Selecting service providers based on reputation

Sandip Sen & Neelima Sajja
Department of Mathematical & Computer Sciences
University of Tulsa
e-mail:sandip@kolkata.mcs.utulsa.edu, sajjane@euler.mcs.utulsa.edu

Abstract

We consider the problem of user agents selecting service providers to process tasks. We assume that service providers are drawn from two populations: high and low-performing service providers with different averages but similar variance in performance. For selecting a service provider an user agent queries other user agents for their high/low rating of different service providers. We assume that there are a known percentage of "liar" users, who give false estimates of service providers. We develop a trust mechanism that determines the number of users to query given a target guarantee threshold likelihood of choosing high-performance service providers in the face of such "noisy" reputations. 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.

Introduction

Trust can be a critical parameter in interaction decisions of autonomous agents (Castelfranchi & Falcone 1998; Marsh 1994). We believe that in 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 for economic transactions. 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 (eBay) and Amazon (Amazon), for example, provide recommendations for items, ratings of sellers, etc. A host of reputation mechanism variants are used at various other Internet sites (Schafer, J.Konstan, & J.Riedl 2001). 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 (Breeze, Heckerman, & Kadie 1998; Grouplens). The above mentioned 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 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's goal is 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 what percentage of agents in the population 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 performance degradation as the liar population is increased, until that population becomes a majority in the population.

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 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; 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.

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 false 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 to whom the performance of a provider agent is broadcasted when the latter 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 the latter by assigning a task, or gets to observe the service provider's performance on task assigned by another user agent. The following reinforcement learning (Sutton & Barto 1998) based action update rules are used for updating the estimate e_{ij}^t (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 low-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 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 its updated performance value e_{ij}^t for the service provider after observing its performance t times. We assume that liar agents lie consistently and that all liar agents have the same lying approach. 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. The user agent receives q different performance values from other users for the same provider. To determine the true performance measure of the service provider, our goal is to group the ratings into two groups, one for the truthful estimates and the other for the lying ones. To find the means and s.d's of the two respective Gaussian distributions, we use the EM algorithm (Mitchell 1997) which divides the opinions given by the users into two different groups. The larger group is assumed to represent the nearest approximation to the provider's actual performance.

Given the guarantee threshold g and the liar and total population sizes, l and N respectively, we now present

```
Procedure SelectProvider(N,P,l,g)
{
  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
  {
     tempRatings[] <-- 0

     for each j in Q // for each of the selected user agents to query
       tempRatings[k++] <-- rating(j,i) // store users ratings of the provider

     finalRating[i] <-- EMAlgorithm(tempRatings); // final estimated rating of provider i


  }
  return service provider selected probabilistically from the top
  25% highly rated providers
}
```

Figure 1: Service provider selection algorithm.

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. In our previous work (Sen & Sajja 2002) we assumed that the users give boolean ratings for the service providers and used a simple unbiased algorithm for randomly selecting a high performer from the group of high performers i.e. providers whom the majority of the polled users rated highly.

In this work we use the continuous ratings given by the user agents for providers. As all users know that the provider's performance means are drawn from two Gaussian distributions, we use the EM algorithm which is a widely used approach to learning in the presence of unobserved variables also called the hidden variables z_{ij} (Mitchell 1997)(in our case the two means and the corresponding standard deviations from which the ratings are sampled). The algorithm searches for a maximum likelihood hypothesis by re-estimating the expected values of the probabilities that a given rating comes from one of two distributions, given its current hypothesis $\langle \mu_1, \mu_2, \sigma_1, \sigma_2 \rangle$, then recalculating the maximum likelihood hypothesis using these expected values for the means. The stepwise procedure of the algorithm is given below:

First initialize the maximum likelihood hypothesis to $h = \langle \mu_1, \mu_2, \sigma_1, \sigma_2 \rangle$, where μ_1, μ_2, σ_1 and σ_2 are arbitrary initial values. Then iteratively re-estimates h by repeating the following two steps until the procedure converges to a stationary value for h.

Step 1: Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming the current hypothesis $h = \langle \mu_1, \mu_2 \rangle$ holds. This $E[z_{ij}]$ is just the probability that instance x_i was generated by the jth normal distribution.

$$E[z_{ij}] = \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i | \mu = \mu_n)} = \frac{e^{-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma_n^2}(x_i - \mu_n)^2}}.$$

Step 2: Calculate a new maximum likelihood hypothesis $h' = \langle \mu_1', \mu_2' \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated in step 1. Then replace the hypothesis $h = \langle \mu_1, \mu_2, \sigma_1, \sigma_2 \rangle$ by the new hypothesis $h' = \langle \mu_1', \mu_2', \sigma_1', \sigma_2' \rangle$ and iterate.

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] x_i}{\sum_{i=1}^{m} E[z_{ij}]}.$$

$$\sigma_j \leftarrow \sqrt{\frac{\sum_{i=1}^{m} E[z_{ij}](x_i - \mu_j)^2}{\sum_{i=1}^{m} E[z_{ij}]}}.$$

The above algorithm for estimating the means of two Gaussian distributions illustrates the essence of EM approach: The current hypothesis is used to estimate the unobserved variables, and the expected values of these variables are then used to calculate an improved hypothesis. It can be proved that on each iteration through this loop, the EM algorithm increases the likelihood $P(\frac{D}{h})$ where D represents the data set of user ratings, unless it is at a local maximum. The algorithm thus converges to a local maximum likelihood hypothesis for $\langle \mu_1, \mu_2, \sigma_1, \sigma_2 \rangle$.

Once the means and standard deviations are estimated, the performance values given by the users are divided into two groups according to the maximum expected probabilities. The group with the majority is considered as the truthful group and the average of that

group's ratings is used as the estimated performance value of the provider.

Given the actual reputation values of the providers, the estimation and selection of the provider is given in Figure 1. 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 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.

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).

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. This happens because there are too few liars. The curves separate after the liar population increases beyond 7.

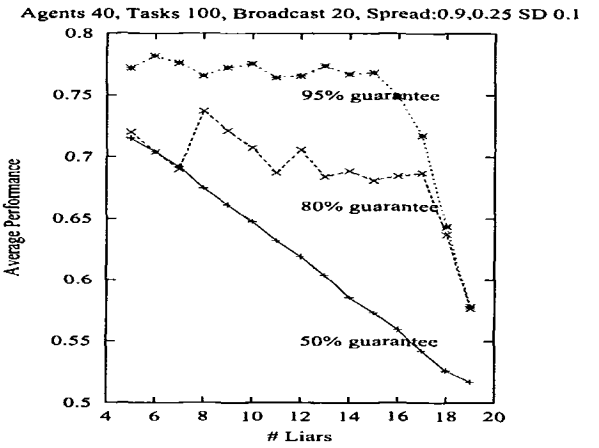

Figure 2: Performance variation with different probabilistic guarantee thresholds.

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.

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.

Varying the standard deviation of the performers

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.

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, as a fraction of the actual liar population, decreased performance worsened as guarantees were undershot by larger values.

Varying the number of agents who can observe an interaction

Figure 6 plots the average performance while varying the number of user agents who can observe a given in-

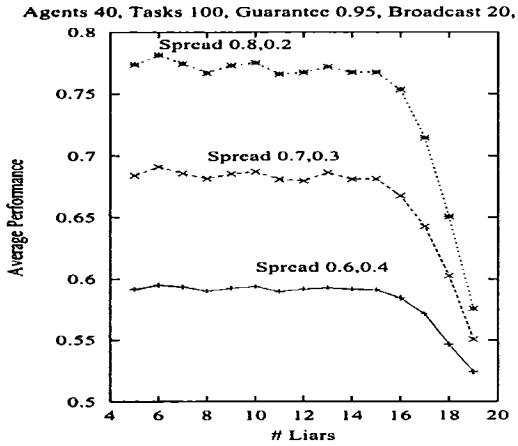

Figure 3: Performance variation with increasing spread between the performance of high and low-performance providers.

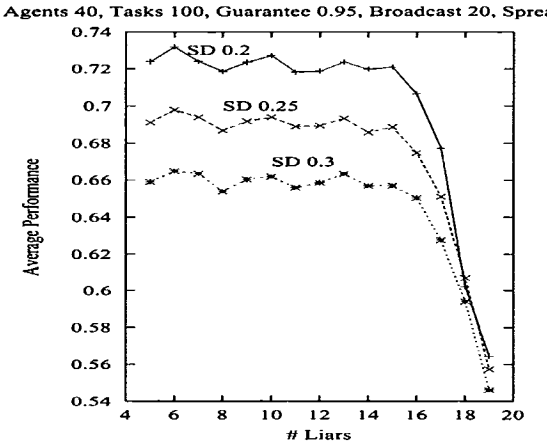

Figure 4: Performance variation with increasing variability in provider performance.

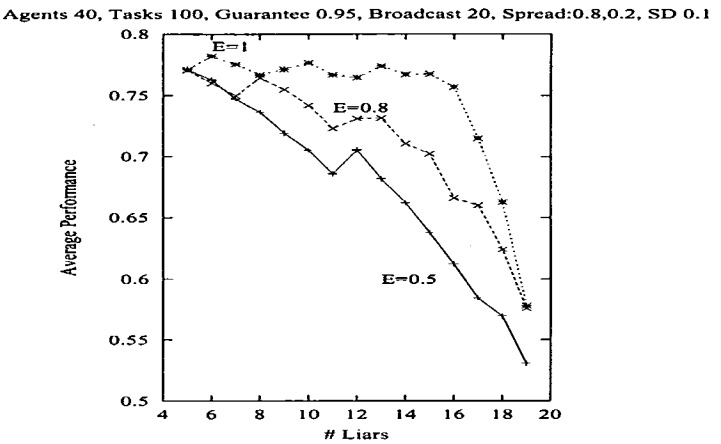

Figure 5: Performance variation with error in estimate of liars in population.

teraction. As the number of observers decreased, there is considerably less deviation in average performance.

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 (Zacharia & Maes 2000). 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 (Barber & Kim 2000). Tan and Then 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 (Tan & Thoen 2000). 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 (Schillo, Funk, & Rovatsos 2000). Yu and Singh develop an extensive distributed reputation management model which develops and updates with experience a social network of trust based on referrals (Yu & Singh 2001).

Our work is in some sense simpler than some of the social reputation mechanisms (Schillo, Funk, & Rovatsos 2000; Yu & Singh 2001), 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.

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 proba-

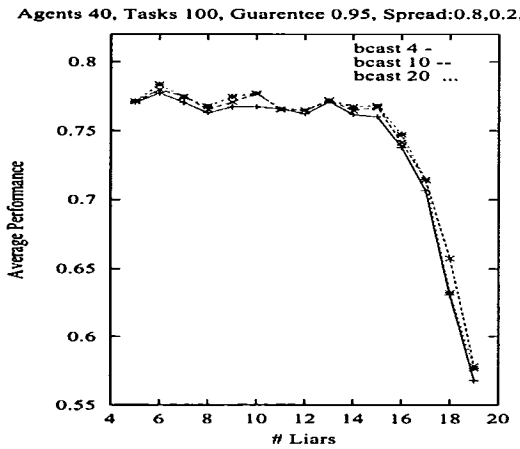

Figure 6: Performance variation for different agents observing an interaction.

bilistic 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. In addition to that we have used the EM algorithm to approximately estimate the providers performance from the responses of these user agents asked.

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.

References

Amazon.com. URL: http://www.amazon.com/.

Barber, K., and Kim, J. 2000. 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*, 15–26.

Breeze, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers.

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

ebay. URL: http://www.ebay.com/.

Movie lens. URL:http://www.cs.umn.edu/Research/GroupLens/research.ht

Marsh, S. 1994. *Formalising Trust as a Computational Concept.* Ph.D. Dissertation, University of Stirling.

Mitchell, T. 1997. *Machine Learning.* Boston, MA: WCB McGraw-Hill.

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

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

Sen, S., and Sajja, N. 2002. Robustness of reputation-based trust. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems [AAMAS 2002](to appear).*

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press.

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

Yu, B., and Singh, M. P. 2001. Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud, and Trust in Agent Societies*, 125–137.

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