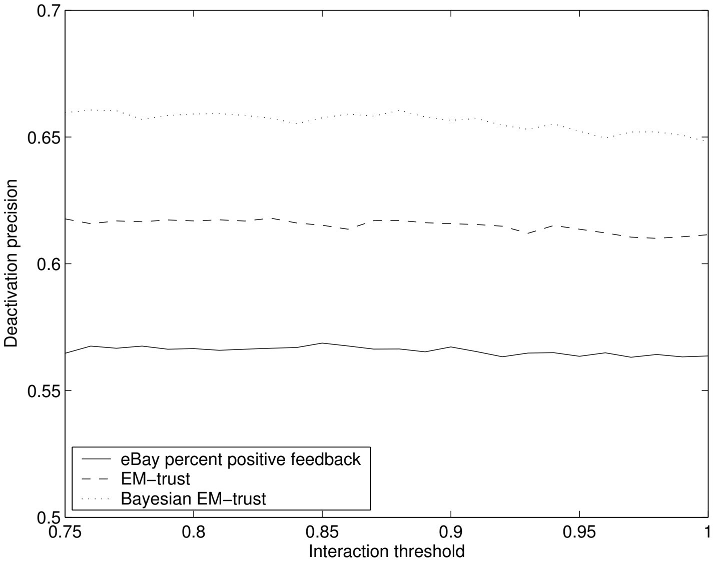


Figure 3: Effect of the interaction threshold on the deactivation precision when using eBay percent positive feedback, EM-trust, and Bayesian EM-trust reputation systems.

requires no algorithm to speak of, just a filtering of any agents against which are any black marks. This is typical of “precision-recall” results, in which the best value of one desiderata may be achieved by ignoring the other. However, assigning infinite value to safety is probably not useful in real markets: This strategy may introduce opportunity costs by making it harder to source or sell certain goods, particularly scarce or unpopular ones.

5.2.1 Combining Metrics

When there is a tradeoff between multiple performance metrics, it can be helpful to combine them in some way to form a single performance index for the systems being evaluated. In the case of traditional precision and recall, the harmonic mean is commonly used.

In this case, taking the harmonic mean of the transaction success rate and the deactivation precision yields a scalar performance index that ranges from 0 to 1, with the ideal reputation system having a value of 1.0. Figure 4 presents the results of the classification experiment in terms of this performance index.

According to this performance index, Bayesian EM-trust performs best, then standard EM-trust, and finally eBay percent positive feedback. The performance index weights the transaction success rate and deactivation precision as equally important. If we accept the assumption that both metrics have equal weight, the results imply that the observed reduction in the transaction success

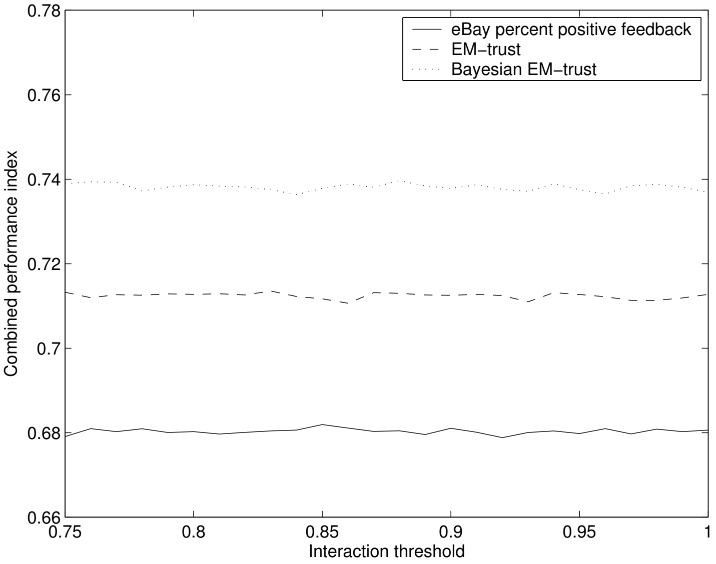


Figure 4: Effect of the interaction threshold on the overall performance index when using eBay percent positive feedback, EM-trust, and Bayesian EM-trust reputation systems.

rate is a small price to pay for the larger improvement in deactivation precision.

Furthermore, we can explore the overall performance when the two metrics are not considered equally important by using a performance metric computed with a weighted harmonic mean. Using such a metric, we found that eBay consistently outperforms EM-trust only if the transaction success rate is considered more than twenty times as important than the deactivation precision. For Bayesian EM-trust, the transaction success rate would have to be about 30 times more important before it is outperformed by eBay percent positive feedback.

5.3 Bayesian Priors

All of the above results were conducted with a “perfect” prior for Bayesian EM-trust. Since we control the actual distribution of agent honesties in the simulator, we can simply use this same distribution for the prior distribution in Bayesian EM-trust. In a real marketplace, however, we will not have access to the actual distribution of agent honesties. Instead, this prior will have to be determined through some combination of domain knowledge and historic data.

While techniques for determining this prior are beyond the scope of this paper, we can demonstrate how Bayesian EM-trust is affected by less than ideal priors. We tested six different priors, whose mixture of beta parameters are described in Table 2. The perfect prior is the prior used in the previous tests: we

Name	γ	$\boldsymbol{\alpha_1}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\alpha_2}$	$\boldsymbol{\beta_2}$	**Mean**
Perfect	0.98	18.0	2.0	2.0	18.0	0.884
Non-informative	1.0	1.0	1.0	—	—	0.5
Prior 1	0.98	10.0	1.0	1.0	10.0	0.893
Prior 2	0.9	18.0	2.0	2.0	18.0	0.82
Prior 3	0.9	10.0	1.0	1.0	10.0	0.83
Prior 4	1.0	15.24	2.0	—	—	0.884

Table 2: Parameters and mean of tested Bayesian priors

took the distribution used to generate the agents' performance parameter and used it as the Bayesian prior. In a real system, it is very unlikely we could ever get this accurate of an estimate of the true prior, so the perfect prior should be considered an upper bound on Bayesian EM-trust performance. At the opposite end of the spectrum is the non-informative prior. The non-informative prior indicates no prior knowledge of agent performance and is simply the Uniform(0,1) distribution expressed as a mixture of Betas.

The other four priors express different levels of inaccuracy in knowledge of the prior. Prior 1 has the mixing parameter correct but has incorrectly estimated sub-distributions. Prior 2 is the opposite: its sub-distributions are correct but it has the wrong mixture parameter. Prior 3 has both incorrect sub-distributions and an incorrect mixture parameter. Finally, Prior 4 is a simple Beta distribution with the second sub-distribution ignored.

Figure 5 shows the effect of different choices of prior on Bayesian EM-trust's MAE results in a marketplace with a 25% retaliation rate for good agents, and a 75% rate for bad agents. For each prior, we set the interaction threshold to the prior mean, implying that new users have 0.5 probability of interacting under all priors. Each variant was simulated 48 times and the results were averaged for this figure.

The choice of prior clearly influences performance of Bayesian EM-trust; however, performance is still high even with priors that deviate from the ideal. The algorithm seems more sensitive to the value of the mixture parameter than the sub-distributions. In particular, it seems that underestimating the mixture parameter — assuming there are more bad agents in the system than really exist — hurts performance significantly. Nevertheless, it is encouraging that Prior 4, a simple Beta distribution whose mean is equal to the average honesty, gave results slightly better than the theoretically ideal prior.

While imperfect priors give acceptable performance, it is still necessary to put considerable effort toward choosing a sufficiently good approximation to the true underlying prior distribution in order to realize the benefits of Bayesian EM-trust. Using a non-informative prior distribution results in a reputation system with performance much worse than both of the non-Bayesian alternatives.

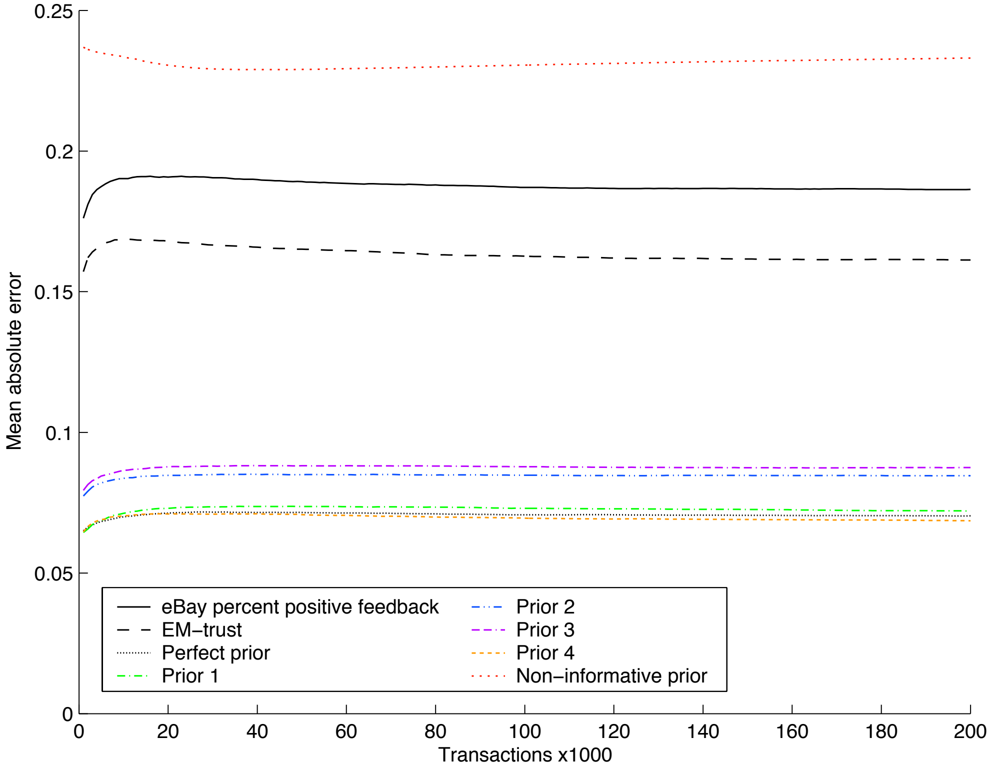


Figure 5: Effect of Bayesian prior on Bayesian EM-trust accuracy.

6 Future Work

Both EM-trust variants have clear benefits compared to eBay percent positive feedback. Moreover, there are many avenues for further exploration and improvement:

1. Temporal models of reputation. Currently both EM-trust variants assume that an agent's honesty is an immutable property. In reality, agent performance can change with time. Some users start off badly but learn from their mistakes. Others may be honest for a while but later decide that running a scam is more profitable than a legitimate business. In particular, we would like the reputation system to formalize the concepts of "initiation dues," the requirement that new agents prove themselves before their reputation will grow, and "stoning bad behavior," the quick destruction of the reputation of a good agent who turns bad, that Resnick et al. [12] observed as social phenomena on eBay.

2. Estimating the Bayesian prior. As discussed above, it is necessary to have a reasonably good estimate of the distribution of agent honesties in order for Bayesian EM-trust to show its advantages. We need to investigate techniques for performing this estimation, ideally with real marketplace data.

3. Improving simulator accuracy. Currently, our simulator models and parameters are based on the results in [12]. However, with more complete information about how agents interact in a real marketplace, we could fit more accurate models and create a more realistic simulator.

4. Exploring strategies for leaving feedback. Currently, we have a simple model for how agents leave feedback that is based on observations of eBay user behavior and anecdotal reports. Many of the strategies used by real users — e.g. threatening retaliation to deter negatives — have evolved as users learn how to play the feedback game. We would like to simulate this evolution under different reputation systems both to discover hidden flaws that allow the system to be gamed and to help guide the development of more robust algorithms.

5. Incorporating additional information sources. In addition to fellow users' feedbacks, there is other information available that can help an agent decide whether or not to conduct business with a potential partner. We would plan to investigate how data such as previous items bought/sold, prices, bidding behavior, and other auction information might help avoid failed transactions.

6. Application of reputation systems to other peer-to-peer systems. While EM-trust is targeted toward peer-to-peer markets, similar systems may be useful in other loosely coordinated networks, such as grid computing and distributed hash tables, for reducing freeloading, unreliable nodes, and other antisocial behavior.

7 Conclusion

In large scale peer-to-peer markets, the chances that a participant has any previous experience with a trading partner is vanishingly small. Therefore, the information provided by the reputation system is vital for building the trust necessary for a functional market. However, nearly all reputation systems require marketplace participants to provide honest feedback. If the reputation system does not provide accurate reputations, it will discourage participation in the system, running the risk of entering a vicious cycle of declining feedback rates leading to even less accurate reputations. This problem is particularly serious for reputation systems that give unfairly poor reputations to honest agents, the ones we would most likely expect to provide accurate feedback.

The EM-trust and Bayesian EM-trust algorithms we present in this report make improvements to the simple eBay averaging approach that we believe will help make reputation systems more accurate and useful. In the tests we conducted, both EM-trust variants estimated true agent honesty more accurately than eBay percent positive feedback did. While EM-trust did permit a 3% increase in failed transactions, it gave 19% fewer good agents unfairly poor reputations. Bayesian EM-trust had comparable success in preventing failed transactions and was even less likely to deactivate honest agents: about 37% less likely than eBay percent positive feedback.

Finally, both EM-trust variants virtually eliminate the problem of retaliatory negative feedback. Bayesian EM-trust is essentially uninfluenced by the rate of retaliatory negatives. Standard EM-trust is slightly affected by increasing retaliation rates, but still significantly less than eBay percent positive feedback.

In addition to the higher intrinsic accuracy of the two EM-trust algorithms, we believe that their abilities to ignore retaliatory negatives and to prevent good users from receiving unfairly low reputations will encourage more users to trust and thus participate in the feedback process. It is our hope that this increased participation will further increase feedback accuracy, creating instead a virtuous cycle that improves both the participation in and the accuracy of the reputation system.

References

[1] Patrick Bajari and Ali Hortaçsu. The winner's curse,reserve prices and endogenous entry: Empirical insights from ebay. *RAND Journal of Economics*, pages 329–355, Summer 2003.

[2] K. Suzanne Barber, Karen Fullam, and Joonoo Kim. Challenges for trust, fraud and deception research in multi-agent systems. *Trust, Reputation, and Security: Theories and Practice*, pages 8–14, 2003.

[3] Doug Bryan, David Lucking-Reiley, Naghi Prasad, and Daniel Reeves. Pennies from ebay: the determinants of price in online auctions. Papers 00-

w03, Vanderbilt - Economic and Business Administration, January 2000. available at http://ideas.repec.org/p/fth/vander/00-w03.html.

[4] Mary M. Calkins. My reputation always had more fun than me: The failure of ebay's feedback model to effectively prevent online auction fraud. *The Richmond Journal of Law and Technology*, 7(4), Spring 2001. http://law.richmond.edu/jolt/v7i4/note1.html.

[5] Chrysanthos Dellarocas. Building trust online: The design of robust reputation reporting mechanisms in online trading communties. In G. Doukidis N. Mylonopoulos N. Pouloudi, editor, *Information Society or Information Economy? A combined perspective on the digital era.* Idea Book Publishing, 2003.

[6] Alberto Fernandes, Evangelos Kotsovinos, Sven Östring, and Boris Dragovic. Pinocchio: Incentives for honest participation in distributed trust management. In *Proceedings of the 2nd International Conference on Trust Management (iTrust 2004)*, March 2004.

[7] Jennifer Golbeck and James Hendler. Reputation network analysis for email filtering. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, July 2004.

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

[9] John Kennes and Aaron Schiff. The value of a reputation system. Technical Report 0301011, Economics Working Paper Archive at WUSTL, January 2003. available at http://ideas.repec.org/p/wpa/wuwpio/0301011.html.

[10] Rohit Khare and Adam Rifkin. Weaving a web of trust. *World Wide Web J.*, 2(3):77–112, 1997.

[11] Stephen Marsh. *Formalising Trust as a Computational Concept.* PhD thesis, University of Stirling, 1994.

[12] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In Michael R. Baye, editor, *The Economics of the Internet and E-Commerce.* Elsevier Science, 2002. Volume 11 of Advances in Applied Microeconomics.

[13] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems: Facilitating trust in internet interactions. *Communications of the ACM*, 43(12):45–48, December 2000.

[14] Steven Tadelis. What's in a name? reputation as a tradeable asset. *The American Economic Review*, 89(3):548–563, June 1999.

[15] Bin Yu and Munindar Singh. A social mechanism for reputation management in electronic communities. In *Proceedings of the 4th International Workshop on Cooperative Information Agents (CIA)*, 2000.

[16] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic markets. In *Proceedings of the 32nd Hawaii International Conference on System Sciences*, 1999.