

Inferring implicit preferences from negotiation actions

Angelo Restificar*
University of Wisconsin-Milwaukee

Peter Haddawy
Asian Institute of Technology

Abstract

In this paper we propose to model a negotiator's decision-making behavior, expressed as preferences between an offer/counter-offer gamble and a certain offer, by learning from implicit choices that can be inferred from observed negotiation actions. The agent's actions in a negotiation sequence provide information about his preferences and risk-taking behavior. We show how offers and counter-offers in negotiation can be transformed into gamble questions providing a basis for inferring implicit preferences. Finally, we present the results of experiments and evaluation we have undertaken.

1 Introduction

As agent technology is applied to problems in e-commerce, we are beginning to see software agents that are able to represent and make decisions on behalf of their users. For example, in Kasbah [6] agents buy and sell on behalf of their human users. In delegating negotiation to an agent, we would like that agent to be as savvy a negotiator as possible. In negotiation, a more effective formulation of proposals can be achieved if an agent has some knowledge about the other negotiating party's behavior, in terms of preferences and risk attitude, since solutions to bargaining problems are affected by both [8, 9, 11]. But such information is private to each party and we cannot expect that the other negotiating party will provide it freely. In addition, the elicitation of preferences and attitudes towards risk, or utility, is in general difficult and continues to be a challenge to researchers. Although various elicitation techniques for decision makers have been widely used, *e.g.*, see techniques in [5], they are not readily applicable in the negotiation scenario. An agent, for instance, cannot simply ask gamble questions to assess its opponent's utility function. On the other hand, the use of learning mechanisms in negotiation has been investigated in several recent studies, see for example [1, 17], and has been shown to be an effective tool in handling uncertainty and incompleteness.

In this paper, we propose a novel method for constructing a model of a negotiator's decision-making behavior by learning from implicit preferences that can be inferred from only observed negotiation actions. We show, in particular, how actions in a negotiation transaction can be transformed into offer/counter-offer gamble questions which can subsequently be used to construct a model of the agent's decision-making behavior via machine learning techniques. The model contains information about the decision maker's preferences between a certain offer and an offer/counter-offer gamble which can be expressed in the form of comparative preference statements. We present theoretical results that allow us to exploit data points implicit in observed actions. We use multiple layer feed-forward artificial neural networks to implement and evaluate our approach. Finally, we present the results of our experiments.

2 Preliminaries

Decision-making based on the principle of maximum expected utility requires the negotiator to choose from a set of possible actions that action which maximizes his expected utility. An action with uncertain outcomes can

*Many thanks to Matt McGinty, Department of Economics, University of Wisconsin-Milwaukee, whose valuable comments and discussions with the first author have benefited this work.

be viewed as a gamble whose prizes are the outcomes themselves. In negotiations where agents exchange offers and counter-offers according to an alternating offers protocol [12], a counter-offer to a given offer can be viewed as a gamble with two possible outcomes: either no agreement is reached, or a counter-offer is accepted in which case an agreement is reached. The probability in this particular gamble is the decision maker's own subjective probability of reaching an agreement with the other negotiating party. A utility maximizing negotiator can decide between a certain offer and the corresponding gamble by comparing the utility of the certain offer and the expected utility of the particular gamble. Thus, the decision-making behavior of a negotiating agent can be modeled by comparative statements that express his preferences between certain offers and particular gambles.

In this paper, we will assume that each party's utility is completely *unknown* to the other and no trusted third party exists. Each agent has a limited reasoning capacity so that endless outguessing regress does not occur. We consider a 2-level agent whose nested modeling capability consists only of a model of its own utilities and a model of the other agent's decision-making behavior based on observations of past actions [16]. As a specific example that we will use to illustrate our approach, we consider a scenario of two agents negotiating over the price of an indivisible item: one agent observing the behavior of a second agent. The first agent constructs a model of the second agent's decision-making behavior by learning from preferences inferred from the latter's observed actions. This scenario is similar to an online seller (first agent) observing a potential buyer's (second agent) actions for the purpose of making effective proposals.

We also assume that the item can be delivered as promised to the buyer and that the seller is paid for it at the agreed price. In each negotiation there is negligible bargaining cost and discount rate. Both bargaining cost and discount rate are considered private information. We assume that the two agents' bargaining behavior does not deviate from their true utilities and that the utility functions of the agents are stationary over the period in which the negotiation takes place. Due to space limitation, we will focus only on constructing a model of an agent's decision-making behavior from observed negotiation actions. We will tackle the relaxation of some of the assumptions above in subsequent work. The claim we make in this paper is that we have a method for constructing a model of an agent's decision-making behavior expressed as comparative preference statements between certain offers and corresponding gambles. The gamble is associated with the agent's counter-offer given his opponent's offer. We will call this the *offer/counter-offer gamble*, or *o/c gamble*.

Note that the set of comparative preference statements that can be obtained from our model is, in general, a proper subset of the set of comparative preference statements that can be obtained from a utility function. Thus, we can use some of the tools for eliciting utilities to construct a model of the agent's decision-making behavior. An important tool in eliciting utilities is the use of lottery or gamble questions and the concept of certainty equivalence.

Definition 1 *Let D be a domain, U be a utility function over D , and let o_i and o_j be outcomes in a gamble G where o_i occurs with a probability p , o_j occurs with a probability $(1 - p)$, and $o_i, o_j \in D$. Let us denote this gamble G by the shorthand $(o_i; p; o_j)$. If $p = 0.5$, then G is called the standard gamble. A certainty equivalent is an amount \hat{o} such that the decision maker (DM) is indifferent between G and \hat{o} . Thus, $U(\hat{o}) = pU(o_i) + (1 - p)U(o_j)$ or $\hat{o} = U^{-1}[pU(o_i) + (1 - p)U(o_j)]$.*

Using the standard gamble, a decision maker can be asked what amount he would assign to \hat{o} such that he would be indifferent between the gamble and \hat{o} . The answer to this standard gamble question is the certainty equivalent \hat{o} . Alternatively, given the outcomes o_i and o_j and an amount \hat{o} , the decision maker can be asked what probability p would he assign such that he would be indifferent between \hat{o} and the gamble $(o_i; p; o_j)$. Utility functions can be constructed by interviewing the decision maker and asking him the answers to gamble questions [5]. A gamble question is nondegenerate if $p \neq 0$ and $p \neq 1$. A decision maker with an increasing utility function is risk-averse (resp. risk-prone, risk-neutral) iff his certainty equivalent for any nondegenerate gamble is less than (resp. is more than, equal to) the expected value of the gamble. For a decreasing utility function, a decision maker is risk-averse (resp. risk-prone, risk-neutral) iff his certainty equivalent for any nondegenerate gamble is more than (resp. is less than, equal to) the expected value of the gamble.

We will model the agents' exchanges as alternating offers [12] but will use Zeuthen's concept of probability to risk a conflict [18] as a basis for transforming negotiation offers and counter-offers into gamble questions. In

the alternating offers game we refer to the history of the agents' actions as a negotiation sequence. To simplify the type of actions, we will simply call the first action the initial *offer* and denote all actions after that as either an *accept* action, a *reject* action, or a *counter-offer* action. To avoid confusion, we will use subscripts to denote which agent the action belongs to. The negotiation terminates in one of the following ways: either an agent has accepted an offer, an agent has rejected the offer, or the negotiation has reached an impasse, *i.e.*, a deadlock. An impasse happens when no agent makes a concession, giving the same counter-offers they have given previously for a consecutive finite number of times. We formally define a negotiation sequence below.

Definition 2 Let \mathbf{o} , \mathbf{a} , \mathbf{r} , and \mathbf{c} denote *offer*, *accept*, *reject*, and *counter-offer*, respectively. An action α is a negotiation action iff $\alpha \in \{\mathbf{o}, \mathbf{a}, \mathbf{r}, \mathbf{c}\}$. Let $D = [P_{min}, P_{max}]$ be the domain of an attribute over which a commodity is to be negotiated. Any $x \in D$ is called a position. Let $\mathcal{A} = \{S, B\}$ where S and B denote the seller and buyer agents, respectively. A negotiation sequence N is the sequence $N = (\alpha_j, x_1), (\alpha_k, x_2), \dots$ where $j, k \in \mathcal{A}$, $j \neq k$, $x_i \in D$, and the last action is in $\{\mathbf{a}, \mathbf{r}, \mathbf{c}\}$. A negotiation transaction is any pair of consecutive actions.

Consider the following example where buyer B negotiates the price of a certain commodity with seller S .

Negotiation Fragment 1

- B_1 . I would like to offer \$45.00 for it.
- S_2 . This is a fine item. I am selling it for \$66.00.
- B_3 . I think \$45.00 is more reasonable.
- S_4 . Alright, \$60.00 for you.
- B_5 . Let's see. I'd be willing to pay \$51.00 for it.
- S_6 . \$60.00 is already a good price.
- B_7 . \$51.00 is fair, I think.
- S_8 . I'll reduce it further to \$57.00, just for you.
- B_9 . I really like the item but I can only afford \$54.00.
- S_{10} . \$57.00 is a bargain.
- B_{11} . \$54.00 is my last offer.
- S_{12} . Alright. Sold at \$54.00.

In Negotiation Fragment 1, $D = [45, 66]$. The negotiation sequence $N = (\mathbf{o}_B, 45), (\mathbf{c}_S, 66), (\mathbf{c}_B, 45), (\mathbf{c}_S, 60), (\mathbf{c}_B, 51), (\mathbf{c}_S, 60), (\mathbf{c}_B, 51), (\mathbf{c}_S, 57), (\mathbf{c}_B, 54), (\mathbf{c}_S, 57), (\mathbf{c}_B, 54), (\mathbf{a}_S, 54)$. The buyer B makes an offer in B_1 by offering \$45.00. In S_2 , the seller counter-offers by quoting a price of \$66.00 for the item. This continues until S_{12} where the seller accepts the buyer's counter-offer. The terminating move is $(\mathbf{a}_S, 54)$. We will now show how counter-offers can be viewed as gamble questions that provide implicit data points for learning an agent's decision-making behavior.

3 Negotiation actions as gamble questions

The theoretical results in this section point out implicit information that may be exploited from observed counter-offers. First, we define an agent's probability to risk a conflict following Zeuthen.

Definition 3 Let U_B and U_S be B and S 's utility function, respectively. Let x_B be B 's position and x_S be S 's position. The probability that B will risk a conflict, π_B , and the probability that S will risk a conflict, π_S , are defined as follows:

$$\pi_B = \begin{cases} \frac{U_B(x_B) - U_B(x_S)}{U_B(x_B)} & \text{if } U_B(x_B) > U_B(x_S) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\pi_S = \begin{cases} \frac{U_S(x_S) - U_S(x_B)}{U_S(x_S)} & \text{if } U_S(x_S) > U_S(x_B) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The probability that an agent would risk a conflict is proportional to the difference between the agent's position and what he is offered. The closer the other agent's offer is to his position, the smaller this probability should be. The farther away the other agent's offer is from his position, the larger the readiness to risk a fight or conflict. If an agent is offered something equal to his position or is offered something better, then his probability of risking a conflict is zero. We assume the following about the agents' utilities:

A. 1 B 's utility function, U_B , is monotonically decreasing.

A. 2 S 's utility function, U_S , is monotonically increasing.

By computing the expected utility, an agent can decide whether to accept the other agent's offer, reject it, or insist its own position.

Definition 4 Let U_B and U_S be B and S 's utility function, respectively, and α_B and α_S be B and S 's action, respectively. Let the gambles G_B and G_S refer to $(\text{conflict}_B; p_S; x_B)$ and $(\text{conflict}_S; p_B; x_S)$, respectively. B 's (resp. S 's) subjective probability that S (resp. B) will risk a conflict is p_S (resp. p_B). Also, let $EU[G_B] = (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)$, and $EU[G_S] = (1 - p_B)U_S(x_S) + p_B U_S(\text{conflict}_S)$.

$$\alpha_B = \begin{cases} \text{accept} & \text{iff } U_B(x_S) \geq EU[G_B] \wedge \\ & EU[G_B] > U_B(\text{conflict}_B) \\ \text{counter-offer} & \text{iff } U_B(x_S) < EU[G_B] \wedge \\ & U_B(x_S) > U_B(\text{conflict}_B) \\ \text{reject} & \text{iff } U_B(\text{conflict}_B) \geq U_B(x_S) \end{cases} \quad (3)$$

$$\alpha_S = \begin{cases} \text{accept} & \text{iff } U_S(x_B) \geq EU[G_S] \wedge \\ & EU[G_S] > U_S(\text{conflict}_S) \\ \text{counter-offer} & \text{iff } U_S(x_B) < EU[G_S] \wedge \\ & U_S(x_B) > U_S(\text{conflict}_S) \\ \text{reject} & \text{iff } U_S(\text{conflict}_S) \geq U_S(x_B) \end{cases} \quad (4)$$

Note that $(1 - p_S)$ is B 's subjective probability that he himself will succeed, *i.e.*, the probability that S will not risk a conflict. The knowledge of either p or $(1 - p)$ as a subjective estimate guides an agent's decision. In the case where an agent's probability to risk a conflict can be estimated via empirical frequency, the method we propose here still applies. The latter, however, usually requires a large sample size which may not be easily available in practice.

Definition 4 specifies what relation is true when a specific action is taken. In particular, counter-offers which comprise most actions in a negotiation sequence imply preference between an o/c gamble and an offer. We do not know what specific values p_B and p_S have, but we can learn the relation that triggered the negotiation action. B 's estimate, for example, of S 's probability to risk a conflict p_S is incorporated in the relation on which B 's actions are based. According to Savage [14], the use of a gamble question as a probability elicitation function when B is maximizing the effect of an outcome would induce B to reveal his opinion as expressed by his subjective probability. This clearly means that for any counter-offer by B some specific value of p_S was used by the decision maker B although this value can not be directly observed. Thus, the p_S in the gamble G_B of the preference relation $G_B \succ x_S$ has a specific value. The same can be said about B 's subjective probability of success, $(1 - p_S)$, when insisting x_B . Although the exact value of p_S or $(1 - p_S)$ is not known, by learning the relation that triggered B 's action along with the parameters x_S , x_B , and conflict_B , instances are learned where the values of p_S and $(1 - p_S)$ are taken into account.

Learning relations also offers additional flexibility. If the value of conflict is known then we can use it in the training instances. Note that conflict_B and conflict_S is the outcome for the buyer and the seller, respectively, whenever no agreement is reached. In a buyer-seller scenario the value of conflict for the buyer might be the prevailing market price. conflict_B could be the highest possible value a buyer could end up paying should a non-agreement occur or for the seller, conflict_S could be a low constant if the seller ends up having to sell the item resulting in a deep discount or a zero profit. In the case where the price of conflict is not known,

relations can still be learned using some default constant $c_B \in D$ provided $U_B(c_B) < U_B(x_S)$ in B 's case and $U_S(c_S) < U_S(x_B)$ in S 's case. As long as the same constant c_B (resp., c_S) used for training is also the one used in the query, the learned model is expected to output the corresponding preference information based on the relations it has learned. In essence, a model of the agent's decision-making behavior can be constructed without necessarily knowing the exact value of conflict.

The next three theorems allow us to generate training sets implicit from observed actions. Due to space limitation, we will only present theoretical results for the buyer. Analogous results also hold for the seller.

Theorem 1 *Let x_B be B 's counter-offer, x_S be S 's offer, and let \hat{x}_B be B 's certainty equivalent for gamble $G_B = (\text{conflict}_B; p_S; x_B)$. If $(G_B \succ x_S)$ then for any nondegenerate G_B , $\hat{x}_B \in (x_B, x_S)$.*

Proof. $G_B \succ x_S$ means $(1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B) > U_B(x_S)$. Since $p_S \neq 0$, it follows that $U_B(x_B) > (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)$ and so by transitivity, we have $U_B(x_S) < (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B) < U_B(x_B)$. Note that $U_B^{-1}[(1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)] = \hat{x}_B$. By monotonicity assumption A.1, $\hat{x}_B > x_B$ and $\hat{x}_B < x_S$. Hence, $\hat{x}_B \in (x_B, x_S)$.

Theorem 2 (Inferior Offers) *Let x_B be B 's counter-offer, x_S be S 's offer, and $G_B = (\text{conflict}_B; p_S; x_B)$. Let p_S^* be S 's readiness to risk a conflict at x , $x_S < x$. If $(G_B \succ x_S)$ then $((1 - p_S^*)U_B(x_B) + p_S^* U_B(\text{conflict}_B)) > U_B(x)$, $\forall x$ such that $x_S < x$.*

Proof. Since $U_B(x_S) > U_B(x)$ for any $x > x_S$, then $((1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)) > U_B(x)$. S 's offer is at x_S , so any $x > x_S$ must be better for S than x_S by A.2. Since x is better than S 's position, by definition, S 's readiness to risk a conflict, p_S^* , can not increase. By transitivity, $(1 - p_S^*)U_B(x_B) + p_S^* U_B(\text{conflict}_B) > U_B(x)$.

Theorem 3 (Irresistible Offers) *Let x_B be B 's counter-offer, x_S be S 's offer, and $G_B = (\text{conflict}_B; p_S; x_B)$. Let p_S^* be S 's readiness to risk a conflict at x , $x_B > x$. If $(G_B \succ x_S)$ then $((1 - p_S^*)U_B(x_B) + p_S^* U_B(\text{conflict}_B)) < U_B(x)$, $\forall x$ such that $x_B > x$.*

Proof. By A.1, $U_B(x) > U_B(x_B)$. $p_S \neq 0$ because B 's position, x_B , is not the same as S 's offer, x_S . Since $0 < p_S \leq 1$, $U_B(x_B) > (1 - p_S)U_B(x_B)$. By transitivity, $U_B(x) > (1 - p_S)U_B(x_B)$. However, $U_B(x_B) > U_B(\text{conflict}_B)$. Thus, $U_B(x) > (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)$. Since x is less than S 's position, x_S , S 's readiness to risk a conflict increases, by definition. So $p_S^* > p_S$. Since $U_B(x) > (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict}_B)$, by transitivity $U_B(x) > (1 - p_S^*)U_B(x_B) + p_S^* U_B(\text{conflict}_B)$.

Theorem 1 states that a counter-offer reveals the interval in which the certainty equivalent lies. According to Theorem 1, if B decides to make a counter-offer x_B to an offer x_S by S , we can infer that B 's certainty equivalent for gamble G_B lies in the interval (x_B, x_S) . This is useful because, intuitively, for B any offer by S above x_S can be considered worse than G_B and any offer by S below x_B is preferred to G_B . Theorem 2 states that if B prefers the gamble to an offer of x_S by S then B would also prefer the gamble to any offer by S that is greater than x_S . Theorem 3 says that an offer by S that is less than what B would insist is acceptable to B .

Example 1 *Consider the transaction (S_{10}, B_{11}) in Negotiation Fragment 1. According to Theorem 1, the behavior observed in (S_{10}, B_{11}) implies that \hat{x}_B lies in the interval $(\$54.00, \$57.00)$. According to Theorem 2, for all offers $x > \$57.00$, B would prefer the gamble insisting $\$54.00$ to x . By Theorem 3, for all offers $x < \$54.00$, B would prefer the offer to a gamble insisting $\$54.00$.*

Each distinct pair of exchanges in a negotiation sequence gives different information with respect to the risk attitude that each agent shows throughout the negotiation sequence. In particular, each p_B and p_S may be different between transactions, hence each pair gives additional information. If a transaction or a pair of counter-offers are observed repeatedly then the training instances based on this information are also increased, in effect, adding more evidence to the decision maker's specific risk behavior. In summary, Theorems 1-3 state information that is implicit in each counter-offer. Using these, we can generate explicit data points to be used

as training sets for artificial neural networks. In previous work (see [10, 4]), artificial neural networks are used to construct a decision maker’s utility model. The network can be trained to learn from the instances generated from observed data. In this work, the weights in the network encode the function that describes the decision-making behavior of the agent being modeled. The network takes as input a certain offer, counter-offer, and the price of conflict, and outputs a prediction of whether or not that o/c gamble is preferred by the decision maker.

We envisage the models discussed here to be used as tools to formulate effective offers which include tradeoff exploration, and as heuristic to help guide the search for efficient solutions in negotiation problems. Often, models predict actions of negotiating agents using expectations of what a decision maker might do given relevant information. In game theory literature, these specific sets of relevant information that affect the decision maker’s decision processes are also known as *types* [3]. They may include, for example, reservation price, cost and profit structure, negotiation style, amount of resources, etc. [17]. Types are often modeled using a probability distribution. As the negotiation progresses, updates using Bayes’ rule on the probability distribution over these specific sets of relevant information are made. A difficulty in the use of types is how prior probability distribution might be updated over a very large set of information, possibly one that contains infinite number of types. As in most cases, the modeler is forced to assume a finite number of types say, strong and weak types [13] or is forced to choose a smaller set of ‘relevant’ information say, reservation price as opposed to everything that affects a decision maker’s action, as exemplified in Zeng and Sycara’s work [17].

A disadvantage in the use of probability distribution is not necessarily in determining which information is relevant but in computing posteriors given the limited information that can be inferred from observed negotiation actions. For example, if the only observations acquired during negotiations are the responses of the negotiating agents expressed as counter-offers then clearly, one could not use the observations to directly infer certain specific sets of relevant information such as the cost and profit structure. As we have shown above, we can infer the agent’s decision-making behavior in terms of preferences between a certain offer and an o/c gamble from observed actions. All that is needed is the assumption that a rational player evaluates his decision by taking into consideration *all* the relevant information he has which affect the issue(s) under negotiation. This might include indirectly observable information like his cost and profit structure, deadline, etc. If, given this relevant information, an offer is more favorable to him than say a gamble then he takes it because he prefers its outcome over that gamble. Hence, whatever observations are made can be taken as a result of such evaluation that is reflected through the decision maker’s behavior. For example, one need not explicitly model the cost and profit structure of the opponent. If the opponent accepts an offer then this can be viewed as something that he prefers, after considering his cost and profit structure, over what other consequences might result. In short, our approach allows the construction of one *particular* model that represents the decision maker’s behavior, as opposed to choosing (via a probability distribution) the opponent’s type from a very large, possibly infinite, set of information.

4 Model construction and evaluation

In the previous section, we have laid out the framework for constructing a model of a negotiating agent’s decision-making behavior using implicit data from observed negotiation transactions. We will describe in this section the construction and evaluation of a model of *B*’s decision-making behavior. The results from our experiments using synthetic data sets suggest significant predictive accuracy using only an average of 5 pairs of exchanges per negotiation. For convenience, we have chosen to implement a model of *B*’s decision-making behavior using a multilayer feed-forward artificial neural network, a statistical learning method capable of approximating linear and nonlinear functions. Note that the concept we discussed above on how implicit data points maybe generated from observed negotiation actions is independent of the machine learning technique that maybe used. We believe, however, that the choice of a particular technique must be made so that the intended preference relations are learned given the available additional training data. Since artificial neural networks have been widely studied and used as universal function approximators in various problems across many disciplines, we believe that a demonstration of our model via artificial neural networks would make our technique more accessible to a wider audience. In addition, extensions and variants of artificial neural networks

like the knowledge-based artificial neural network [15] can potentially be used to improve the model’s predictive performance as well as reduce its training time [10, 4].

We have assumed that B ’s decision-making behavior is guided by his utility function. We used a control utility function for B , $\gamma(x)$, to generate negotiation actions where each counter-offer x made by B is such that $\gamma^{-1}((1 - p_S)\gamma(x) + p_S\gamma(c)) < y$, for any offer y made by S and for a given conflict c . This means that a counter-offer is made because B ’s expected utility of the gamble is higher than his utility for y . B accepts S ’s offer when the utility of y is equal to or exceeds that of the gamble. Moreover, we have arbitrarily chosen two separate functions for $\gamma(x)$ representing risk-averse and risk-prone behavior and have run several experiments varying the following parameters to construct a synthetic data set: probability to risk a conflict (p_S), negotiation domain, and negotiation strategy. Note that we use p_S together with other parameters only to simulate behavior from which actions can be observed. The construction of the model does not assume knowledge of p_S . The control utility function, $\gamma(x)$, allows us to test the predictive performance of our model. Its purpose is to represent B ’s unknown utility function against which our constructed model will be tested. Our data set contains observable actions based on $\gamma(x)$. However, $\gamma(x)$ itself is unknown to the system that constructs the model.

For any nondegenerate gamble the certainty equivalent partitions the offers into two regions: the region below the certainty equivalent contains offers that B prefers to a gamble and the region above it contains those where a gamble is preferred. The $\gamma(x)$ curve contains the utility of the certainty equivalent x for any nondegenerate gamble. According to the previous section, the counter-offers in a negotiation sequence can be used as interval constraints to approximate the certainty equivalents. We evaluate the effectiveness of our approach by training the artificial neural networks using data points implied by these intervals. We then test the network’s predictive performance by comparing the learned model against the control utility function which guides the decision maker’s behavior. An effective model should demonstrate a significant improvement over a random guess, *i.e.*, given S ’s offer and a conflict value held constant, the model should be able to correctly classify more than 50% of the time whether a given offer is preferred by B over a gamble.

We generated negotiation sequences and ran experiments using two control utility functions: $\gamma_1(x) = 1 - \frac{e^{0.05x}}{149}$ (risk-averse, decreasing function) and $\gamma_2(x) = e^{-0.0025x}$ (risk-prone, decreasing function). In each negotiation sequence, p_S is either generated randomly or is chosen from $\{0.50, 0.25, 0.60\}$. The negotiation using $\gamma_1(x)$ is over the domain $D_1 = [50, 100]$ and that of $\gamma_2(x)$ is over $D_2 = [200, 700]$. The value of conflict c is set at the maximum value of the domain. The buyer-seller negotiation strategy, σ , vary among Boulware-Conceder, Conceder-Boulware, and Conceder-Conceder pairs. We define our Boulware strategy as one where the agent concedes only 10% of the time and a Conceder strategy as one where concession is frequent at 90% of the time. Whenever an agent concedes, concession is randomly chosen between 0 – 50% of the difference between both agents’ most recent counter-offers.

The artificial neural network used in our experiments has one hidden layer with four nodes. The input layer contains three input nodes and the output layer contains two nodes. The output nodes represent $G_B \succ x_S$ and $G_B \prec x_S$ where $G_B = (c_B; p_S; x_B)$. The input to the network are: B ’s price of conflict c_B , B ’s counter-offer x_B , and S ’s offer or counter-offer x_S . Data fed into the input layer are scaled so that values only range between 0 and 1. We point out that the input layer does not include a node for the specific p_S of the gamble G_B . Although p_S is not directly observable, the observed gamble G_B which is used to generate the training instances together with x_S has a specific value of p_S associated with it. Whenever we want to ask the question whether $G_B \succ x_S$ or $G_B \prec x_S$ given certain offer x_S we are referring to a specific G_B with an associated specific value of p_S . Note that the input query in this case would only contain c_B , x_B , and x_S .

Negotiation sequences used for training, tuning, and testing are randomly generated using a chosen strategy pair, a control utility function $\gamma(x)$, a negotiation domain, and a constant conflict value. We used a $k - 1$ cross validation method to train and tune the network, where k is the number of negotiation transactions in each negotiation sequence. Network training is stopped when either no improvement in performance is detected for a successive 2,000 epochs or the number of epochs reaches 20,000. Among the data generated using the intervals, 90% is used for training and 10% is used for tuning. For the examples below the interval, the network is trained to output ($o_1 = 0.1, o_2 = 0.9$) and for the examples above the interval it is trained to

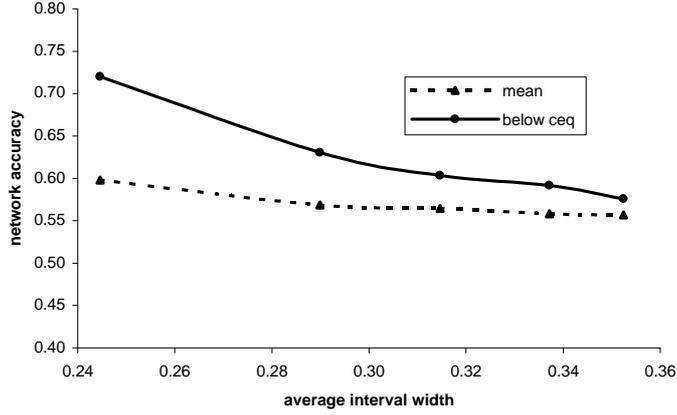


Figure 1: Overall Performance Results

output ($o_1 = 0.9, o_2 = 0.1$). Data points used to test the network are separately generated using the true \hat{x}_B of the gamble in each negotiation transaction. Our data set contains a total of 97 negotiation sequences. The total number of negotiation transactions is 477 which gives an average of 5 transactions per negotiation sequence. The training instances are obtained by generating a total of 200 random data points for each observed negotiation transaction; 100 random data points for each of the region below and above the interval. These are simply data points that are directly inferable from the observed transaction and help to fill out the data presented to the neural network. The certainty equivalent, which is obtained from the control utility function, lies inside each interval. For each of the regions below and above the certainty equivalent 100 test points are generated. We then evaluate the approach by comparing how well the model performs when trained using the intervals against the test points from the control utility function. We consider data points to be correctly classified when $o_1 \in [0, 0.2], o_2 \in [0.8, 1]$ for test points below the \hat{x}_B and when $o_1 \in [0.8, 1], o_2 \in [0, 0.2]$ for test points above the \hat{x}_B . All (pseudo)random numbers in the experiment are generated using the Mersenne Twister algorithm [7].

Intuitively, not all negotiation transactions may be useful. For example, an offer that is near the maximum domain value and a counter-offer that is near the minimum domain value has an interval width that is close to the width of the domain. Since we are using the interval to estimate the certainty equivalent such a negotiation transaction would be less useful than one in which both the offer and counter-offer are closer to the certainty equivalent. Moreover, regression analysis indicates that the distance of the lower interval limit to the true \hat{x}_B has significant influence on our model’s predictive performance. Predictive accuracy is 50% or better when the average distance of the lower interval limit is within 0.50 (normalized) of the true \hat{x}_B . In real scenario, however, we have no idea how far the lower interval limit is to the true \hat{x}_B since this is considered private information. We would, therefore, like to have a practical basis for choosing which negotiation transactions are useful as training examples. We used the interval width of each negotiation transaction, *i.e.*, the distance between the lower and upper limit of the interval, as a basis to eliminate data points that may not be useful in constructing the model.

To test the overall performance, negotiation sequences were grouped into subsets where normalized interval widths are no greater than 0.50, 0.45, 0.40, 0.35, and 0.30. A normalized interval width is computed as the ratio of the interval width to the domain width. The average interval width for each of the subset above is 0.35, 0.34, 0.31, 0.29, and 0.24, respectively. The average network performance of each of these respective subsets is shown in Figure 1. The overall network performance increases as the average interval width corresponding to the negotiation transactions decreases. The solid curve shows the performance of the model in predicting whether a certain offer is preferred by B to an o/c gamble using only implicit data points below the interval. This is important because B ’s counter-offers only correspond to the lower limit of the interval. The dotted curve shows the accuracy of the model in predicting whether a certain offer is preferred by B to an o/c gamble

and whether B prefers an *o/c* gamble to a certain offer. The mean accuracy is obtained by averaging the results using implicit data points below *and* above the interval. The results suggest that for intervals with average width of 0.24 the network can predict about 72% of the time whether a certain offer is preferred by B to an *o/c* gamble. For intervals with average width of less than or equal to 0.31, we are able to predict with more than 60% accuracy whether B prefers a certain offer to an *o/c* gamble. In addition, the predictive accuracy of the model when implicit data points above and below the interval are used is better than a random guess. We ran four right-tailed z -tests and one right-tailed t -test using the following hypotheses: $H_0 : \mu = 0.50$ and $H_a : \mu > 0.50$. For the t -test the null hypothesis is rejected at $\alpha = 0.005$. In each of the z -tests, the null hypothesis is rejected at $\alpha = 0.001$.

5 Related Work and Summary

Zeng and Sycara [17] propose to model sequential decision making by using a subjective probability distribution over an agent's beliefs about his environment and about his opponent, including the opponent's payoff function. A negotiating agent chooses the action that maximizes his expected payoff given the available information. Zeng and Sycara use observed actions to obtain a posterior distribution over an agent's belief set using Bayesian rules. In their framework, a decision maker's action is predicted using the posterior distribution. In our framework, a decision maker's action is predicted using a learned model that contains information about the decision maker's preferences between a certain offer and the *o/c* gamble associated with a counter-offer. As suggested above, our approach aims to avoid the difficulty associated with obtaining posterior distributions from the limited information that can be inferred from observed actions.

Bui et al. [1] propose to augment a negotiation architecture with an incremental learning module, implemented as a Bayesian classifier. Using information about past negotiations as sample data, agents learn and predict other agents' preferences. These predictions reduce the need for communication between agents, and thus coordination can be achieved even without complete information. Here, preferences are modeled using a probability distribution over the set of possible agreements. An agent's preference is estimated using the expected value of the distribution. Our work differs from [1] on several points: first, our model includes both preferences and risk-taking behavior; second, we use a model of an agent's decision-making behavior to arrive at a possible offer instead of using a Bayes classifier; and finally, the negotiation context in which we consider is largely adversarial rather than cooperative in that the agents can insist on their respective positions even if this leads to a breakdown in negotiation, agents can not be asked about their preferences directly, and the only type of information exchanged between agents are offers and counter-offers.

Chajewska et al. [2] propose to elicit utility functions from observed negotiation behavior. An elicited utility function can be used by the decision maker to determine which action gives the maximum utility. The model we have discussed here does not elicit a utility function but rather represents and learns the negotiating agent's decision-making behavior in terms of preferences between a certain offer and an *o/c* gamble. Some other notable differences between our work and that of Chajewska et al. are as follows: (1) The approach proposed by Chajewska et al. starts off from a database of (partially) decomposable utility functions elicited via standard techniques. Standard elicitation techniques, like interviews using gamble questions, is in general inapplicable in most negotiation scenarios. We have demonstrated how counter-offers can be viewed as gamble questions. (2) Observed behavior is used to eliminate inconsistent utilities and from the consistent ones the expectation of the distribution of utilities is chosen. Our approach uses observed behavior to infer and learn implicit preferences. (3) In our framework, we exploit the subjective probabilities which is an inherent hidden component in an agent's move. However, probabilities from empirical frequency, whenever available, can also be used. The only disadvantage in using objective probabilities as is suggested in [2] is that sufficiently large sample size is needed which may not be available. We, however, agree with Chajewska et al. about the use of existing knowledge in the elicitation process. In previous work we have shown that such knowledge can be used to guide an elicitation process [4] and improve the predictive performance of a constructed model [10].

In summary, we have presented a novel method for constructing a model of an agent's decision-making behavior by learning from implicit preferences inferred from observed negotiation actions. In particular, we

have presented theoretical results which allow counter-offers to be transformed into gamble questions, providing the basis for inferring implicit preferences. Training instances can be generated from intervals implicit in counter-offers which can then be used to train artificial neural networks. Our model, which can be augmented by existing knowledge, determines whether a certain offer is preferred to an o/c gamble or vice-versa. In our experiments, we have obtained statistically significant results which indicate over 70% accuracy for intervals whose widths are roughly 25% of the domain width. Moreover, the accuracy of the model when implicit data points above and below the interval are used is better than a random guess.

References

- [1] H. H. Bui, S. Venkatesh, and D. Kieronska. Learning other agents' preferences in multi-agent negotiation using the bayesian classifier. *International Journal of Cooperative Information Systems*, 8:275–294, 1999.
- [2] Urszula Chajewska, Daphne Koller, and Dirk Ormoneit. Learning an agent's utility function by observing behavior. In *Proceedings of the International Conference on Machine Learning*, 2001.
- [3] Drew Fudenberg and Jean Tirole. *Game Theory*. The MIT Press, 1991.
- [4] Peter Haddawy, Vu Ha, Angelo Restificar, Ben Geisler, and John Miyamoto. Preference elicitation via theory refinement. *Journal of Machine Learning Research*, 4:317–337, Jul 2003.
- [5] Ralph Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.
- [6] Patti Maes, Robert H. Guttman, and Alexandros G. Moukas. Agents that buy and sell. In *Communications of the ACM*, volume 42. ACM, March 1999.
- [7] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, 1998.
- [8] John Nash. The bargaining problem. *Econometrica*, 18(2):115–162, 1950.
- [9] Martin J. Osborne. The role of risk aversion in a simple bargaining model. In Alvin E. Roth, editor, *Game-theoretic models of bargaining*. Cambridge University Press, 1985.
- [10] Angelo Restificar, Peter Haddawy, Vu Ha, and John Miyamoto. Eliciting utilities by refining theories of monotonicity and risk. In *Working Notes of the AAAI-02 Workshop on Preferences in AI and CP: Symbolic Approaches*. American Association for Artificial Intelligence, July 2002.
- [11] A. Roth and U. Rothblum. Risk Aversion and Nash's Solution for Bargaining Games with Risky Outcomes. *Econometrica*, 50:639–647, 1982.
- [12] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.
- [13] Ariel Rubinstein. A bargaining model with incomplete information about time preferences. *Econometrica*, 53(5):1151–1172, 1985.
- [14] Leonard J. Savage. Elicitation of Personal Probabilities and Expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- [15] Geoffrey G. Towell and Jude W. Shavlik. Knowledge-Based Artificial Neural Networks. *Artificial Intelligence*, 70, 1995.
- [16] Jose M. Vidal and Edmund H. Durfee. The impact of nested agent models in an information economy. In *Proceedings of the Second International Conference on Multiagent Systems*, 1996.
- [17] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *Int. Journal Human-Computer Studies*, 48:125–141, 1998.
- [18] Frederick Zeuthen. *Problems of Monopoly and Economic Warfare*. Routledge and Kegan Paul, Ltd., 1930.