Robustness of reputation-based trust: Boolean Case

Sandip Sen
Math & CS Department
University of Tulsa
Tulsa, OK, USA
sandip@kolkata.mcs.utulsa.edu

Neelima Sajja
Math & CS Department
University of Tulsa
Tulsa, OK, USA
sajjane@ens.utulsa.edu

ABSTRACT

We consider the problem of user agents selecting processor agents to processor tasks. We assume that processor agents are drawn from two populations: high and low-performing processors with different averages but similar variance in performance. For selecting a processor, a user agent queries other user agents for their high/low rating of different processors. We assume that a known percentage of "liar" users, who give inverse estimates of processors. We develop a trust mechanism that determines the number of users to query given a target guarantee threshold likelihood of choosing high-performance processors in the face of such "noisy" reputation mechanisms. We evaluate the robustness of this reputation-based trusting mechanism over varying environmental parameters like percentage of liars, performance difference and variances for high and low-performing agents, learning rates, etc.

1. INTRODUCTION

Trust can be a critical parameter in interaction decisions of autonomous agents [5, 8]. We believe that in the dynamic, open societies, agents will have to routinely interact with other entities about which they have little or no information. It is also likely that often an agent will have to select one or few of several such less familiar entities or agents. The decision to interact or enter into partnerships can be critical both to the short term utility and in some cases long term viability of agents in open systems.

There can be various combinations of prior and experiential knowledge that an agent can use to make interaction or partner selection decisions. It can also use reputation mechanisms to decide on who to interact with. Such reputation mechanisms assume the presence of other agent who can provide ratings for other agents that are reflective of the performance or behavior of the corresponding agents. An agent can use such social reputation mechanisms to select or probe possibly fruitful partnerships.

Reputation based service and product selection has proved to be a great service for online users. Well-known sites like e-Bay [6] and Amazon [2], for example, provide recommendations for items, ratings of sellers, etc. A host of reputation mechanism variants are used at various other Internet sites [11]. Significant research efforts are under way, both in the academia and industrial research labs, that allow users to make informed decisions based on peer level recommendations. Most of this research develops and analyses collaborative filtering techniques [4, 7]. The above approaches assume that a user can be categorized into one of several groups and the choices made by the other members of the group can be used to predict the choice that would be preferred by the given user. The onus is on finding the appropriate group for a given user.

Our current work is motivated by a complementary problem. We assume that a user has identified one of several agents that can provide a service that it needs. The performance of the service providers, however, varies significantly, and the user is interested in selecting one of the service providers with high performance. As it lacks prior knowledge of the performances of the different providers, the user polls a group of other users who have knowledge about the performances of the service providers. The ratings provided by the other users constitute reputations for the service providers. An agent trusts, or selects to interact with, the service providers who have higher reputation.

In this paper we will evaluate the robustness of such reputation based trusting mechanisms by analyzing the effect of deceitful or lying user agents who provide false ratings about service providers when queried. We develop a trust mechanism that selects the number of agents to query to ensure, with a given probabilistic guarantee, that it is selecting a high-performing service provider. The mechanism uses the knowledge of the percentage of agents in the population that is expected to be such deceitful agents. We present results from a series of experiments varying the percentage of liars in the population, probabilistic guarantee thresholds, performance level differences of high and low performance service providers, performance variations of the service providers, level of observability, error in estimates of the liar population, etc. Results show that our proposed reputation based trust mechanism exhibits a graceful degradation property with a gentle fall-off in system performance as the liar population is increased until that population becomes a majority in the population. The mechanism is also robust to other problems like limited observability and incorrent estimates of the liar population.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

2. PROBLEMS OF REPUTATION-BASED TRUST MECHANISMS

There are a few caveats to the approach mentioned above, which on the first glance appears a reasonable thing to do. A minor problem is that the performance of service providers are noisy, i.e., their performance varies from time to time due to environmental variables which cannot be observed by the users. Thus depending on the quality estimation process used, different users may arrive at significantly varying estimates of performance of the same service provider. Secondly, different users may be able to observe different instances of the performance of a given service provider. This means that they are drawing their inference about the same provider from different, possibly overlapping, sets of experiences.

A more important problem is the presence of deceitful agents in the user population. There can be a host of different reasons for the existing of users that provide false ratings when queried. We will assume that a given percentage of users can provide false ratings and an analysis of why and how agents decide to "lie" is beyond the scope of this paper. A lying user agent can both provide poor estimates for good suppliers and good estimates for poor suppliers. Such pervasive and repeated deceitful behavior can severely affect the viability of gullible user agents who can wind up selecting low-performing service providers a significant fraction of the time.

3. A PROBABILISTIC REPUTATION MECHANISM

We assume the following framework of interaction of the user agent group:

- a population of N user agents,
- a population of P service providers,
- $l \leq \frac{N}{2}$ are liars, i.e., agents who give inverted ratings of producer agents,
- g, is the probabilistic guarantee threshold; we require that a service provider selection mechanism should be able to guarantee that the likelihood of selecting a high-performance service provider is at least g given l and N,
- b is the number of user agents that gets to know about the performance of a provider agent when it performs a task for any user agent. The observations are noisy, i.e., the observations differ somewhat from the actual performance which is conveyed accurately only to the user agent whose task was performed.

Each user agent updates its rating of a service provider every time it either directly interacts with is by assigning a task, or gets to observe its performance on as task assigned by another user agent. The following reinforcement learning [14] based action update rules are used for updating the estimate e_{ij}^{t+1} (the ith agent's estimate of the jth service provider after t interactions and observations):

$$e_{ij}^{t+1} = (1 - \alpha_i)e_{ij}^t + \alpha_i r_t,$$

$$e_{ij}^{t+1} = (1 - \alpha_o)e_{ij}^t + \alpha_o r_t,$$

where r_t is the performance received or observed and α_i and α_o are interaction and observation specific learning rates respectively. The learning rate values are selected in the range $(0, 1]$ and following the constraint $\alpha_i > \alpha_o$, i.e., direct interaction should affect performance estimates more than observations. This is particularly necessary because of the noise underlying observations.

The service provider agents are one of two types: high and low performers. The actual performances of the service providers are drawn from truncated Gaussian distributions returning values in the range $[0, 1]$. Each high-performing service provider agent has the same performance mean, μ_H. Similarly, each high-performing service provider agent has the same mean, μ_L. Both high and low performing service agents have the same standard deviation, σ, of performance. If $\mu_H - \mu_L$ is decreased and σ_p is increased it becomes more difficult to differentiate between high and low-performing agents based just on their performances on individual tasks. For a given interaction instance, let v be the performance value generated from the performance distribution of the corresponding service provider. Then the user agent who assigned this task observes a performance of v. But the b observers to this event observes performance values drawn from a Gaussian distribution with mean v and standard deviation σ_o.

When a user agent queries another user agent about the performance of a given provider agent, the queried agent returns a boolean answer which corresponds to a high or low rating for the service provider. We assume that liar agents lie consistently. That means every time they are queried they return a high rating for a provider if they believe it is a low-performing service provider and vice versa. We have assumed that each agent knows that $\mu_H > 0.5$ and $\mu_L < 0.5$, Accordingly, a user agent i believes that a provider agent j is a low-performer if $e_{ij}^t < 0.5$, or a high-performer if $e_{ij}^t > 0.5$ after interacting with the provider or observing its performance t times.

Given the guarantee threshold g and the liar and total population sizes, l and N respectively, we now present a mechanism for deciding how many user agents should be queried and how to select a provider based on their recommendations. Let q be the number of user agents that a given user agent will query to make a decision about which provider agent to choose. The algorithm for selecting a service provider is given in Figure 1. Note that the algorithm is not optimized. For example, the selection of agents is not biased towards those who gate a high rating from more user agents. Rather all agents, such that at least a majority of the users queried rate an agent high, have the same selection probability. We used this simpler approach initially to test the robustness of the criteria to select the number of agents to query. Note also that the q agents to query are selected randomly from the population of user agents as in our model there is no explicit rating of the user agents regarding whether they are truthful or not. This can easily be added, but that is not the focus of our task as we have discussed earlier. The `computAgentsTOQuery` function in the algorithm calculates the lowest q value for which the

```
Procedure SelectProvider(N,P,l,g)
{
  create empty lists of Preferred, Uncertain agent lists
  numPreferred <-- 0
  numUncertain <-- 0
  q <-- computeAgentsToQuery(N,l,g) // Calculates # agents to query
  Q <-- selectUsers(q,N) // randomly select n out of N user agents
  for each i in P // for each service provider
  {
     highCount <-- 0
     lowCount <-- 0
     for each j in Q // for each of the selected user agents to query
     {
       if (rating(j,i)) // if j gives good rating for i
         highCount++
       else
         lowCount++
     }
     if (highCount > lowCount) // majority rates i as good
     {
       numPreferred++
       include i in the Preferred list
     }
     else
       if (highCount == lowCount)
       {
         numUncertain++
         include i in the numUncertain list
       }
  }
  if(numPreferred>0)
    return service provider selected randomly from the numPreferred list
  if (numUncertain>0)
    return service provider selected randomly from the numUncertain list
  return service provider selected randomly from the entire population P
}
```

Figure 1: Service provider selection algorithm

.

following inequality holds:

$$\sum_{i=\max(\lceil \frac{q}{2} \rceil, \lfloor \frac{q}{2} \rfloor + 1)}^{P} \frac{\binom{N-l}{i}\binom{l}{q-i}}{\binom{N}{q}} \geq g.$$

The summation represents the probability that at least a majority of the q selected agents are non-liars. We propose to query the minimum number of agents for which this probability is greater than the required guaranteed, g. We can increase the robustness of the query mechanism by using more than the minimum q value calculated as above, but that would incur additional communication costs and is redundant if the requirement is to only meet the provided guarantee.

4. EXPERIMENTAL RESULTS

We assume that $\forall i, j$, $e_{ij}^0 = 0.5$, i.e., user agents start off with neutral estimates of provider agents. We performed a series of experiments by varying the number of liars for different guarantee thresholds, the spread between μ_H and μ_L, the standard deviation in performance σ_p, the number of agents who can observe an interaction, and the estimation error of the number of liars in the population (i.e., the querying user agent believes there are less liars in the population than the actual number).

4.1 Varying number of liars with different guarantee threshold

Figure 2 presents the average performance over all interactions when the guarantee threshold is varied for different number of liars. For a guarantee threshold of 0.95 the agents appear to be able to withstand the increase in liar population until they become so numerous that the required number of agents to query increases beyond the population size. This happens at around $l = 16$ and thereafter the performance starts decreasing with further increase in liar population. The same trend is observed for other plots as well.

The performance of the population with g =0.5 and 0.8 (corresponds to 50 and 80% on the plots) are initially identical because they choose the same q value, i.e., in both cases the same number of agents are queried. This happens because there are too few liars. The curves separate after the

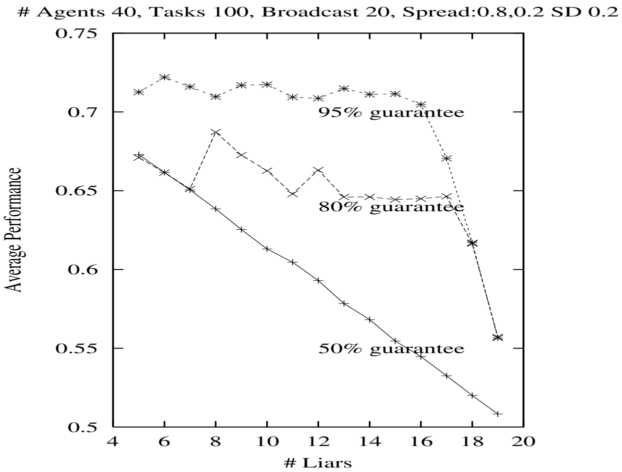

Figure 2: Performance variation with different probabilistic guarantee thresholds.

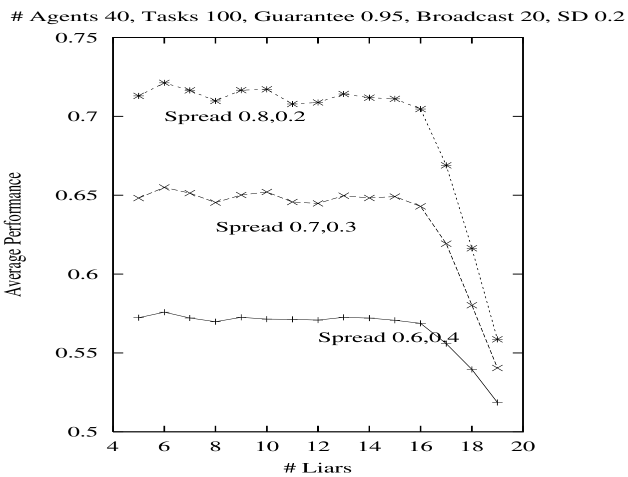

Figure 3: Performance variation with increasing spread between the performance of high and low-performance providers.

liar population increases beyond 7, when higher guarantees require querying more agents.

This plots demonstrate that the selection procedure prescribed in this paper works well and maintains a steady performance even with increasing liar population. The robustness of our simple probabilistic scheme was surprising and encouraging at the same time.

4.2 Varying the spread between high and low performers

Figure 3 plots the average performance of the population when the high and low means were set to the following pairs: (0.8, 0.2), (0.7,0.3), and (0.6,0.4). As the spread decreased, it was more difficult to accurately identify high performers. The performance also suffered because the level of performance of the high performers decreased.

4.3 Varying the standard deviation of the performers

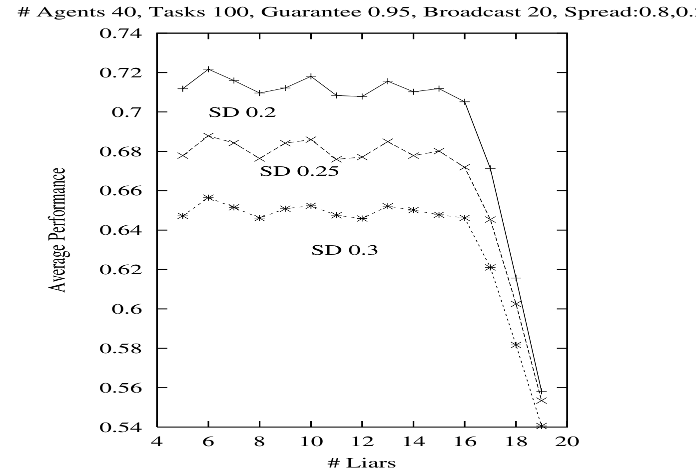

Figure 4: Performance variation with increasing variability in provider performance.

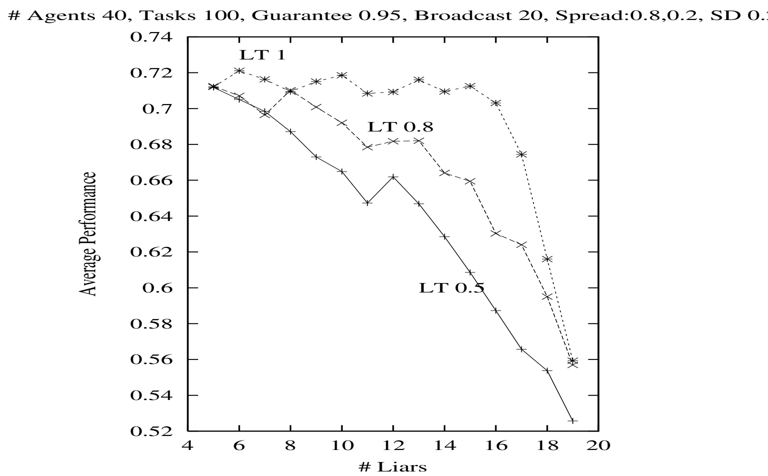

Figure 5: Performance variation with error in estimate of liars in population.

Figure 4 plots the variation in performance as the standard deviation in the provider performance is increased keeping their means constant. With increasing standard deviation performance decreased for reasons stated as above.

4.4 Varying the error estimate of the number of liars

Figure 5 plots average performance while varying the difference between the actual and estimated number of liars in the population. As the estimate, given as a fraction of the actual liar population, decreased, performance worsened as guarantees were undershot by larger values.

4.5 Varying the number of observers

Figure 6 plots the average performance while varying the number of user agents who can observe a given interaction. As the number of observers decreased, user agents had poorer estimates of the provider performances, and this led to worsening average performance.

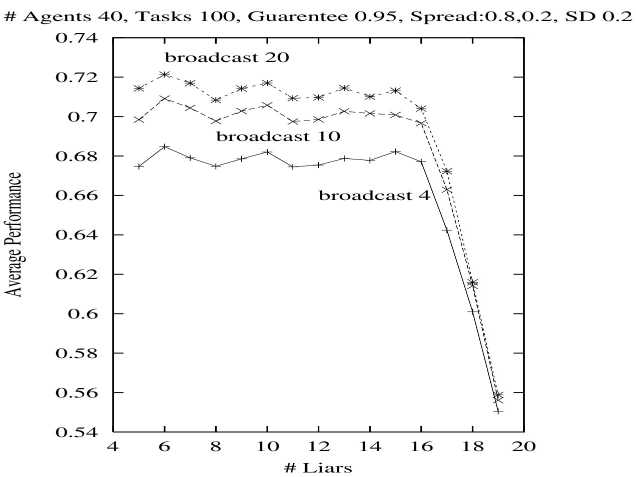

Figure 6: Performance variation for different agents observing an interaction.

5. RELATED WORK

The recent literature on evaluating various aspects of the concept of trust in computational agent systems is quite rich. Here we briefly visit some of the representative work in the area without attempting to be comprehensive in coverage. Zacharia and Maes have proposed a framework by which agents can participate in online communities and develop a reputation over time based on their performance in providing useful recommendations [17]. Barber and Kim have used a belief revision framework to motivate information sources to maintain consistent performance and avoid risking the fallout from a bad reputation in the user community [3]. Tan and Thoen present a generic model for trust in electronic commerce with dual emphasis on trusting the party with which a transaction is to be performed and trusting the infrastructure or mechanism that facilitates the execution of the transaction [15]. Schillo, Funk, and Rovatsos use a game-theoretic model of contracting and a probabilistic model of updating beliefs about other players to build a TrustNet [12]. Yu and Singh develop an extensive distributed reputation management model which develops and updates with experience a social network of trust based on referrals [16]. Sullivan, Grosz, and Kraus [13] evaluate the effectiveness of socially conscious agents, i.e., agents who care about their own reputation in the group, in generating high payoffs in collaborative groups.

Our work is in some sense simpler than some of the social reputation mechanisms [12, 16], but addresses a complementary problem of providing a probabilistic guarantee of selection of service providers given only summary statistics of the population distribution. As elaborate long-term modeling is not required, new agents to the community can immediately start using the reputation-based trust mechanisms without maintaining a lot of history and knowledge about the social network. Whereas performance can be improved by modeling the trustworthiness of recommending agents, the current work will enable user agents to make prudent selections in volatile groups as long as the percentage mix of lying and truthful user agents remains approximately constant.

The economists have studied problems of information asymmetry where the provider and the recipients of information have different viewpoints and incentives for revealing and interpreting information [1, 10]. In this paper, we have not dealt with the reason or motivation behind information assymetries between the querying agent and the rating agents. Rather, we have concentrated on developing a mechanism by which the user can make judicious selection of a service provider from the ratings provided by other agents.

6. CONCLUSIONS

In this paper we have considered the situation where a user agents uses the word-of-mouth reputations from other user agents to select one of several service provider agents. The goal is to provide a decision mechanism that allows the querying user to select one of the high-performing service providers with a minimum probabilistic guarantee. We provide an algorithm for determining which provider to trust based on the reputation communicated by the user agents who are queried. At the core of this algorithm is an equation to calculate the number of user agents to query to meet the prescribed probabilistic guarantee.

The mechanism is experimentally evaluated for robustness by varying a number of parameters in the domain. It is encouraging to see good performance over a range of liar population.

The model presented here is simple. It can easily enhanced to model the nature of user agents (whether they can be trusted or not), etc. But each of these extensions may limit the applicability of this mechanism, e.g., agents must be in a system for some time before they can effectively rate other agents.

One interesting extension is to use continuous, rather than boolean, ratings reported by user agents for providers. This will be particularly relevant if the provider agent performance means were drawn from a continuous, rather than a bipolar distribution. We plan to extend the current framework to handle this scenario.

It would be instructive to compare our approach to engineering techniques like the use of Kalman filters to estimate system parameters [9]. Kalman filters, however, are optimal estimators only under assumptions of white, Gaussian noise in measurements in systems that can be represented by linear models. These assumptions are violated in the "lying agents" scenario we have investigated in this paper.

Acknowledgments: This work has been supported in part by an NSF CAREER award IIS-9702672.

7. REFERENCES

[1] G.A. Akerlof. The market of lemons: Qualitative uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84:488–500, 1970.

[2] Amazon.com. URL: http://www.amazon.com/.

[3] K.S. Barber and J. Kim. Belief revision process based on trust: Agents evaluating reputation of information sources. In *Proceedings of the Agents-2000 Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 15–26, 2000.

[4] John S. Breeze, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[5] Cristiano Castelfranchi and Rino Falcone. Principles of trust for MAS: Cognitive autonomy, social importance, and quantification. In *Proceedings of the Third International Conference on Multiagent Systems*, pages 72–79, Los Alamitos, CA, 1998. IEEE Computer Society.

[6] ebay. URL: http://www.ebay.com/.

[7] Movie lens. URL:http://www.cs.umn.edu/Research/GroupLens/research.html/.

[8] S.P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994.

[9] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[10] Michael Rothschild and Joseph Stiglitz. Equlibrium in competitive insurance markets: An essay on the economics of imperfect information. *The Quarterly Journal of Economics*, 90(4):630–649, 1976.

[11] J.B. Schafer, J.Konstan, and J.Riedl. Electronic commerce recommender applications. *Journal of Data Mining and Knowledge Discovery*, 5:115–152, 2001.

[12] Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceiptful agents in artificial societies. *Applied Artificial Intelligence*, 14:825–848, 2000.

[13] David G. Sullivan, Barbara Grosz, and Sarit Kraus. Intention reconciliation by collaborative agents. In *Proceedings of the Fourth International Conference on Multiagent Systems*, pages 293–300, Los Alamitos, CA, 2000. IEEE Computer Society.

[14] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[15] Y.H. Tan and W. Thoen. An outline of a trust model for electronic commerce. *Applied Artificial Intelligence*, 114(8):849–862, 2000.

[16] Bin Yu and Munindar P. Singh. Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 125–137, 2001.

[17] Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14:881–908, 2000.