Detecting Deception in Reputation Management[*]

Bin Yu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7535, USA
byu@eos.ncsu.edu

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7535, USA
singh@ncsu.edu

ABSTRACT

We previously developed a social mechanism for distributed reputation management, in which an agent combines testimonies from several witnesses to determine its ratings of another agent. However, that approach does not fully protect against spurious ratings generated by malicious agents. This paper focuses on the problem of deception in testimony propagation and aggregation. We introduce some models of deception and study how to efficiently detect deceptive agents following those models. Our approach involves a novel application of the well-known weighted majority technique to belief function and their aggregation. We describe simulation experiments to study the number of apparently accurate witnesses found in different settings, the number of witnesses on prediction accuracy, and the evolution of trust networks.

Keywords

deception, trust networks, reputation, belief functions, weighted majority algorithm

1. INTRODUCTION

Reputation management is attracting much attention in the multiagent systems community [11, 12, 13, 15, 19, 20]. We consider the problem of distributed reputation management in large distributed systems of autonomous and heterogeneous agents. In such systems, it is generally inadvisable to assume that there are universally accepted trustworthy authorities who can declare the trustworthiness of different agents. Consequently, agents must rely on social mechanisms for accessing the reputation of other, unknown agents.

The basic idea is that the agents should help each other weed out undesirable (selfish, antisocial, or unreliable) participants. In our setting, the agents are in principle equal. They form ratings of others that they interact with. But to evaluate the trustworthiness of a given party, especially prior to any frequent direct interactions, the agents must rely on incorporating the knowledge of other agents—termed *witnesses*—who have interacted with the same party. The right witnesses cannot be found through any central mechanism either, so the agents must rely on a social mechanism for that purpose. Our method of choice is through referrals generated by agents who are directly acquainted with a potential witness. In other words, we use referrals to find witnesses and then combine the witnesses' testimonies to evaluate the party of interest. The testimonies are based on direct, independent observations, not on communications from others. As a consequence, we are assured that the testimonies can be combined without any risk of double counting of evidence. Double counting of evidence is risky in a distributed system, because it leads to rumors: agents holding opinions about others, just because they heard them from someone.

We developed an evidential model of reputation management based on the Dempster-Shafer theory of evidence. To do so effectively presupposes certain representation and reasoning capabilities on the part of each agent. Each agent has a set of *acquaintances*, a subset of which are identified as its *neighbors*. The neighbors are the agents that the given agent would contact and the agents that it would refer others to. An agent maintains a model of each acquaintance. This model includes the acquaintance's abilities to act in a trustworthy manner and to refer to other trustworthy agents, respectively. The first ability we term *expertise* and the second ability we term *sociability*. Each agent may modify its models of its acquaintances, potentially based on its direct interactions with the given acquaintance, based on interactions with agents referred to by the acquaintance, and based on ratings of this acquaintances received from other agents. More importantly, in our approach, agents can adaptively choose their neighbors, which they do every so often from among their current acquaintances.

The above approach helps find agents who receive high ratings from others. However, like other reputation approaches, the above approach does not fully protect against spurious ratings generated by malicious agents. This is because we assume that all witness are honest and always reveal their real ratings in their testimonies. The requesting agent does not consider the reputation of the witnesses and simply aggregates all available ratings. However, sometimes the witnesses may exaggerate positive or negative ratings, or offer testimonies that are outright false.

This paper studies deception as it may occur in rating aggregation. What makes the problem nontrivial are the following requirements. One, we wish the basic mechanism to aggregate testimonies as above so as to avoid the effect of rumors. Two, we would like to continue to use Dempster-Shafer belief functions to represent testimonies so as to capture uncertainty as well as rating. To do so, this paper develops a variant of the weighted majority algorithm applied to belief functions. It considers some simple models of deception and studies how to detect corresponding deceptions.

The rest of this paper is organized as follows. Section 2 provides

[*] This research was supported by the National Science Foundation under grant ITR-0081742.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2001 ACM X-XXXXX-XX-X/XX/XX ...$5.00.

some necessary background on distributed reputation management, especially involving local trust ratings and propagation through referrals. Section 3 introduces some deception models and describes our weighted majority algorithm as applied in detecting deception. Section 4 presents our experimental results. Section 5 compares our contributions to some related approaches. Section 6 concludes our paper with a discussion of the main results and directions for future research.

2. REPUTATION MANAGEMENT

The idea that the rating assigned to a party be based on direct observations as well the ratings assigned by other sources is well-known in the literature on reputation management. However, our approach addresses some important challenges: How does the agent find the right witnesses? How does the agent systematically incorporate the testimonies of those witnesses? First, our approach applies a process of referrals through which agents help one another find witnesses. Second, our approach includes the TrustNet representation through which the ratings can be combined in a principled manner.

2.1 Dempster-Shafer Theory

We use the Dempster-Shafer theory of evidence as the underlying computational framework. As observed by Kyburg, the Dempster-Shafer theory explicitly handles the notion of evidence pro and con [8]. There is no causal relationship between a hypothesis and its negation, so lack of belief does not imply disbelief. Rather, lack of belief in any particular hypothesis is allowed and reflects a state of uncertainty. This leads to the intuitive process of narrowing a hypothesis, in which initially most weight is given to uncertainty and replaced with belief or disbelief as evidence accumulates. We now introduce the key concepts of the Dempster-Shafer theory.

DEFINITION 1. *A frame of discernment, notated Θ, is the set of possibilities under consideration.*

Let T mean that the given agent considers a specified party to be trustworthy. Then, there are only two possibilities. That is, $\Theta = \{T, \neg T\}$.

DEFINITION 2. *A basic probability assignment (bpa) is a function $m : 2^\Theta \mapsto [0,1]$ where (1) $m(\phi) = 0$, and (2) $\sum_{\hat{A} \subset \Theta} m(\hat{A}) = 1$.*

Thus $m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}) = 1$. A bpa is similar to a probability assignment except that its domain is the subsets and not the members of Θ. The sum of the bpa's of the singleton subsets of Θ may be less than 1. For example, given the assignment of $m(\{T\}) = 0.8$, $m(\{\neg T\}) = 0$, $m(\{T, \neg T\}) = 0.2$, we have $m(\{T\}) + m(\{\neg T\}) = 0.8$, which is less than 1.

For $\hat{A} \subseteq \Theta$, the *belief function* $\mathsf{Bel}(\hat{A})$ is defined as the sum of the beliefs committed to the possibilities in $\hat{A}$. For example,

$$\mathsf{Bel}(\{T, \neg T\}) = m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}) = 1$$

For individual members of Θ (in this case, T and $\neg T$), Bel and m are equal. Thus

$$\mathsf{Bel}(\{T\}) = m(\{T\}) = 0.8, \text{ and } \mathsf{Bel}(\{\neg T\}) = m(\{\neg T\}) = 0$$

2.2 Local Trust Ratings

When agent A_i is evaluating the trustworthiness of agent A_j, there are two components to the evidence. The first component is the services offered by agent A_j. The second component is the testimonies from other agents in case A_i has no transactions with A_j before. Suppose agent A_i has rated the quality of service of the latest H interactions with A_j, $S_j = \{s_{j1}, s_{j2}, \ldots, s_{jH}\}$, where $0 \leq s_{jk} \leq 1$. We set s_{jk} to 0 if there is no response in the kth episode of interaction with A_j. Let $f(x_k)$ denote the probability that a quality of service of x_k is obtained in the kth episode of interaction with A_j. Here $0 \leq x_k \leq 1$.

For convenience, consider a case where the quality of service ratings are discretized, e.g., to one of the 11 values in $\{0.0, 0.1, \ldots, 1.0\}$. Let h be the number of episodes of interaction with agent A_j, where h is bounded by the allowed history H. Now if g of the latest h episodes have a quality of service equal to x_k, then $f(x_k) = g/H$.

Following Marsh [10], we define for each agent an upper and a lower threshold for trust ratings. For each agent A_i, there are two thresholds ω_i and Ω_i, where $0 \leq \omega_i \leq \Omega \leq 1$.

DEFINITION 3. *Given a series of responses from agent A_j, $S_j = \{s_{j1}, s_{j2}, \ldots, s_{jH}\}$, and the two thresholds ω_i and Ω_i of agent A_i, we compute the bpa toward A_j as $m(\{T\}) = \sum_{x_k=\Omega_i}^{1} f(x_k)$, $m(\{\neg T\}) = \sum_{0}^{x_k=\omega_i} f(x_k)$, and $m(\{T, \neg T\}) = \sum_{x_k=\omega_i}^{x_k=\Omega_i} f(x_k)$.*

2.3 Combining Belief Functions

When an agent has not interacted often enough with a correspondent, it must seek the testimonies of other witnesses. Next we discuss how to combine such evidence.

A subset $\hat{A}$ of a frame Θ is called a *focal element* of a belief function Bel over Θ if $m(\hat{A}) > 0$. Given two belief functions over the same frame of discernment but based on distinct bodies of evidence, Dempster's rule of combination enables us to compute a new belief function based on the combined evidence. For every subset $\hat{A}$ of Θ, Dempster's rule defines $m_1 \oplus m_2(\hat{A})$ to be the sum of all products of the form $m_1(X)m_2(Y)$, where X and Y run over all subsets whose intersection is $\hat{A}$. The commutativity of multiplication ensures that the rule yields the same value regardless of the order in which the functions are combined.

DEFINITION 4. *Let Bel_1 and Bel_2 be belief functions over Θ, with basic probability assignments m_1 and m_2, and focal elements $\hat{A}_1, \ldots, \hat{A}_k$, and $\hat{B}_1, \ldots, \hat{B}_l$, respectively. Suppose*

$$\textstyle\sum_{i,j,\hat{A}_i \cap \hat{B}_j = \phi} m_1(\hat{A}_i) m_2(\hat{B}_j) < 1$$

Then the function $m : 2^\Theta \mapsto [0,1]$ that is defined by

$$m(\phi) = 0, \textit{ and}$$

$$m(\hat{A}) = \frac{\sum_{i,j,\hat{A}_i \cap \hat{B}_j = \hat{A}} m_1(\hat{A}_i) m_2(\hat{B}_j)}{1 - \sum_{i,j,\hat{A}_i \cap \hat{B}_j = \phi} m_1(\hat{A}_i) m_2(\hat{B}_j)}$$

for all non-empty $\hat{A} \subset \Theta$ is a basic probability assignment [16].

Bel, the belief function given by m, is called the *orthogonal sum* of Bel_1 and Bel_2. It is written $\mathsf{Bel} = \mathsf{Bel}_1 \oplus \mathsf{Bel}_2$.

2.4 Trust Networks

It helps to distinguish between two kinds of beliefs: *local belief* and *total belief*. An agent's local belief about a correspondent is from direct interactions with it and can be propagated to others upon request. An agent's total belief about a correspondent combines the local belief (if any) with testimonies received from any witnesses. Total belief can be used for deciding whether the correspondent is trustworthy. To prevent non-well-founded cycles, we restrict agents from propagating their total beliefs.

To evaluate the trustworthiness of agent A_g, agent A_r will check if A_g is one of its acquaintances (Here A_r is called the requesting agent, and A_g is called the goal agent). If so, A_r will use its existing local belief to evaluate the trustworthiness of A_g. Otherwise, A_r will query its neighbors about A_g. When an agent receives a query about A_g's trustworthiness, it will check if A_g is one of its acquaintances. If yes, it will return the information about A_g; otherwise, it will return up to F referrals to A_r based on its past experiences, where F is the *branching factor*. A_r, if it chooses, can then query any of the referred agents.

A referral r to agent A_j returned from agent A_i is written as $\langle A_i, A_j \rangle$. A series of referrals makes a referral chain. Observing that shorter referral chains are more likely to be fruitful and accurate [7] and to limit the effort expended in pursuing referrals, we define *depthLimit* as the bound on the length of any referral chain. The referral process begins with A_r initially contacting a neighbor A_i, who then gives a referral, and so on. The process terminates in success when a rating is received and in failure when the depthLimit is reached or when it arrives at an agent neither gives an answer rating nor a referral.

Now suppose A_r wants to evaluate the trustworthiness of A_g, after a series of l referrals, a testimony about agent A_g is returned from agent A_j. Let the entire referral chain in this case be $\langle A_r, \ldots, A_j \rangle$, with length l. The depth of any agent A_i on the referral chain is its distance on the shortest path from the root to agent A_i, where the depth of the root or the requesting agent is zero.

A TrustNet is a representation built from the referral chains produced from A_r's query. It is used to systematically incorporate the testimonies of the various witnesses regarding a particular correspondent.

DEFINITION 5. *A TrustNet $TN(A_r, A_g, \mathbf{A}, R)$ is a directed graph, where $\mathbf{A}$ is a finite set of agents $\{A_1, \ldots, A_N\}$, and R is a set of referrals $\{r_1, \ldots, r_n\}$.*

Given a series of referrals $\{r_1, r_2, \ldots, r_n\}$, the requester A_r constructs a TrustNet TN by incorporating the each referral $r_i = \langle A_i, A_j \rangle$ into TN. Algorithm 1 describes the process of constructing a trust network from a series of referrals. The depth of a TrustNet TN is equal to $depthLimit$.

Algorithm 1 Constructing a trust network

```
Suppose agent $A_r$ is the requesting agent, $R$ is a series of referrals, and $\mathbf{A}$ is a finite set of agents being visited. For any referral $r = \langle A_i, A_j \rangle$, agent $A_r$ add $r$ to the trust networks $TN$
if ($depth(A_j) \leq depthLimit$) then
    if ($A_j \notin \mathbf{A}$) AND ($A_j$ returns a rating to $A_g$) then
        add $A_j$ to the set of witnesses
        record the belief rating from $A_j$
    else if $A_j \notin \mathbf{A}$ then
        Append r to the trust network
        Send a request to $A_j$
    else
        Ignore the referral r
    end if
end if
```

Suppose agent A_r wants to evaluate the trustworthiness of agent A_g, and $\{W_1, \ldots, W_L\}$ are a group of witnesses towards agent A_g. We now show how testimonies from witnesses can be incorporated into the trust rating of a given agent. Let τ_i and π_i be the belief functions corresponding to agent A_i's local and total beliefs, respectively.

DEFINITION 6. *Given a set of witnesses $\Delta = \{W_1, W_2, \ldots, W_L\}$, agent A_r will update its total belief value of agent A_g as follows*

$$\pi_r = \tau_1 \oplus \ldots \oplus \tau_L$$

3. DECEPTION

We consider the problem of deception when a witness gives the rating about the goal agent to the requesting agent. Figure 1 describes the process of testimony aggregation, where agent A collects the testimonies of witnesses W_1, W_2, and W_3 about agent B. The witnesses can be deceptive to varying degrees. For example, witness W_1 may rate agent B at 0.9, but produce a rating of 0.1 instead of 0.9. (Strictly, in our approach, the witnesses return belief functions instead of scalars, but deception can still occur.) If the deception is detected, future testimonies from a deceptive witness should have a reduced effect on the aggregated ratings.

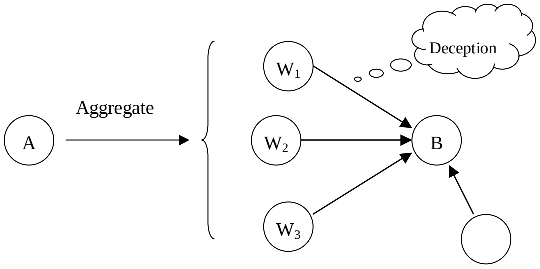

Figure 1: Deceptive witnesses can influence aggregated ratings

3.1 Deception Models

Suppose agent A_i considers the latest H episodes of interaction with agent A_j, with the true ratings of $S_j = \{x_1, x_2, \ldots, x_H\}$, where $1 \leq k \leq H$. Now A_i can be deceptive in providing a rating of A_j to others. We consider three kinds of deception: complementary, exaggerated positive, and exaggerated negative. Figure 2 shows the normal rating and three deception models. Below, α ($0 < \alpha < 1$) is the *exaggeration coefficient*.

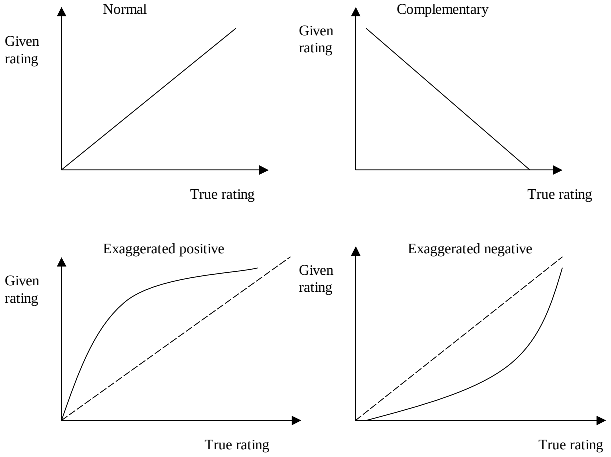

Figure 2: Normal rating and deception models

- *Normal:* for each rating x_k, the rating given is x_k.

- *Complementary:* for each rating x_k, the rating given is $1 - x_k$.
- *Exaggerated positive:* for each rating x_k, the rating given is $\alpha + x_k - \alpha x_k$.
- *Exaggerated negative:* for each rating x_k, the rating given is $x_k - \alpha x_k/(1-\alpha)$.

3.2 Weighted Majority Algorithm

The weighted majority algorithm (WMA) deals with how to make an improving series of predictions based on a set of advisors [9]. The first idea is to assign weights to the advisors and to make a prediction based on the weighted sum of the ratings provided by them. The second idea is to tune the weights after an unsuccessful prediction so that the relative weight assigned to the successful advisors is increased and the relative weight assigned to the unsuccessful advisors is decreased. WMA applies to the combination of evidence without regard to the reasoning on which the individual ratings might be based.

We adapt the algorithm to predict the trustworthiness of a given party based on a set of testimonies from the witnesses. Each agent maintains a weight for each of the other agents whose testimonies it requests. This weight estimates how credible the given witness is. However, applying the classical WMA for reputation management presents a technical challenge, because the ratings received from witnesses are not scalars, but belief functions. Therefore, our approach extends the WMA to accommodate belief functions. In simple terms, our approach maps belief functions to probabilities so that we can compute the difference between a prediction and the observed trustworthiness and accordingly update the weights for each witness. Moreover, we study the number of witnesses found in different settings (depth of trust networks, branching factor, and simulation cycles), the effect of the number of witnesses on the prediction, and the evolution of trust networks.

To motivate our approach, we describe a variant of WMA called WMA Continuous (WMC). WMC allows the predictions of the algorithms to be chosen from the interval $[0,1]$, instead of being binary. The predictions of WMC are also chosen from the interval $[0,1]$. The term *trial* refers to an update step. We assume that the master algorithm is applied to a pool of n algorithms, letting x_i^j denote the prediction of the ith algorithm of the pool in trial j. Let λ^j denote the prediction of the master algorithm in trial j, ρ^j denote the result of trial j and $w_1^j, \ldots, w_n^j$ denoted the weights at the beginning of trial j. Consequently, $w_1^{(j+1)}, \ldots, w_n^{(j+1)}$ denoted the weights following the final trial. We assume that all initial weights w_i^1 are positive. Let $s^j = \sum_{i=1}^n w_i^j$.

Prediction: The prediction of the master algorithm is

$$\lambda^j = \frac{\sum_{i=1}^n w_i^j x_i^j}{s^j}.$$

Update: For each algorithm in the pool, the weight w_i^j is multiplied by some factor θ that depends on β, x_i^j, and ρ^j.

$$w_i^{(j+1)} = \theta w_i^j,$$

where θ can be any factor that satisfies

$$\beta^{|x_i^j - \rho^j|} \leq \theta \leq 1 - (1-\beta)|x_i^j - \rho^j|$$

3.3 Deception Detection

Now we introduce a version of WMC geared to belief functions. Suppose agent A_r wishes to evaluate the trustworthiness of agent A_g. Our algorithm is given from the perspective of A_r. Let $\{W_1, \ldots, W_L\}$ be a set of witnesses that A_r has discovered for agent A_g. Let A_r assign a weight w_i to witness W_i. All the weights are initialized to 1.

Let the belief rating given by W_i to A_g be $m_i(\{T\})$, $m_i(\{\neg T\})$, $m_i(\{T, \neg T\})$. Then the new belief rating is

$$\begin{aligned} m_i'(\{T\}) &= w_i * m_i(\{T\}) \\ m_i'(\{\neg T\}) &= w_i * m_i(\{\neg T\}) \\ m_i'(\{T, \neg T\}) &= 1 - m_i'(\{T\}) - m_i'(\{\neg T\}). \end{aligned}$$

Prediction: Suppose the set of witnesses is $\{W_1, \ldots, W_L\}$, then the total belief of agent A_r for agent A_g is

$$\pi_r = \tau_1' \oplus \ldots \oplus \tau_L' \tag{1}$$

for any witness W_i, $\tau_i' = \{m_i'(\{T\}), m_i'(\{\neg T\}), m_i'(\{T, \neg T\})\}$.

Update: Since the prediction is in the form of a belief function, we cannot use WMA directly. Therefore, we first convert the belief function to the probabilities of T and $\neg T$. Next, we compute the difference between the prediction and the true rating of A_r.

Below is our approach to convert a belief function to probabilities. Suppose $m_i(\{T\})$, $m_i(\{\neg T\})$, $m_i(\{T, \neg T\})$ are the new belief ratings without considering the weight of the witness W_i. Then define a likelihood rating of T and $\neg T$ as follows

$$\begin{aligned} q_i(\{T\}) &= m_i(\{T\}) + m_i(\{T, \neg T\}) \\ q_i(\{\neg T\}) &= m_i(\{\neg T\}) + m_i(\{T, \neg T\}) \end{aligned}$$

The probabilities of T and $\neg T$ are

$$\begin{aligned} prob_i(\{T\}) &= \frac{q_i(\{T\})}{q_i(\{T\}) + q_i(\{\neg T\})} \\ prob_i(\{\neg T\}) &= \frac{q_i(\{\neg T\})}{q_i(\{T\}) + q_i(\{\neg T\})} \end{aligned}$$

The probability of T from witness W_i is

$$prob_i(\{T\}) = \frac{m_i'(\{T\}) + m_i'(\{T, \neg T\})}{1 + m_i'(\{T, \neg T\})}. \tag{2}$$

Suppose ρ is the rating from A_r, where $0 \leq \rho \leq 1$. Then the weight of witness W_i will be updated as follows.

$$w_i' = \theta w_i$$

where θ can be any factor that satisfies

$$\theta = 1 - (1-\beta)|prob_i(\{T\}) - \rho|. \tag{3}$$

The above formula can be simplified as follows if $\beta = 0.5$.

$$1 - \frac{|prob_i(\{T\}) - \rho|}{2} \tag{4}$$

Figure 3 shows the values of θ for different values of $prob(\{T\})$ when ρ = 0.1, 0.5, and 0.9.

Algorithm 2 Deception detection by requesting agent A_r

```
Initialize all witness weights to 1
Let $A_g$ be the goal agent
Let $\{W_1, \ldots, W_n\}$ be the witnesses found by $A_r$ for $A_g$ by applying Algorithm 1
Generate a prediction $\lambda$ as specified in Equation 1
for Each witness $W_i$ do
    Compute a probability from the belief function testimony of $W_i$ according to Equation 2
    Update the weight $w_i$ according to Equation 3
end for
```

Algorithm 2 describes the algorithm used by each agent to adjust the weights of the witnesses it uses.

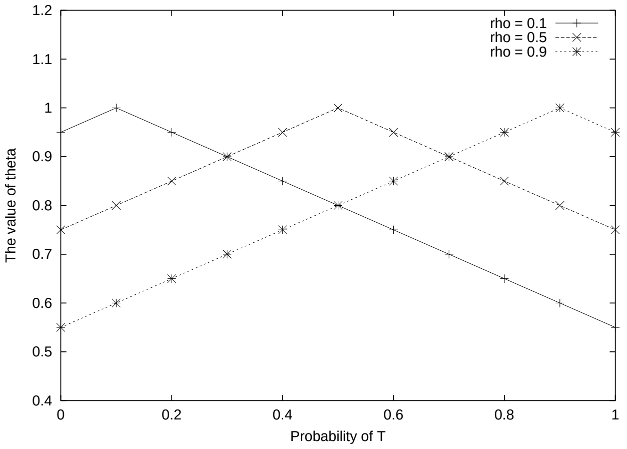

Figure 3: The value of θ when ρ = 0.1, 0.5, and 0.9

4. EXPERIMENTAL RESULTS

Our experiments are based on a simulation testbed we have developed. This testbed models the *interest* and *expertise* for each agent via term vectors of dimension 5. Roughly, the interest encodes what the agent's queries are like and the expertise encodes what the agent's responses are like. The closeness of the match between a response and a query translates to how high a rating is given by a querying agent to a responding agent (in a given query-response episode).

Briefly, the simulations proceed as follows. In each simulation cycle, we randomly designate an agent to be the querying agent. The queries are generated as vectors by perturbing the interest vector of the querying agent. When an agent receives a query, it may ignore the query, answer it based on its expertise vector, or refer to other agents. The originating agent collects all suggested referrals, and continues the process by contacting some of them. Finally, the referral process draws to an end. The querying agent agent aggregates the ratings based on weights it has assigned to the witnesses; the originating agent may decide to interact with the specified agent. Depending on the outcome of the interactions, the originating agent adjusts the weights it assigns to the witnesses involved.

Each agent keeps up to the 10 latest episodes of interactions with another agent. The agents are limited to having no more than 4 neighbors and 16 acquaintances. Queries are sent only to and referrals are given to neighbors. Periodically, each agent decides which acquaintances are promoted to become neighbors and which neighbors are demoted to be ordinary acquaintances.

The length of each referral chain is limited to 6. For any two agents A_i and A_j, the initial values of the belief function τ are defined as follows: $\tau_i(\{T_j\}) = \tau_i(\{\neg T_j\}) = 0$, $\tau_i(\{T_j, \neg T_j\}) = 1$. The sociability for all agents in the acquaintance models is initialized to 0.5. For each agent A_i, we set $\omega_i = 0.1$ and $\Omega_i = 0.5$.

We initialize the network of agents in the following manner. Following Watts and Strogatz [17], we begin from a ring but, unlike them, we allow for edges to be *directed*. We use a regular ring with 100 nodes, and 4 out-edges per node (to its neighbors) as a starting point for the experiment.

Of the total of 100 agents, 10 agents give complementary ratings, 10 agents exaggerate positive ratings ($\alpha = 0.1$), and 10 agents exaggerate negative ratings ($\alpha = 0.1$). The rest of the agents always give normal ratings. We randomly choose 10 agents as requesting agents. Each requesting agent A_i evaluates the trustworthiness of all agents except itself. After every 500 rounds, we compute the weights for the witnesses, and the rating error (i.e., the difference between the prediction and the true rating). The computation is not counted in the simulation cycle.

4.1 Metrics

We now define some useful metrics with which to intuitively capture the results of our experiments.

DEFINITION 7. *Suppose $\{W_1, \ldots, W_L\}$ are exactly L witnesses for agent A_g, then the total belief of agent A_r for agent A_g is*

$$\pi_r = \tau'_1 \oplus \ldots \oplus \tau'_L$$

for any witness W_i, τ'_i is the new belief ratings and is equal to $\{m'_i(\{T\}), m'_i(\{\neg T\}), m'_i(\{T, \neg T\})\}$. The rating error is defined as

$$|prob_r(\{T\}) - \rho|.$$

where $prob_r(\{T\}) = \frac{m'_r(\{T\}) + m'_r(\{T,\neg T\})}{1 + m'_r(\{T,\neg T\})}$, and ρ is the rating for agent A_g from agent A_r.

DEFINITION 8. *The average weight of a witness W is*

$$\Pi_W = 1/N \sum_{i=1}^{N} w_i,$$

where w_i is the weight of witness W from agent A_i's acquaintance model, and N is the total number of agents in whose acquaintance model W_i occurs.

4.2 Number of Witnesses

Our first experiment discusses the depth of trust networks and branching factor and their effects on the number of witnesses. The 10 chosen agents evaluate the trustworthiness of other agents. Figure 4 shows the average number of witnesses found at different depths with a branching factor F of 1, 2, 3, and 4, respectively, after 5000 simulation cycles. As intuitively expected, more witnesses are found when the requesting agent searches deeper and wider.

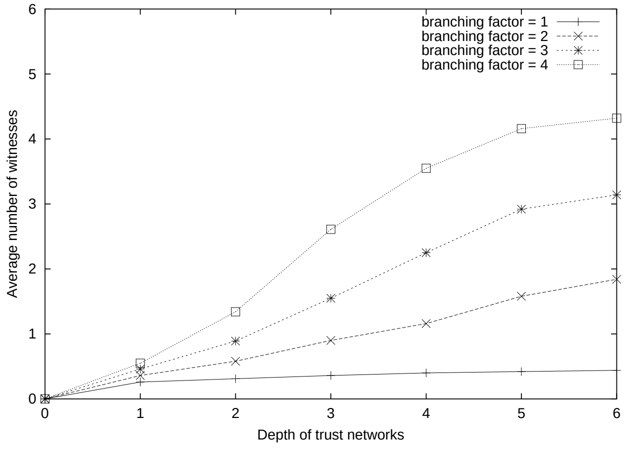

Figure 4: Average number of witnesses found for different depths with branching factor $F = 1, 2, 3, 4$ (after 5000 cycles)

More interestingly, the number of witnesses also depends on the queries (i.e., simulation cycles). The more the queries the more witnesses can be found in the trust networks with the same depth and branching factor. Figure 5 shows the total number of witnesses and the number of witnesses that can be found in the trust networks with depth six and branching factor four. With the help of trust networks, the requesting agent can only find 10% witnesses at simulation cycle zero. After 15000 cycles, the number increases to 50% (see Figure 6).

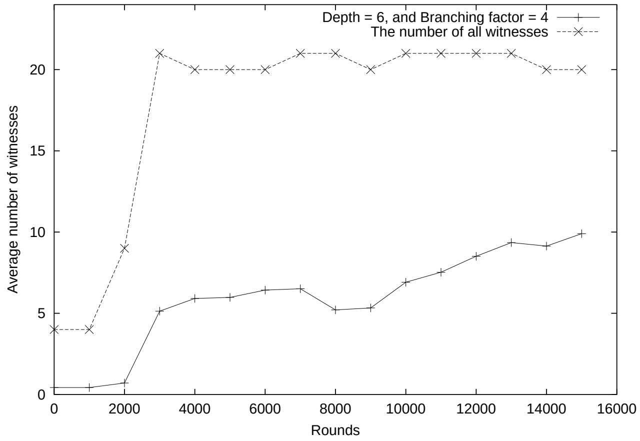

Figure 5: Average number of witnesses found from 0 to 15000 cycles

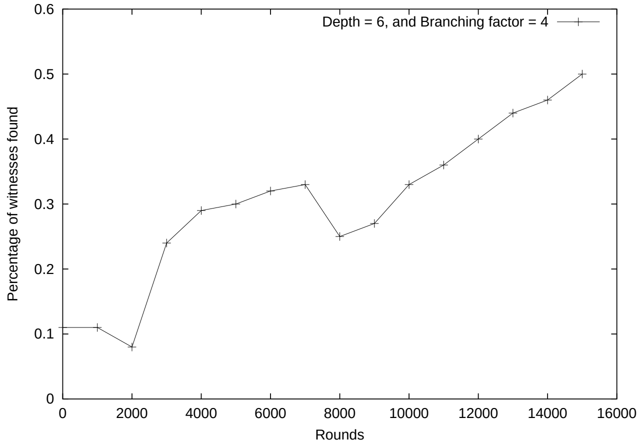

Figure 6: Percentage of witnesses found from 0 to 15000 cycles

We also study the effect of the number of witnesses on the accuracy of the prediction. Figure 7 shows the rating error for different numbers of witnesses at cycles 0, 2500, and 5000. We find that the number of witnesses does not affect the prediction accuracy much. The rating error only improves from 20% to 17% at 5000 simulation cycles when the number of witnesses increases from one to six. This is possibly due to a combination of two reasons. One, there are few (only 10%) witnesses who give complementary ratings. Two, the updating of weights dominates the number of witnesses (see below).

For the next two experiments, the depth six and branching factor four are applied in the trust networks.

4.3 Prediction

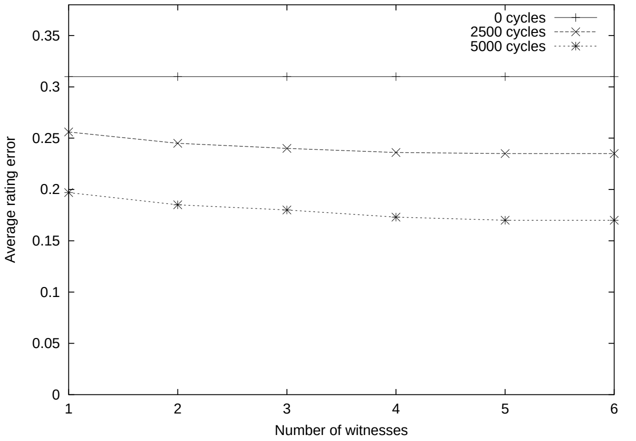

Figure 7: The rating error for different numbers of witnesses (at 0, 2500, and 5000 cycles)

Figure 7 tells us that the requesting agents can make better predictions through weight updating. For the same population and the same number of witnesses, the rating error changes from 0.31 to about 0.17 after 5000 simulation cycles. Figure 8 shows the whole process in greater detail. We compute the average rating error for the given sets of requesting agents and goal agents every 500 cycles. We find the average rating error becomes less then 0.05 after 8000 cycles.

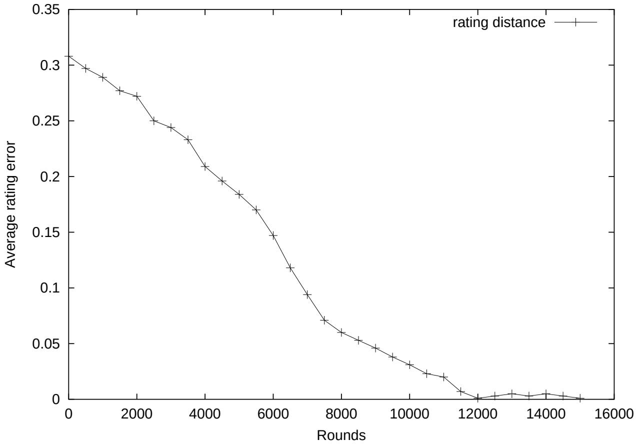

Figure 8: Average rating error during weight learning

4.4 Weights of Witnesses

The reason that the requesting agent can make better prediction is that it adjusts the weights for different types of witnesses. Therefore, the testimonies from lying witnesses will have less effect on the process of testimony aggregation. Figure 9 shows the change of average weights for different types of witnesses: normal, complementary, exaggerated positive, and exaggerated negative. We find the weights for witnesses with normal ratings are almost the same, but the weights for witnesses with complementary ratings change a lot. For the witnesses with complementary ratings, their average weights decrease from 1 to about 0.2 after 5000 cycles.

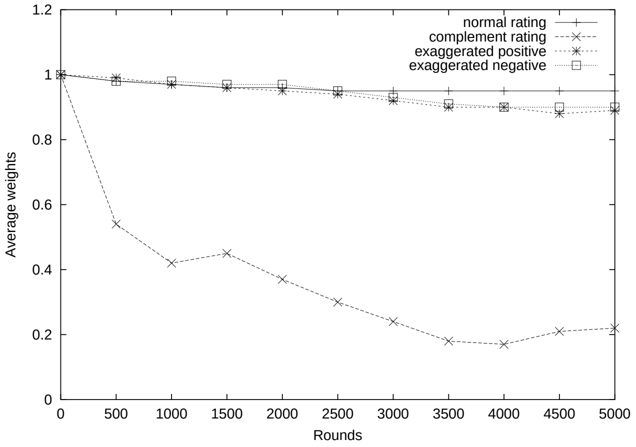

Figure 9: Average weights of witnesses for different deception models

The default exaggeration coefficient for witnesses with exaggerated positive or negative ratings is 0.1 in our previous experiments. The present experiment studies the average weights for such witnesses with different exaggeration coefficients. Figure 10 shows the average weights for witnesses with exaggerated negative ratings when exaggeration coefficient α is set to 0.1, 0.2, and 0.3, respectively. The results indicate that our approach can effectively detect witnesses lying to different degrees.

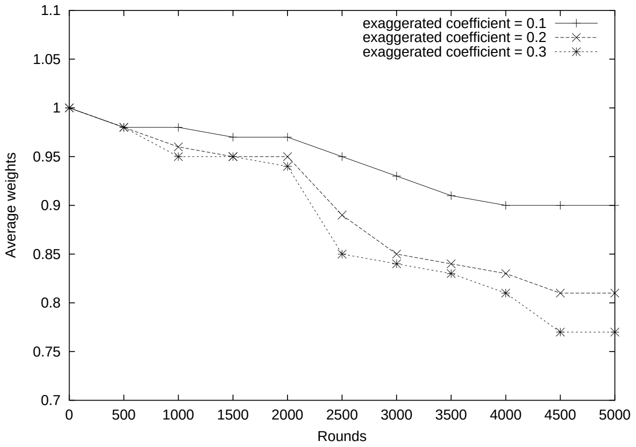

Figure 10: Average weights of witnesses for different exaggeration coefficients

5. RELATED WORK

One of the first works that tried to give a formal treatment of trust was that of Marsh [10]. His model attempted to integrate aspects of trust taken from sociology and psychology. Since Marsh's model has strong sociological foundations, the model is rather complex and cannot be easily used in today's electronic communities. Moreover the model only considers an agent's own experiences and doesn't involve any social mechanisms. Hence, a group of agents cannot collectively build up a reputation for others.

Rahman and Hailes [1] proposed an approach for trust in virtual communities. In simple terms, this is an adaptation of Marsh's work wherein some concepts were simplified (for example, trust can have only four possible values) and some were kept (such as situation or contexts). The main problem with Rahman and Hailes' approach is that every agent must keep rather complex data structures that represent a kind of global knowledge about the whole network. Usually maintaining and updating these data structures can be laborious and time-consuming.

Another, more computational, method is from *Social Interaction Framework* (SIF) [14]. In SIF, an agent evaluates the reputation of another agent based on direct observations as well through other *witnesses*. Schillo's work motivates some of our experiments for reputation management. However, SIF does not describe how to find such witnesses, whereas in the electronic communities, deals are brokered among people who often would never have met each other.

In our first work on this subject, we developed an approach for social reputation management, in which we represented an agent's belief ratings about another as a scalar and combined them with testimonies using combination schemes similar to certainty factors [18]. The drawbacks of the certainty factor models led us to consider alternate approaches, specifically an evidential model of reputation management based on the Dempster-Shafer theory [19], which is extended further by the present work to accommodate deception.

Aberer and Despotovic [2] simplified our model and use that to manage trust in a peer-to-peer network where no central database is available. Their model is based on binary trust, i.e., an agent is either trustworthy or not. In case a dishonest transaction is detected, the agents can forward their complaints to other agents. Aberer and Despotovic use a special data structure, namely the P-Grid, to store the complaints in a peer-to-peer network. In order to evaluate the trustworthiness of another agent B, an agent A searches the leaf level of the P-Grid for complaints on agent B.

Barber and Kim [3] present a multiagent belief revision algorithm based on belief networks. In their model the agent is able to evaluate incoming information and generate a consistent knowledge base, and to avoid fraudulent information from unreliable or deceptive information source or agents. Barber and Kim emphasize on modeling the reliability of information sources and maintaining the knowledge base of each agent, whereas we emphasize effectively detecting untrustworthy agents in a group.

Pujol *et al.* [12] propose an approach to establish reputation based on the position of each member within the corresponding social networks. Pujol *et al.* seek to reconstruct the social networks using available information in the community, and measure each member's reputation with an algorithm called *NodeRanking*, which can operate without knowing the entire graph. Pujol *et al.* view reputation roughly as the popularity of the node in the social networks, whereas we model reputation as the past experiences of members in the networks.

Sabater and Sierra [13] show how social network analysis can be used as part of the *Regret* reputation system, which considers the social dimension of reputation. They use a different approach to find the witnesses and calculate the witness reputation based on the subset of the selected sociogram over the agents that had interactions with the target agent. However, Sabater and Sierra apply some simple rules to decide the trustworthiness of the information from the witness, and do not consider deception and the effect of deception on information aggregation.

Mui *et al.* [11] summarize existing works on reputation across diverse disciplines, i.e., distributed artificial intelligence, economics, and evolutionary biology. They discuss the relative strength of the different notions of reputation using a simple simulation based on evolutionary game theory. Mui *et al* focus on the strategies of each agent, and do not consider gathering reputation information from other parties in the network.

Sen and Sajja [15] consider the situation where an agent uses the word-of-mouth reputation from other agents to select one of several service provider agents. Their mechanism allows the querying agent to select one of the high-performing service providers with a minimum probabilistic guarantee based on the reputation communicated by the agents who are queried. Sen and Sajja's algorithm can decide the number of agents being queried in order to meet the probabilistic guarantee.

Brainov and Sandholm [4] study the impact of trust on contracting in electronic commerce. Their approach shows that in order to maximize the amount of trade and of agents' utility functions, the seller's trust should be equal to the buyer's trustworthiness. Advanced payment contracts can eliminate inefficiency caused by asymmetric information about trust and improve the trustworthiness between sellers and buyers. By contrast, we focus on the computational model of distributed reputation management for electronic commerce and multiagent systems.

There has been much work on the cognitive view of trust. In the cognitive view, trust is made up of underlying beliefs, and trust is a function of the value of these beliefs [6, 5]. It is usually unlikely in a computational setting that an agent will have direct access to the mental state of another agent. Instead our approach is based on observation and concentrates on representations of trust, propagation algorithms, and formal analysis. However, the cognitive concepts explored by Castelfranchi and Falcone can be thought of as underlying and motivating the mechanisms we study here.

6. CONCLUSION

This paper studies the problem of deception in reputation management. We focus on how to effectively detect deception in the process of reputation information propagation and aggregation. Our approach helps an agent distinguish reliable witnesses from deceptive witnesses, and to minimize the effect of testimonies from deceptive witnesses. For simplicity, this work assumes that the witnesses behave in a consistent manner. However, it is easily seen that this approach applies to more complex kinds of deception, e.g., not lying to all agents, or lying with a certain probability. For the first case, the weights given to an agent by others depend on whether this agent lies to them or not. Probabilistic lying would dilute the weights as well.

Reputation management is related to trust. Reputation is important in making effective and informed trust decisions. In future work, we plan to integrate these mechanisms in the design of multiagent systems and electronic commerce systems. We will also study decision-making models and the evolution of different strategies of each agent, e.g., how an agent can adapt its strategy to the dynamic social structure of the given multiagent system and whether an agent should trust another agent based on the collected reputation information.

7. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of Hawaii International Conference on Systems Science 33*, 2000.

[2] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01)*, pages 310–317, 2001.

[3] K. S. Barber and J. Kim. Belief revision process based on trust: Simulation experiments. In *Proceedings of Autonomous Agents '01 Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 1–12, May 2001.

[4] S. Brainov and T. Sandholm. Contracting with uncertain level of trust. In *Proceedings of the First International Conference on Electronic Commerce (EC'99)*, pages 15–21, 1999.

[5] C. Castelfranchi, R. Conte, and M. Paolucci. Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation*, 1(3), 1998.

[6] C. Castelfranchi and R. Falcone. Principle of trust for MAS: cognitive anatomy, social importance, and quantification. In *Proceedings of Third International Conference on MultiAgent Systems*, pages 72–79, 1998.

[7] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *Proceedings of the National Conference on Artificial Intelligence*, pages 3–9, 1996.

[8] H. E. Kyburg. Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, 31:271–293, 1987.

[9] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[10] S. P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling, Apr. 1994.

[11] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 280–287, 2002.

[12] J. M. Pujol, R. Sanguesa, and J. Delgado. Extracting reputation in multiagent systems by means of social network topology. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 467–474, 2002.

[13] J. Sabater and C. Sierra. Reputation and social network analysis in multiagent systems. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 475–482, 2002.

[14] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 14:825–848, 2000.

[15] S. Sen and N. Sajja. Robustness of reputation-based trust: boolean case. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 288–293, 2002.

[16] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[17] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[18] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165, 2000.

[19] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 294–301, 2002.

[20] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14:881–908, 2000.