Learning Task-Specific Trust Decisions

(Short Paper)

Ikpeme Erete
Georgia Institute of Technology
Atlanta, GA30332, USA
ikpeme@cc.gatech.edu

Erin Ferguson
University of Tulsa
Tulsa, OK74104
erin-ferguson@utulsa.edu

Sandip Sen
University of Tulsa
Tulsa, OK74104, USA
sandip@utulsa.edu

ABSTRACT

We study the problem of agents locating other agents that are both capable and willing to help complete assigned tasks. An agent incurs a fixed cost for each help request it sends out. To minimize this cost, the performance metric used in our work, an agent should learn based on past interactions to identify agents likely to help on a given task. We compare three trust mechanisms: success-based, learning-based, and random. We also consider different agent social attitudes: selfish, reciprocative, and helpful. We evaluate the performance of these social attitudes with both homogeneous and mixed societies. Our results show that learning-based trust decisions consistently performed better than other schemes. We also observed that the success rate is significantly better for reciprocative agents over selfish agents.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence-Multiagent systems

General Terms

Algorithms, Performance

Keywords

Trust decision mechanism, learning

1. INTRODUCTION

To facilitate collaboration and coordination in a multiagent system (MAS), agents need to be able to learn other agents' capabilities, preferences and goals. Previous research in agent learning focuses primarily on agents that are competent in one particular task type [5]. In reality, however, an agent's expertise can vary for different task types, and so our research framework takes into consideration varying levels of agent expertise.

The scenario we study is the following: an agent needs to find another agent to help it complete a given task. Each agent can perform a subset of the task types with varying competency levels. Finding an agent that can help depends on the task at hand, and the asking agent must be able to locate such an agent quickly. We assume that the asking agent incurs a fixed cost for each help request and needs to minimize this cost. Agent expertise is not advertised, and an agent does not directly know the expertise of the other agents in the system. Over time an agent's expertise and social attitude must be learned and the corresponding expertise model built using past interactions with the agent. The learning goal is for an individual agent to map any given task to a list of other agents ordered by their likelihood of helping with that task. We refer to this mapping as a trust decision mechanism. An agent would like to form these trust decisions quickly, while keeping the learning cost to a minimum.

An agent requesting help expects that the asked agent will help if it is able to perform the given task, but this is not always the case. We refer to an agent's attitude toward help requests as its social attitude. In our work, we evaluate three social attitudes: helpful, selfish and reciprocative. Helpful agents always help when they are capable, selfish agents never help, and reciprocative agents weigh factors such as cost of performing the given task and past interactions with the asking agent in determining whether to help.

So, in generating the ordered list of agents to help, each agent may consider past interactions and the current task. This list of agents, arranged based on perceived likelihood to help, can be viewed as an ordered ranking of which agents are most trustworthy.

Trust can be defined as "the subjective probability by which an individual expects that another individual will perform a given action on which its welfare depends" [4]. Using this definition, the act of asking another agent can be seen as a trust decision. An agent incurs a cost when asking another agent for help, and this cost is only worth expending if the asking agent believes that the other agent will respond positively. Should the trustee refuse to complete the task, because they are incapable or unwilling, the trustier suffers a loss and must ask the next agent in the ordered list.

The trust decision mechanism, and hence the order in which agents are asked, is instrumental in determining the total asking cost incurred. To minimize this, learning can be used. We compare a heuristic, success-based scheme based on frequency of past helps, with a more sophisticated learning-based scheme, which identifies agents that are likely to help for any given task type. We compare the effectiveness of these different trust mechanisms and a random agent selection scheme in minimizing asking cost. We varied other parameters that affected the expertise levels of agents and

Cite as: Learning Task-Specific Trust Decisions (Short Paper), Ikpeme Erete, Erin Ferguson, Sandip Sen, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

study the corresponding effects on their performance.

The rest of the paper is organized as follows: Section 2 describes the environment modeled and the various agents, including their decision making processes, social attitudes, and expertise modeling. Section 3 presents experimental results and analysis. Section 4 discusses the related work, and Section 5 presents conclusions and possible future work.

2. ENVIRONMENT AND AGENT MODEL

In this section, we present the model of our environment and details about agent properties and strategies. We simulated an environment where agents are assigned a set of tasks and seek help from other agents to complete them. The asking agent incurs a fixed cost for every help request, and the goal of each agent is to minimize total asking cost. To achieve this, an agent must locate, with as few attempts as possible, another agent that can and will provide help with an assigned task.

2.1 Environmental Setup

The environment is defined as the collection containing the set of agents and their corresponding expertise representations, the set of task descriptions, and a numerical value that is the asking cost.

2.2 Task Description

A task is described by a set of attribute-value pairs. Task descriptions are randomly generated based on the attributes and corresponding values of a domain. Task descriptions consist of eight attributes and their associated discrete values. Each of these attributes can define a feature of tasks such as task type, priority, duration and advance notice. Each of these attributes was given three possible associated values. The values of different attributes of a task are stored as a vector of pairs. For fairness, every agent in a system is assigned the same set of tasks.

2.2.1 Task Competency

Each agent possesses a given expertise profile which determines if it is able to perform a specific task and the associated cost. These competencies are stored in a decision tree that is stochastically built using domain attributes. Decision trees are an effective representation of agent expertise because they can represent, in an accessible format, disjunctions of conjunctions. So when a task description is presented to an agent, it traverses its decision trees based on the task attribute values until it reaches a decision at a leaf node: leaves are labeled "true" if it can perform such a task, or "false" if it cannot. These labels are determined using the probability $P_{success}$, which is the same for all agents and represents the probability that an agent can do any arbitrary task. Another probability used in the construction of this decision tree is P_{leaf}, the probability that any given node is a leaf. This can be used to control the depth of trees. By our construction, a deeper tree implies that an agent is more specialized in tasks: the deeper the tree, the more attributes are used to define the path to a specific leaf. Because a deeper task description with a positive outcome implies specialization, the cost to an agent of doing a task is inversely proportional to the length of the path.

2.3 Social Attitudes

The focus of our work is increasing the efficiency of locating another agent who is capable and willing to help with an assigned task. To emphasize this issue, we require our agents to seek help for an assigned task even if it has the ability to complete it. Put another way, an agent can always benefit if another agent performs a task on its behalf. An agent's response to a help request is influenced both by its expertise level and its social attitude. We use three types of agents in representing three common social attitudes: helpful, selfish and reciprocative. Helpful agents honor any help requests, and are therefore prone to exploitation in mixed societies. Selfish agents never honor a help request, and benefit in the presence of helpful agents. Reciprocative agents use a balance of cost and savings to determine whether to accept a given request. The probabilistic decision policy is based on Sen *et al*s work [6] and considers the cost of carrying out the given task, the average cost of tasks performed, and the balance of cost, i.e. the difference between help offered and help given over past interactions. The probability that a reciprocative agent will honor a help request is calculated as

$$Pr(i,k,j) = \frac{1}{1 + \exp\frac{C_{ij}^k - \beta \cdot C_{avg}^k - B_{ki}}{\tau}} \qquad (1)$$

where
$Pr(i,k,j)$: the probability that $agent_k$ will carry out task t_{ij} for $agent_i$,
C_{ij}^k: the cost of carrying out t_{ij} for $agent_k$,
C_{avg}^k: the average cost of tasks performed by $agent_k$,
B_{ki}: the balance of cost, and
β, τ: constants with which the social attitude can be tuned.

The use of cost balance in the function incorporates past interactions between the agents. The above function is a sigmoidal function where the probability of helping increases with balance and decreases with increasing task cost. The constants in the equation are used to bias the helping decision. A higher rate of β makes the agent more inclined to help initially, while τ can be used to control the steepness of the curve. Over time, a reciprocative agent can adapt different help attitudes with other agents in the environment.

2.3.1 Trust Mechanisms

We use three trust mechanisms in our experiments. The *random* mechanism is a baseline algorithm where an agent decides randomly whom to ask for help. The second is *success-based*: agents orders other agents by the frequency of accepted help requests by these agents in the past. The third mechanism uses *learning*: agents model other agents' help-giving behavior for each task type based on previous interactions with that agent. For a given task, the asking agent uses this model to predict the likelihood of the other agents helping with that specific task. We use a nearest neighbor algorithm to build such models and derive corresponding predictions.

In our approach, previous acceptance and denials of help request of the i^{th} agent, $agent_i$, is stored and indexed by the corresponding task descriptions. Given a new task description, m, we find the k nearest stored points, K, in the task space and the probability that $agent_i$ will agree to help with this task is calculated as the sum of the least mean square distances for those points in K marked as acceptances, divided by the sum of the distances for all points in

K. Therefore, as the number of positive responses from an agent increases and the sum of distances decreases, the predicted probability of helping increases. Agents with higher predicted probability for helping are approached with help requests first.

3. EXPERIMENTAL RESULTS

We conducted a series of experiments to study the effects of social attitude of an agent and trust mechanisms for modeling other agents' help decisions. Our goal is to evaluate if agents can effectively learn to select which other agents to trust given an assigned task. We use the asking cost as our primary performance metric. We ran experiments using two environments: a homogeneous environment where all agents use similar trust mechanisms, and a heterogeneous environment where different agents use different trust mechanisms.

3.1 Experimental setup

We simulated an environment with 10 agents, where each agent is assigned the same set of 3000 tasks for fairness of comparison and a value of k=5 is used in the nearest neighbor algorithm. The results are averaged over 10 runs.

3.1.1 Homogeneous Environment

We first compare performances of the social attitudes using similar trust mechanisms. Next we compare performances of each social attitude using different trust mechanisms and analyze the effects of increasing $P_{success}$.

Effect of Decision mechanism: We derive the following conclusions from results with different homogeneous groups as well as mixed groups of reciprocative and selfish agents using different trust mechanisms:

- Helpful agents perform better than the other types of agents with each trust decision mechanism. This is because these agents always accept help requests. The drawback to this approach is that it will not be effective in open groups containing exploitative agents.

- In groups comprising selfish and reciprocative agents, reciprocative agents identify selfish agents and rarely help such agents. The calculated rate of success to asking cost shows that only 0.038% of requests for help from selfish agents are honored irrespective of the trust mechanism used. The reciprocative agents' success rate is best with learning-based trust.

The performance of agents depends on how well they are able to learn effective trust policies. In general, all social attitudes except for selfish agents perform better when the trust mechanism is learning-based. Selfish agents perform better using random trust mechanism. This is due to its social attitude: selfish agents do not help others and after a while other agents learn not to help them. Within a few interactions with other agents, the selfish agents using trust mechanisms other than random have identified who to request help from. But other agents have identified them as selfish and refuse to assist. As agents who originally helped may no longer help, it is more effective to ask randomly.

Effects of varying $P_{success}$: Our next series of experiments is aimed at observing how the performance of agents using similar trust mechanisms varies with increasing expertise levels in the system. We expect that, with increasing agent expertise, an agent can complete more task types, leading to a reduction in total asking cost.

We observed results when varying the outcome probability, $P_{success}$, in the range $0.1 - 0.3$, for each social attitude and trust decision mechanism. We observed the following:

- With increasing $P_{success}$ the average asking cost decreases for helpful agents irrespective of the trust mechanism. The asking cost is lowest for learning-based, and highest for random.

- With learning and success-based schemes, the performance of selfish agents improve initially with increasing $P_{success}$ but then worsens for $P_{success} > 0.2$. The performance of random trust mechanism for selfish agents is lower than other algorithms, but is consistent throughout the probability range.

- Reciprocative agents perform better than selfish agents and random trust schemes perform worse than others.

- An increase in $P_{success}$ increases the number of positive responses from any agent. This results in an increase in the frequency of help requests to these agents. Consequently, there is a race condition between savings in cost and incurred cost, which produces a fluctuating trust relationship between agents. This explains the slight performance drop at higher values of $P_{success}$.

3.1.2 Groups with heterogeneous trust schemes

We also studied how agents would perform if the trust decision mechanisms were not the same for all agents in the system. We ran experiments with eleven helpful agents where the learning and success-based trusting schemes were used by four agents each and the remaining three agents asked randomly for help. Using the ratio of asking cost to success, we find that the learning-based trust schemes performs better than the other algorithms in the system even though they have the lowest number of successes. The differences in success rate, however, are not statistically significant.

3.1.3 Varying the dimension of task descriptions

In our experiments, tasks are described as a vector of eight attribute values. In one part, these attributes were changed to continuous numerical values in the range of zero to ten. The nearest neighbor algorithm of the agents was adapted to calculate the Euclidean distance between these points, which were stored in a kd-tree to allow more efficiency in accessing stored values.

We changed the number of attributes used to describe a task and observe the results. This was done in a homogeneous environment, to more clearly see the effect of increasing complexity of task descriptions on average asking cost. We found that, using 3,000 tasks, the number of dimensions does not significantly affect the asking cost per success of each trust mechanism. But it does affect the number of tasks that learning-based agents require to reach a lower asking cost than the random and success-based agents. This is because, while success-based and random agents do not consider the task description in deciding asking order and thus their strategies are unaffected by changes in the descriptions, learning-based agents require more trials to fill a sufficient training set when dealing with more complex descriptions.

4. RELATED WORK

One of the most challenging issues in open multiagent systems is the issue of trust and reputation among agents [1, 2]. Therefore, one of the critical research issues in multiagent systems involves how we learn to trust other agents. We now discuss some learning algorithms that have been used for this purpose in multiagent systems and some representative applications thereof.

Fullam [3] has shown how environmental rewards can be used to learn comprehensive trust strategies. In her work, the loss incurred by interacting with unreliable agents is mitigated by first obtaining reputation information of the agent. Therefore, the agent only loses a referral cost, instead of the full cost of services exchanged.

Other researchers, such as Sen [6], show that adaptive probabilistic reciprocity strategies can be used to develop and sustain trust and cooperation between self-interested agents. Even though reciprocity does not address the problem of task-specific learning, it shows that trust relations between agents can be developed in order to facilitate efficient decision making and identify exploitative agents. Sen's work shows that this probabilistic reciprocity scheme generates stable and cooperative relationships between self-interested agents with a fair distribution of the workload. Such reciprocal exchanges improve both individual and group performances in the long run.

Our work could be seen as a blend of Fullam and Sen's work. We used the adaptive reciprocity strategy by Sen to develop and maintain trust relationship between agents. We also used learning to develop trust models which agents use as a basis for initiating interaction with other agents.

In competitive situations such as negotiation and bargaining, agents interact without complete information about others' preferences. Given such a scenario, an agent can benefit from learning its opponent's decision model and thereby optimizing its utility. In our work each agent in the environment has skills that other agents are not initially aware of. But these agents must be able to discover models of the capabilities of other agents based on interaction and then use these models to optimize their performances.

5. CONCLUSIONS AND FUTURE WORK

We are interested in enabling agents to learn to form trusting, cooperative relationships with other agents in stable environments. We addressed the problem of quickly identifying collaborators based on specific task attributes. Asking for help can be time-consuming and costly, so agents must order probable collaborators by the likelihood of their being able and willing to help with a given task.

We compare our task-attribute based trust decision learning scheme with a success-based approach and a random selection process. We conducted a series of experiments to observe the performances of these agents in different environment settings and we highlight the following conclusions:

· Helpful agents generally perform better than agents with other social attitudes in homogeneous groups. This difference is highest when the trust mechanism used is learning-based. However, this social attitude will perform poorly in open groups.

· Selfish agents have very limited success in obtaining help from reciprocative agents. Their success percentage is independent of the trust mechanism, and these agents tend to perform slightly better when the trust mechanism is random.

· Learning-based approach dominates all other trust decision mechanisms. A reciprocal social attitude dominates selfish.

In the future we intend to use more sophisticated learning mechanisms to reduce the cost of locating collaborators. In particular, more comprehensive learning schemes that can handle continuous and nominal attributes need to be used for practical scenarios. We plan to incorporate referrals in the trust decision scheme. An interesting avenue would be to evaluate the effect of differing task costs from the requesting and asked agents' perspectives. The scale-up properties of our model and the effect of dynamic changes in the agents' capabilities can be studied.

Acknowledgment: An NSF award IIS-0209208, a DOD-ARO Grant #W911NF-05-1-0285 and Tulsa Undergraduate Research Challenge (TURC) program partially supported this work.

6. REFERENCES

[1] C. Castelfranchi and R. 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.

[2] S. D.Ramchurn, D. Huynh, and N. R. Jennings. Trust in multiagent system. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[3] K. Fullam. Learning complex trust decision strategies. In *Autonomous Agent and Multiagent Systems Conference*, page 1241, 2006.

[4] D. Gambetta. *Trust*. Basil Blackwell, Oxford, 1990.

[5] S. Saha, S. Sen, and P. S. Dutta. Helping based on future expectations. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Mulitagents System*, pages 289–296, 2003.

[6] S. Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Second International Conference on Multiagent Systems, AAAI Press, Menlo Park, CA*, pages 322–329, 1996.