A Bayesian Approach to Learning about the Opponent in Negotiation

Chhaya Mudgal, Julita Vassileva
Department of Computer Science
University of Saskatchewan, Canada
{chm906, jiv}@cs.usask.ca

Abstract:

Negotiation in multi-agent systems allows the agents to cooperate and coordinate their activities. The need for negotiation stems from interdependence of the agents upon each other due to the scarcity of resources. To assume that agents will be cooperative is not reasonable especially when agents are self-interested personal assistants. This paper describes an agent-based negotiation model applied for finding peer help in a workplace or educational setting. The objective is to achieve negotiation with incomplete (partial) information about the opponent. Having knowledge about the opponent is highly essential for a negotiation to be successful. According to the proposed model for negotiation the agents use decision theoretic approach to decide about the best action at every step of the negotiation process. To compensate for the incomplete information about the opponent, agents keep model of their opponent to predict their actions.

1. Introduction

[Maes, 1997] and [Bui et al., 1996] proposed using agents as smart personal assistants who can reduce the information and the workload of their users by locating resources or filtering information. A smart agent (personal assistant) is a piece of software, which can work autonomously on behalf of its user and can make decisions for him. However the resources are not always readily available to fulfil the needs of the user. Resources usually don't come for free; a price has to be paid to use a resource. By "Resource" we mean anything that can be bought and sold, for example knowledge, processing time or any other commodity, which has a supply and a demand. When goals cannot be achieved independently and there is a need to acquire relevant resources from outside, agents have to reach out in the society and to interact with other agents. In any kind of social interaction where both the parties try to achieve their objectives a conflict situation may arise. Negotiation is a process in which agents involve in order to deal with conflict situations to the best interests of the parties involved.

Most of the research on negotiation in multi-agent systems has been approached using game theory. But to apply negotiation in practical applications where the agents negotiate on behalf of their users, negotiation cannot be treated as a game. In a game, one party wins and the other looses. We believe that for a successful negotiation in a real world situation both the parties must gain some benefits. Negotiating on behalf of the user is not an easy issue. To negotiate successfully an agent must have some knowledge about the opponent. Research on negotiation in various areas of studies (for instance in economic theory, game theory, decision theory and Distributed AI) has shown the importance of knowledge about the opponent. Communication, knowledge acquisition and machine learning approaches have been used for gaining insight into the behaviour of the opponent. The objective of this paper is to incorporate learning among the agents to achieve negotiation in a real life practical environment. In this paper we propose incorporation of learning in an existing negotiation model, based on a decision theoretic approach, described in [Mudgal and Vassileva 1999].

Our domain of application is the I-Help system [Vassileva et al., to appear], a multi-agent based system supporting peer-help among users. In I-Help the helpee's agent negotiate in terms of time and money with the helpers' agent to "buy" a help session from the helper on a given topic. Bargaining for time and money usually leads to conflict situations because different users have different preferences. To be able to resolve the conflict situation a negotiation model based on decision theory was proposed for the peer help environment. To be able to predict opponents' actions is a highly desirable feature of any negotiation system. We incorporate learning into this model that will allow the agents to predict the opponents' actions during the negotiation.

2. Learning Approaches in Negotiation

To know about one's opponent is a very crucial and essential step for negotiation. When negotiation is done by the personal assistants learning about the opponent becomes even more complex because different people have different preferences and personalities. Various learning techniques from the area of machine learning and statistical approaches to learning have been applied to build smart personal assistants that have the ability to learn and adapt according to the environment, however few have focused on agents learning about other agents.

Learning about the opponent in negotiation has been achieved by building a statistical model of the opponent to predict his choices. Such a method has been used to learn opponents' preferences to achieve coordinated choices [Bui et al. 1997]. The learning module gives the agents the ability to know their opponent's preferences via past interactions. Over time agents update their models about the other agents to make coordinated decisions. This mechanism of learning based negotiation has been applied to a meeting-scheduling domain.

Case based reasoning (CBR) is another learning technique used to achieve learning in negotiation. CBR has been used to resolve conflict situation in domains where the conflict is between two organizations or between two individuals for example labor management versus union [Sycara 1987]. This learning technique was deployed in a mediator-based negotiation that consults CBR to generate its plans. CBR reasoning, however, is not beneficial in multi-agent systems, which consist of self-interested agents especially in an economic market based systems because it will involve replication of case examples in all the agents which will lead to unnecessary overhead. Learning by getting feedback from the environment has been found to be useful to model the environment and the other agents. Reinforcement learning (RL) has been used for predicting opponents' reactions. RL has been employed in a market-based environment to study the impact of agent models in an information economy [Vidal et al. 1995]. In RL agents learn by trial and error.

In a highly dynamic environment when the users' preferences are changing it is difficult and not particularly beneficial to model the opponent. Instead of modeling the opponent, stochastic modeling technique has been used for modeling the auction environment and hence predicting the strategies of other agents. For example, [Park et al.1998] use a Markov model, to model their auction environment for the University of Michigan Digital Library.

Issues relating to multi-agent learning have been addressed by modeling the beliefs of the participants and the environment using Bayesian belief networks. BAZAAR, a market based system uses a Bayes' net to model the opponent and Bayes' rule is used to infer about the opponent's decision making and updating the beliefs [Zheng and Sycara 1997].

Predicting other agents' decisions improves the agents' own utility and Chebychev polynomials have been used for modeling the decision making of the opponent. Chebychev polynomials are used for modeling the decision function of the opponent given a collection of decisions made by that agent for load balancing [Sen et al. 1999].

We propose a model-based learning technique in our negotiation model. The agents in our system make use of negotiation history that they build for every negotiation session. Negotiation history and the information from the current negotiation are used to model the opponent. The model of the opponent is built using an influence diagram. The influence diagram model of the opponent is used for assessing the probability of accepting, rejecting or counterproposing an offer.

3. Negotiation in Peer-Help domain

3.1 Domain Description

In an educational environment or at workplace where there is a need to consult peers for help are on-line. To promote peer help in a learning environment, a goal-based multi-agent system is proposed [Vassileva et al., to appear]. I-Help is a multi-agent-based environment in which personal agents represent the users/students. It is believed that by helping not only the recipient's knowledge increases but also the helper benefits in his understanding of the subject. The agents in our system are like personal assistants that take care for finding help for the user at a reasonable price. The personal agents keep representations of their users' resources (knowledge on various topics), goals, relationships and the preference models of their users. In the same time, the I-Help system is a virtual money (similar to the Java-bugs)-based economic society. Help comes at a price, so money has been introduced in our system to motivate the users to help each other. Since different users have different utility values for their money, the agents make decisions on behalf of their user about the price for receiving or giving help and the amount of help-time that the user can get at that price. Hence agents negotiate with each other for better price and time for their users.

The Negotiation model proposed for the system is based on a decision theoretic technique that uses influence diagram to make decisions. Agents in our system are rational in nature and try to maximize their utility. In section 3.2 a brief description of negotiation model is given, section 3.3 describes the model-based learning used for negotiation.

3.2 Negotiation Model

In the peer-help domain the personal agents negotiate on behalf of their users. They do not have complete knowledge about the resources of the compeer. In order to achieve the goals of its user (to get help on a given topic), the helpee agent has to buy resources from the agents of users who are currently working on-line and who have excess of these resources (i.e. are knowledgeable on the topic). To get the list of agents of users, who are currently on-line, the agent contacts a central matchmaker, who has information about the user models of the on-line users. The agents in our domain can act as a helper's agent or as a helpee's agent because a user can play the role of a helpee when he needs help or can act as a helper when his knowledge resources are sufficient to help someone else.

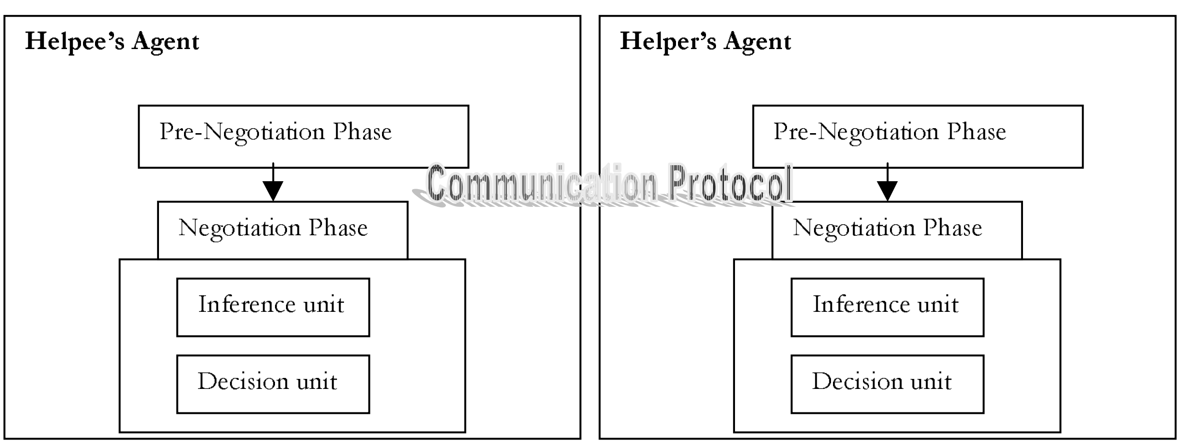


Figure 1 *Negotiation Mechanism in the I-Help System.*

The negotiation mechanism is divided into two phases as shown in figure 1. In the first phase, called the pre-negotiation phase, the two negotiating agents share relevant information about their users' preferences. This pre-negotiation phase is strictly according to the privacy level that the user desires. If the user doesn't want to share his preferences, the opposing agent will use default values for the preferences of the opponent. After the pre-negotiation phase the agents enter into the negotiation phase.

The negotiation phase consists of an inference unit and, a decision-making unit. The inference unit is used for the agent to decide the price that should it should propose or counter propose to the opponent. This unit make inference based on its knowledge about the environment and the knowledge about the opponents' preferences gained from the pre-negotiation phase. The decision –making unit is based on evaluating an influence diagram (ID). The ID allows the agent to make decision that maximises its utility as shown in the figure 2.

Agents learn to assess the probability of the opponents' actions based on the negotiation history. Agents keep a history of all the negotiation interactions they make with every opponent. All the communication between the agents is done using KQML-like speech acts.

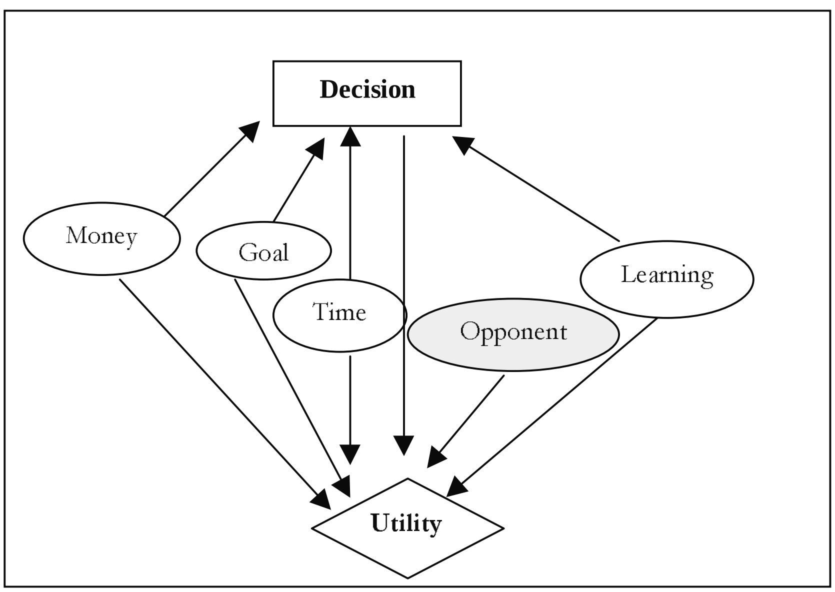


Figure 2*. Infliuence Diagram showing the Helpee's Decision Model.*

A detailed explanation of the negotiation mechanism using influence diagram is presented in [Mudgal and Vassileva 1999].

3.3 Negotiation Mechanism for I-Help

The negotiation protocol is an iterative sequence of agents making offers and counter offers using their decision units. A negotiation strategy is the action that an agent decides to perform at any particular situation i.e. to *accept*, *reject* or *counter propose* the proposal.

When the user needs help, she has to inform her personal assistant about the topic on which help is needed. The agent consults a matchmaker who returns a list of on-line users ranked according to their knowledge level on the topic in which help is required and the standard price for help. The matchmaker produces the list of potential helpers by consulting the user models of all users that are currently on line which store information about the user knowledge on various domain topics. More detail about the user modeling in I-Help is given in [Greer et al, 1998]. The personal agent consults his negotiation history to select a helper. If negotiation history does not exist then the agent selects the first helper from the list. The negotiation history helps the agent to reject a helper that has not been useful in the past, or to contact a helper with whom the user had positive experience (based on he user's feedback). The selection of a helper-agent to be contacted depends also on the preference model of the user that tells her risk tolerance. The basic steps of negotiation are:

1. Agents from both sides enter into a pre-negotiation phase.

2a) Based on the preference model, user model and the information about the environment, the agents determine different possibilities for price that they can offer or ask for. In our system the agents use a decay or the rising function (as a helper or a helpee respectively) and hence calculate the increment or decrement for the price. This is unlike the *Kasbah* market system where the user sets the rising or the decay function to change the price as desired [Chavez and Maes 1996]. The price calculated is fed into the decision unit that chooses to propose the action that maximizes the agent's utility.

2b) The agents create a model of the opponent using an influence diagram and try to predict the opponent's reactions to each action. A detailed explanation of the probability assessment of the opponents' reactions is given in the next section. The parameters that are communicated during the pre-negotiation phase, for example the importance of time, which depends on the time available to the helper and the time requested by the helpee, are deterministic. As shown in figure 1, the agent chooses to implement the action that maximizes the overall utility. Utility calculations are based on multiattribute utility theory.

3. Negotiation is over if the opponent accepts or rejects the offer. If the opponent counter proposes then the new information of the counter proposal is fed back into step 2b.

Since the agents can enter a deadlock of counter proposing each other, we impose a penalty for counter proposing more than three times.

4. Model Based Learning Approach

One can often predict the behaviour of the opponent without going into deep understanding of his motives. In different spheres of negotiation there might be different objectives, still they revolve around meeting the needs of the participants involved. It is essential to look into the past interactions of the opponent that can provide understanding of the opponent's actions by considering and analyzing the proposals he rejected, accepted and counter-proposed. We propose to use a probabilistic influence diagram to model the opponent and to learn to predict his actions using Bayesian learning. Since in our system the model of the opponent is built using an influence diagram, a probability assessment of the actions can be made. The negotiation protocol in our system is based on sequential decision making. In sequential decision making the decision-maker has a chance to update his knowledge after receiving feedback.

Agents in our system acquire prior knowledge about the opponent by looking at their negotiation history with this opponent. The agent keeps a model of the other agent as a helper and - or as a helpee and uses the appropriate model for the opponent depending upon the role it plays at that time. Shown in figure 1 is the decision model of the helpee -- it has a chance node coloured in yellow, which is conditional and gives the probability of the opponents' reactions. The probability assessment of this node is done by modelling the opponent using an influence diagram. In this way, the decision model of the agent in the role of a helpee takes into account the model of the opponent agent as a helper. The agents draw inference from the decision model of the opponent, because the actions taken by the opponent also affect the agents' own utility.

In Figure3 we show the decision model of the helper that generates the probability in the yellow chance node in the decision diagram of the helpee. It is represented as a probabilistic influence diagram. A probabilistic influence diagram [Barlow 1998] is an acyclic directed graph in which

(1) A circle represents random nodes and a double circle represents deterministic nodes.
(2) Directed arcs indicate dependence, and
(3) Attached to each node is a conditional probability function (if the node is a chance node) and deterministic function (if the node is deterministic) that depends on the state of adjacent predecessor nodes.

The chance nodes in the helper agent model are the attributes that affect the decision the helper and are probabilistic in nature; for example how much importance is given to friendship by the helper and the worth of the resources of the helper. The deterministic nodes are those, which are known before the decision is made for example the importance of opponents' goals. The model measures the probability that the helper will accept the offer or counter propose it, provided that it is known how much importance the helper attach to his own goals, the time available. The expected probability that the agent will help is proportional to the importance of the relationship between the users, the importance of the own goal of the helper and the available time. More details about probabilistic influence diagrams can be found in [Shachter 1988].

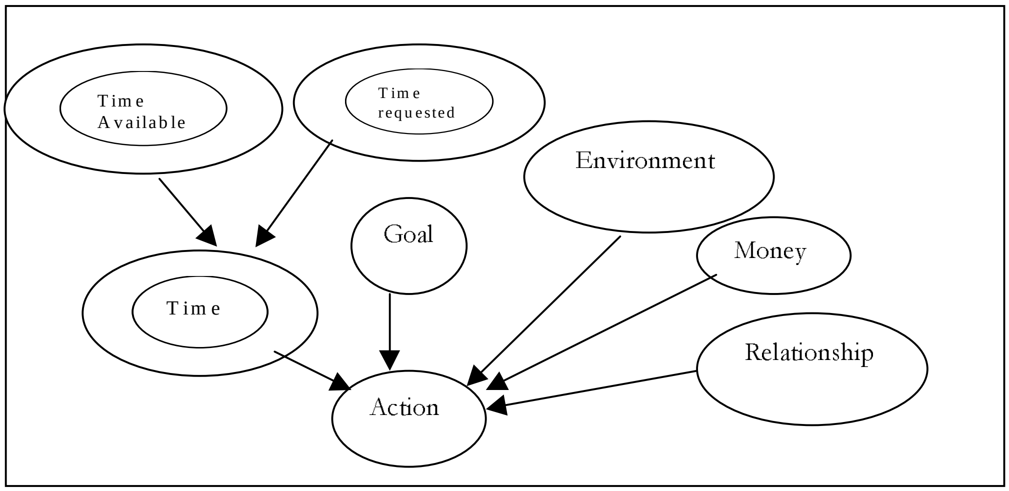


Figure 3. Model of the opponent (the helper) from helpee's viewpoint.

The Bayesian approach for learning is based on probability judgements and follows the laws of probability. For coherence it requires to use probability to measure uncertainty about the random nodes and not to violate the laws of probability while assessing probability. Probabilistic influence diagrams involve three operations

(1) Splitting nodes: A node in influence diagram can represent vector random quantity. It is possible to split it a node into other nodes corresponding to the elements of the vector. For example, if x and y are the two random quantities represented as a vector, then their joint probability function will be P(x,y). From the product of probability we have

$$P(x,y) = P(x)\,\mathbf{P(y|x) = P(x|y)}P(y) \ldots\ldots (1)$$

(2) Merging nodes: If there is only one directed arc from one node x to another node y, then it can be shown from (1) that the two nodes can be merged into one node without changing the joint probability distribution of the nodes.

(3) Arc reversal: This is the operation corresponding to the Bayes' formula which is

$$P(x|y) = P(x)*P(y|x) \,/\, \sum_x P(x)*P(y|x) \ldots\ldots (2)$$

Arcs between two nodes [x, y] can be reversed without changing the joint probability function of the network (if there is no other directed path between x and y). The adjacent predecessors of x become the adjacent predecessors of y in the modified network and vice versa and if the conditional probability functions attached to nodes x and y are also modified according to the laws of probability.

The manipulations of the probabilistic influence diagram are based on eliminating the barren nodes, revising the arcs showing conditional dependence and updating the distributions within the nodes without changing the underlying probability distribution. The process for solving the inference problem involves iterative modifications of the influence diagram.

Using the operations of the probabilistic influence diagram it is possible to predict the actions of the opponent. There is a need for learning in assessing the probabilities of the conditional nodes. In our system the information that the agents can get about each other is deterministic and information such as the importance of money or of the relationship for the opponent is not known, so it will be conditional. Thus the agents can get the information about the opponent that corresponds to the deterministic nodes.

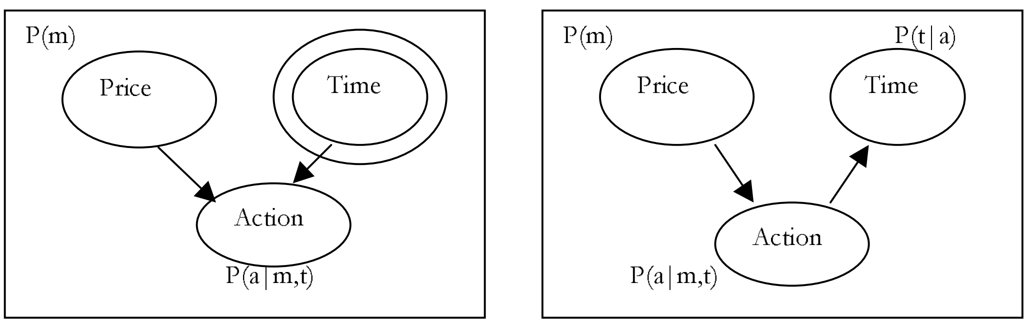
We apply Bayesian learning to update the probability assessment of the random conditional nodes. From the helpee's viewpoint, the helper's actions are dependent on the following factors (see Figure 3): the time available to the helper and the time requested for help, the importance of the helper's current goal, and the utility of money for the helper. The helpee's belief about each attribute of the helper's utility can be represented by a set of hypothesis H_i , I =1,2,3…. Using the history of interactions with the helper (or helpee), a probabilistic assessment over the hypothesis can be made. The Bayesian update rule is used whenever there is a new information (info) from the environment.

$$P(H_i \mid info) = P(info \mid H_i)*P(H_i)/P(info)\ldots\ldots(3)$$

As the interaction with an opponent progresses, the agent learns to assess the probabilities for the unknown parameters affecting the opponent's decision (e.g. the importance of money, importance of the opponents' goal and importance the opponent attaches to relationship) and update them if the prediction for the opponent's decision (manifested in an observed action) has turned out to be wrong. The first round of interaction assumes that the probabilities for *accept*, *reject* and *counter-propose* -actions is equal. Probabilistic inference and reduction of the influence diagram (the model of the opponent) is done to make the predictions. Probability is assigned to all the conditional nodes i.e. the nodes whose values are not known.

5. Example

The example demonstrates the negotiation process as a sequential decision making from helpee's viewpoint. When the helpee's agent decides about making an offer with respect to time and money to the helper's agent, it has to take into account its opponent's (the helper's agent's) reactions. To get good predictions about his opponents' actions, the helpee's agent models his opponent using an influence diagram. The values of probabilistic nodes (for example money) might not be revealed by the opponent in the pre-negotiation phase. Let helpee's hypotheses about the price that the opponent might accept be H_1 = \$100.0, H_2 = \$110.0, H_3 = \$130.0. Using the history of past interactions with the helper's agent, the helpee's agent makes an assessment over these hypothesis $P(H_1) = 0.2$ and $P(H_2) = 0.3$ and so on. Using these assessments, probabilistic operations over the influence diagram representing the model of the opponent can be made. Considering a simplified form of Figure3 with only three nodes, we apply the operations to illustrate how the calculations are made.



(a) **(b)**

Figure 4 *Model of the opponent using two conditional nodes.*

Let's assume Time to be deterministic and the Money (i.e. the price offered with respect to the total amount of money available to the agent) to be conditional. Let Money be abbreviated as m and Time as t and Action as a. Each conditional node as a finite set of outcomes shown as hypotheses and a conditional probability distribution. Outcomes for the action node can be "accept", "reject" and "counter propose". Figure 4a shows the opponent's action dependency on money and time. In Figure 4b the arc from Action to Time is reversed and the deterministic node becomes a chance node. The conditional probability for the time node is calculated using Bayes formula

$P^{new}(t|a) = P(a)*P^{old}(a|t) / \sum_t P(t)*P^{old}(a|t)$ and

$P^{new}(a|t) = P(t)*P^{old}(t|a) / \sum_a P(a)*P^{old}(t|a)$

Where P^{new} is the probability calculated after the arc and P^{old} is the initial probability.

The time node becomes barren and will be reduced. Similarly conditional probability distribution between money and action can be calculated. Using the probabilistic assessment the probability of the opponent's action being "accept" the offer i.e. $P(a=1) = 0.66$ at the price of $90.00.

When during the interaction the opponent does not perform as predicted according to the conditional probability, then the probability for the hypothesis will need to be updated in the presence of the evidence. Suppose the opponent makes a counter offer when the prior probability for action has indicated acceptance. This means that

$$P(H_1|info) = P(H_1)P(info|H_1)/ (P(H_1)*P(info|H_1) + P(H_2)*P(info|H_2))$$

If the standard price[1] is $100 and buyer (helpee) offers $90, if there is new evidence which may be the arrival of more helpers in the system or a counter proposal made by the helper, probability for the H_1, H_2 will now be updated using the above equation. Hence the probability that the helper will ask for a higher price given there are more helpers in the environment will increase the chances of H_1 compared to H_2.

Thus using probabilistic inference and Bayesian learning can lead to a better estimation about the opponents' actions. Since the new offer that an agent will make will be based on more accurate estimation, it will be more beneficial for the agent in negotiation.

6. Conclusions and Summary

Knowing about the opponent is an essential requirement for a successful negotiation. Communication is the most accurate means of acquiring information, but it is not always possible and desirable that the personal agents communicate all the information about their users. To solve this problem, agents need to be smart enough to learn about the opponent based on the past history of interaction, current on-going negotiation and the current environment parameters. In a dynamic system, however, learning about one's opponent is a complex issue. As mentioned in [Holte et al. 1994], the opponents' preferences might change over time. Therefore, even if two agents interact repeatedly in a system, it is still hard to infer accurately the opponent's preferences. In the I-Help domain it is not necessary that the agents will repeatedly interact with each other, especially if there are constantly many users on-line. Hence it is important to learn quickly about the opponent (from few examples).

---

[1] Matchmaker provides a standard price, Buyers of help will try to pay less and Sellers of help will try to get more than the standard price.

We proposed a method for learning about the opponent using an influence diagram to model the attributes affecting his decision and using Bayesian learning to do the prediction for his actions. This method is inexpensive with respect to the number of examples required for learning, and easily accommodates a variable number of factors and preferences (both deterministic and uncertain).

The current I-Help System is under construction. The question that we want to answer with the incorporation of a learning module is to see if it provides measurable benefits in negotiation in a large scale multi-agent environment (with approximately 200-300 agents). To evaluate the proposed method experiments will be made with agents using the model of the opponent for learning versus the agents who use crude probability measure of the opponent’s actions based on inference mechanism.

Acknowledgement:

This work has been partially supported by the Canadian NSERC funded Telelearning Network of Centers of Excellence Project 6.2.4 "Knowledge Management and Collaboration Support in CADRETel".

References

R.E.Barlow. Engineering Reliability. Society of Industrial and Applied Mathematics. 1998.

H.H.Bui, D.Kieronska, S.Venkatesh. Learning Other Agents’ Preferences in Multiagent Negotiation.Thirteenth National Conference on Artificial Intelligence. AAAI’96. Pages 179-184.

A. Chavez and P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods ", Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK, April 1996.

Greer, J., McCalla, G., Cook, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. (1998) The Intelligent HelpDesk: Supporting Peer Help in a University Course, Proceedings ITS'98, San Antonio, Texas.

M.N.Huhns and G.Weiss (Ed.) Machine Learning, 33, 1, Kulwer Publications. 1998.

R.C.Holte, C, and Drummond: A Learning Apprentice for Browsing. Proceedings of the Machine Learning Workshop, Banff, Alberta, Canada, 17 May 1994, Research Report. University of Calgary, Department of Computer Science. x1-x6.

G.I.Nirenberg. Fundamentals of Negotiating. Hawthorn Books. 1975

P.Maes. Agents that Reduce Work and Information Overload. J.Bradshaw (ed.). Software Agents. AAAI Press, 1997.

C.Mudgal and J.Vassileva. Proceeding of the workshop on Agent Based Decision-Support for Managing the Internet Enabled Supply Chain. Third international conference on Autonomous Agents, Seattle, Washington, 1999.

S.Park, E.H.Durfee, W.P.Birmingham. Emergent Properties of a Market based Digital Library with Strategic Agents. Third International Conference on Multiagent Systems.ICMAS'98. Pages 230-237.

S.Sen, A.Biswas, S.Ghosh. Adaptive Choice of Information Sources, Intelligent Information Agents, Matthias Klusch (Ed.), Springer-Verlag, 1999. Pages 253-277.

R.D.Shachter. Probabilistic Inference and Influence Diagrams. Operations Research Vol. 36, No.4, 1988. Pages 589-603.

K.Sycara. Multiagent Compromise via Negotiation. Distributed Artificial Intelligence, Vol2 by L.Gasser and M.H.Huhns, Morgan Kauffman Publishers, 1987.

J. Vassileva, J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant (to appear) A Multi-Agent Approach to the Design of Peer-Help Environments, to appear in Proceedings of AIED'99, LeMans, France, July, 1999.

J.M.Vidal, E.H.Durfee. Recursive Agent Modelling using Limited Rationality. ICMAS 1995. First International Conference on Multiagent Systems. Pages 376-383.

D.Zheng and K.P. Sycara. Benefits of learning in Negotiation. Proceedings, Fifteenth National Conference on Artificial Intelligence. 1997, Pages 36-41.