Modeling Opponent Decision in Repeated One-shot Negotiations

Sabyasachi Saha
Department of Math & Computer Science
The University of Tulsa
saby@utulsa.edu

Anish Biswas
Department of Math & Computer Science
The University of Tulsa
abiswas1@yahoo.com

Sandip Sen
Department of Math & Computer Science
The University of Tulsa
sandip@utulsa.edu

ABSTRACT

In many negotiation and bargaining scenarios, a particular agent may need to interact repeatedly with another agent. Typically, these interactions take place under incomplete information, i.e., an agent does not know exactly which offers may be acceptable to its opponent or what other outside options are available to that other agent. In such situations, an agent can benefit by learning its opponent's decision model based on its past experience. In particular, being able to accurately predict opponent decisions can enable an agent to generate offers to optimize its own utility. In this paper, we present a learning mechanism using Chebychev's polynomials by which an agent can approximately model the decision function used by the other agent based on the decision history of its opponent. We study a repeated one-shot negotiation model which incorporates uncertainty about opponent's valuation and outside options. We evaluate the proposed modeling mechanism for optimizing agent utility when negotiating with different class of opponents.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*

General Terms

Experimentation,Performance

Keywords

Negotiation, Chebychev polynomial, learning

1. INTRODUCTION

Both in human and multi-agent societies, negotiation is the most well known approach to resolving conflicts. It is a widely-studied subject in Economics, Law and Artificial Intelligence. With the massive growth of E-commerce, research in automated negotiation is receiving increasing attention. Different forms of automated negotiation have been studied analytically and experimentally. In this paper, we focus on a one-shot negotiation problem, where two agents are negotiating for an indivisible item. One of these agents, the seller, possesses the item and the other, the buyer, wants to buy it. In this model, the buyer proposes a price and the seller either accepts or rejects it. In real-life negotiation, there may also be other buyers wanting to buy this item from the seller. This is known as outside options in negotiation [11, 12]. Outside option is the price available to the seller when the buyer is making the offer. As Li et. al. [11] have shown, outside options influence the utility of the agent (here seller) via its reservation price. A rational seller will agree to the buyer's proposal if the proposal is better than the outside options. Here price is the only issue being negotiated. Sometimes seller's risk attitude may also play a role in the negotiation outcome. We assume that a seller chooses a decision function based on the risk attitude and expected outside option available from the market. A decision function gives the probability of acceptance of an offer given a price. The function is typically monotonic in the sense that the higher the price offered by the buyer the larger is the probability that the seller will accept that.

The negotiation framework we use is similar to the screening game in the literature of bargaining games with asymmetric information. As in the case of the screening game, one of the agents in our framework has full information about the indivisible item being negotiated, but the other agent, without this information, moves first. In our setting, the buyer always gains if there is a contract. If the seller rejects the proposal, the buyer gets nothing. So, the buyer's goal is to make the contract while keeping the price as low as possible. Under complete information, if the negotiation is played for a single time, the subgame perfect equilibrium exist. In equilibrium, the buyer proposes a small amount more than the outside options available to the seller and keeps the rest of the amount and the seller agrees to it. But the assumption of complete information is a limitation as in most scenarios an agent will not know the other agent's outside option and reservation price. Under incomplete information the equilibrium strategy will be for the buyer to offer its reserve price to the seller and for the seller to accept if it is more than the outside options and reject otherwise.

The scenario is more interesting when this game is repeated between the same buyer and seller. In real life, repeated negotiation is common in many contracting scenarios, and in particular, E-negotiation and different retail relationships. Though repeated interaction is more realistic, unlike the single interaction case, the existing literature does not provide a clear understanding of such

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

scenarios. In this paper we will focus on the repeated negotiation scenario. The objective of the buyer is to approximate the seller's decision function closely and as early as possible to be able to produce an offer that maximizes its long-term profit.

The ability to model the behavior of other agents in an MAS can assist an agent to improve its performance in both cooperative and competitive situations [14]. For example, the ability to model the behavior of another agent can enable the modeling agent to predict the behavior of the other agent provided they sense the world similarly. In a cooperative setting, the ability to predict the actions to be taken by another agent will enable the modeling agent to choose its actions to coordinate with the other agents. In a competitive setting, the ability to predict the behavior of the opponent will provide the modeling agent with the opportunity to choose exploitative actions to maximize its own utility. The latter approach is, for example, adopted by game playing agents that can learn the strategy being used by its opponent [2].

Observation-based coordination is a very attractive and plausible approach to developing coherent behavior in an MAS which alleviates the need for explicit communication between agents [8]. An observation-based plan-recognition approach uses general domain information and observed actions of another agent to predict future plans of that agent. This may not necessarily lead to a modification of the model of the other agent. Our goal is complementary in the sense that we want to observe past actions taken or decisions made by another agent to identify the decision policy being used by that agent. The model thus generated can then be used as a predictive mechanism to anticipate future decisions to be taken by the modeled agent.

An important characteristic that is desirable for modeling procedures in an MAS is that it can be used on-line by agents. This in turn require two things: (a) the procedure should be able to incrementally build models as more and more information is received about the behavior of other agents, (b) the procedure should be easy to compute. In this paper, we have developed a computationally cheap algorithm, using Chebychev polynomials, with proven convergence properties to approximate arbitrary probability functions over an input variable defined over the range [-1,1]. The algorithm is incremental in nature, i.e., it can develop incrementally better models as more and more data (observations) become available. The computational time for incremental updates is linear in the number of orthogonal polynomials used, and hence the computational costs are minimal. We prove that under infinite sampling, the algorithm is guaranteed to converge to the actual underlying probability function. Using extensive experiments, we have shown that the agents using this algorithm outperforms other simple opponent learning algorithms and other representative reactive heuristics of bargaining. We have also experimentally shown that this modeling procedure is robust to noisy data.

2. BUYER AGENT BEHAVIORS

In this section, we describe the following four agent strategies. We assume that buyers use these strategies to bargain with the seller. Each of the buyer agent knows a lower limit, *low*, and a higher limit, *high*, such that the probability that the seller will accept any offer greater than or equal to *high* is 1 and any offer less than or equal to *low* is zero.

Chebychev buyer: This strategy initially explores by playing random offers in the interval [*low, high*]. After exploring for some interactions it approximate the opponent decision function by Chebychev polynomials. We discuss this approximation in the next section. It then plays the offer that maximizes its expected utility. We define utility as *high* minus the offer.

Risk averse buyer: This strategy has an aversion to lose the contract. Though an agent with this strategy starts offering with *low*, if the seller rejects its offer it increases its offer by *incr* in the next interaction. We can interpret this strategy as a safe strategy which tries to maximize the probability of acceptance.

Risk seeking buyer: This is a risk loving strategy. In this strategy, agent starts bargaining with an offer of *low* and increases offer by *incr* in the next interaction if its current offer is rejected consecutively for the last 5 times. But if the seller accepts its offer even once, it will never increase its offer subsequently.

Step buyer: In this strategy agent tries to model opponent's probabilistic decision function by relative frequencies. It partition the interval [*low, high*] in T equal intervals, $(x_{i-1}, x_i]$, where i=1..T. Initially it offers randomly in the interval [*low, high*] for exploration, similar to the Chebychev buyer strategy. Then it approximates the opponent's decision function as the proportion of success in each interval. *i.e.*, if there are o_i number of offers in the i^{th} interval $(x_{i-1}, x_i]$ and out of them s_i times seller accepted the offer, then probability of acceptance for that interval is taken to be $p_i = \frac{s_i}{o_i}$. Then Step buyer offer x_{opt} where $opt = argmax_i((high - x_i) * p_i)$.

3. CHEBYCHEV POLYNOMIALS

Chebychev polynomials are a family of orthogonal polynomials [6]. Any function $f(x)$ may be approximated by a weighted sum of these polynomial functions with an appropriate selection of the coefficients.

$$f(x) = \frac{a_0}{2} + \sum_{i=1}^{\infty} a_i * T_i(x)$$

where

$$T_n(x) = \cos(n * \cos^{-1}(x))$$

and

$$a_i = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) * T_i(x)}{\sqrt{1-x^2}} dx$$

Working with an infinite series is not feasible in practice. We can, however, truncate the above series and still obtain an approximation, $\hat{f}(x)$, of the function [4]. The Chebychev polynomial converges faster than the Taylor series for the same function. For a rapidly converging series, the error due to truncation, i,e., using only the first n terms of the series, is approximately given by the first term of the remainder, $a_n T_n(x)$. We have chosen Chebychev polynomials for function approximation because truncation points can be chosen to provide approximate error bounds.

Function approximation in mathematics is obtained using orthogonal polynomials. This requires the knowledge of the value of the function at certain input values. In our case, however, instead of exact values of the probability function, we observed only *True* or *False* decisions based on a sampling from the probability function at different input values. This, then, constitutes a novel application of Chebychev polynomials.

A viable mechanism for model development based on Chebychev polynomials would require the development of an algorithm for calculating the polynomial coefficients. We would also need to prove that this algorithmic updates would result in a convergence of the learned function to the actual probability function underlying the sampled data. We provide such an algorithm and the associated convergence theorem in the next section.

4. AN ALGORITHM TO LEARN DECISION POLICY

Let $f(x)$ be the target function and $\hat{f}(x)$ be the approximation of it based on the set of samples $S = \{S_j\}$, where $S_j = \langle x_j, v_{x_j} \rangle \forall j = 1, 2, \ldots, k$ and $k =$ Number of instances. We have $v_{x_j} \in [True, False]$. We may have to change the scale for the values x_j, so that all the values are in the range $[-1 : 1]$ and we get the approximated function in the range $[-1 : 1]$, which we may need to scale back to get the desired value.

Let n be the no of Chebychev polynomials we are going to use for the purpose of learning. Let $T_i(x),\ i \in [0..n]$ be the Chebychev polynomials. The steps of the algorithm to calculate the polynomial coefficients are:

1. Initialize $C_i = 0, \quad \forall i = 0, 1, 2, \ldots, n$
2. For all j do
3. $\forall i = 0, 1, 2, \ldots, n$
   If v_{x_j} =T then,
   $$C_i \leftarrow C_i + \frac{T_i(x)}{\sqrt{1-x^2}}$$
4. $\forall i = 0, 1, 2, \ldots, n$
   If v_{x_j} =F then,
   $$C_i \leftarrow C_i - \frac{T_i(x)}{\sqrt{1-x^2}}$$
5. End for
6. Set
   $$\hat{f}(x) = K * (\frac{C_0}{2} + \sum_{i=1}^{n} C_i * T_i(x))$$
   where $K = \psi(k)$, is function of number of interactions.
7. Set
   $$\hat{f}(x) \leftarrow \hat{f}(x) + 0.5$$

Theorem 1 Under complete information the algorithm can approximate the probability function.
Proof: See Appendix.

5. MODELING SAMPLE PROBABILITY FUNCTIONS

In this section, we evaluate the performance of the algorithm presented in the previous section when limited amount of decision samples are available. As a sample target function to test the approximation algorithm, we used a sigmoidal function: $f(x) = \frac{1}{1+\exp(5*x)}$. The actual function and the polynomial approximation obtained for this functions obtained with different number of samples are presented in Figure 1. We focus on the effect of the number of decision samples available on the accuracy of the model developed. We vary the sample size, $k =$ 30, 100, and 1000. Sample x values were first randomly selected from a set of points in the range [-1,1]. The underlying probability function was used to calculate *True* or *False* decisions for these generated points. Corresponding input-decision pairs were then sequentially fed into the algorithm. It is clear from Figure 1 that increasing the number of samples improves the approximation. Though 30 samples produce only a rough approximation of the target function , a fairly close approximations obtained with about 100 samples.

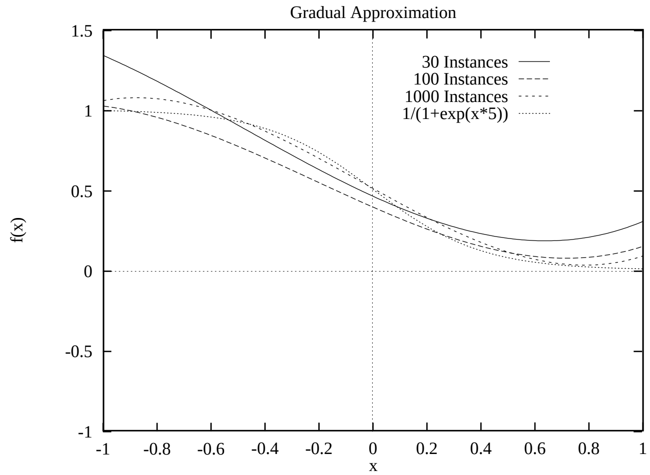

Figure 1: Approximating $f(x) = \frac{1}{1+\exp(5*x)}$. Number of Chebychev Polynomials used is 5 and sampling resolution is 100.

6. EXPERIMENTAL FRAMEWORK AND RESULTS

We have designed our experimental framework to evaluate the effectiveness of the bargaining strategy we have proposed under different negotiation situations. In our simulation, we use the four representative bargaining strategies discussed in Secion 2. To describe our experimental results, we use the following conventions: Chebychev buyer, Risk Averse buyer, Risk seeking buyer and Step buyers are represented by CB, RA, RS and SB respectively. We also consider a hypothetical KN buyer who knows the seller's decision function exactly. We use the performance of the KN buyer as the yardstick for measuring the effectiveness of the other strategies. Unless stated otherwise, number of Chebychev polynomials used by CB is 5. The SB buyer uses $T = 50$. Both these buyers use 600 explorations to approximate the function and subsequently choose offers based on the approximated seller model.

We assume a scenario where a seller produces an unique item in each time period. Such a scenario is common in market situations where the demand of an item is more than the supply and there exists a number of buyers wanting to buy an item for sell. Based on expected outside options and risk attitude, a seller has a decision function $F(x)$, a probability function that gives the probability of acceptance an offer of x. This decision function is an intrinsic property of the seller. We have experimented with different monotonically increasing functional forms for seller decision function $F(x)$. For a given offer, x, $F(x)$ returns the probability of the seller to accept that offer. So, when a particular offer x is made, a biased coin with weight $F(x)$ is tossed by the seller to determine whether or not to accept that offer. We sequentially choose the different buyer strategies and negotiate against the same seller for 2000 times. We have also experimented with sellers which has a "noisy decision function" and with sellers which change their decision functions slowly.

We assume that the upper and lower valuation limits of the item are known to the buyers from experience. For example, if one wants to buy a simple DVD player, he knows a loose upper and lower bound of the price. In our experiments, the buyer knows that the value of the item ranges between $900 to $3100. So in this set up

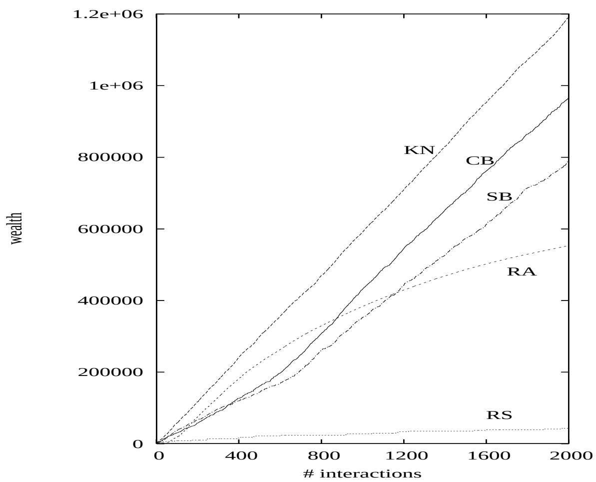

Figure 2: Wealth earned by different buyer types. Seller's decision function is given in Equation 1. Exploration time = 600

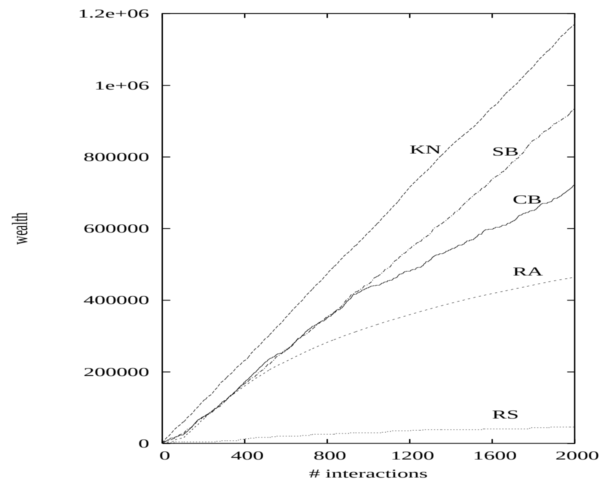

Figure 3: Wealth earned by different buyer types. Seller's decision function is given in Equation 1. Exploration time = 20

the buyer will never propose anything outside this price range.

In the first set of experiments, the seller's decision function is

$$\begin{aligned} F_1(x) &= 1 - \frac{1}{1+\exp\frac{5*(x-2000)}{1000}}, \text{ if } x \in (1000, 3000). \\ &= 1, \text{ if x } \geq 3000 \\ &= 0, \text{ if x } \leq 1000 \end{aligned} \quad (1)$$

Figure 2 shows the wealth generated by different buyer strategies playing against the above seller. The figure shows that the CB outperforms the other buyer strategies. After explorations, its performance is almost same as KN (the two lines become approximately parallel after 600 iterations), which means that it performs as if it had access to the actual decision function used by the seller. The performance of SB comes next to CB as it also approximates seller's decision function though not as effectively. In related experiments we find that with higher number of intervals, T, and sufficiently long exploration, SB can match the performance of CB and KN. But the losses suffered during the extensive exploration periods does not make that strategy competitive with CB. The RA buyer is doing better than RS buyer as most of the time it makes successful offers though keeping very low profit to itself. But this is better than the stubborn strategy of the RS buyers whose offers are rejected by the seller most of the time.

The second experiment is similar to the first one except that the CB buyer is using only 20 iterations for exploration. From Figure 3 we see that the CB buyers is performing worse than the SB buyer. The CB cannot approximate the decision function closely with this minimal exploration. As a result, the model learned is not effective and leads to worse performance compared to SB. Note that the agent continues to learn throughout the experiment, including outcomes from each iteration and re-calculating model approximations. But as these new samples are only for the offer for which expected utility is maximum, the overall input space is not being sampled, and hence no noteworthy improvement in the model approximation is produced.

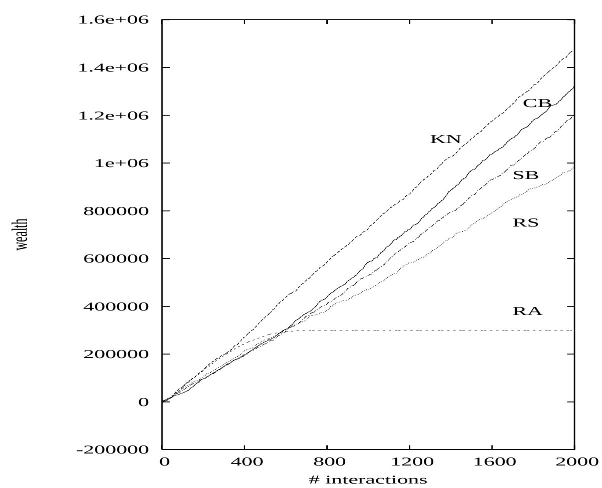

Figure 4: Wealth earned by different buyer types. Seller's decision function is given in Equation 2. Exploration time = 600

In the next experiment, we use a new decision function

$$\begin{aligned} F_2(x) &= \frac{(x-1000)^{0.4}}{(2000)^{0.4}}, \text{ if } x \in (1000, 3000). \\ &= 1, \text{ if x } \geq 3000 \\ &= 0, \text{ if x } \leq 1000 \end{aligned} \quad (2)$$

The function in Equation 2 makes the seller inclined to accept lower offers compared to the one in the first experiment. The wealth generated by the different buyers against this seller is shown in Figure 4. Since sufficient exploration time is given, the CB buyer again outperforms all the other learning buyers and performs as good as KN buyer after the exploration period. The interesting result in this case is the better performance of RS buyers compared to RA buyers. Since the probability of acceptance is higher here, the RS buyers manage to eke out profitable contracts through tough bargaining.

Next we experiment with a somewhat inconsistent seller. The

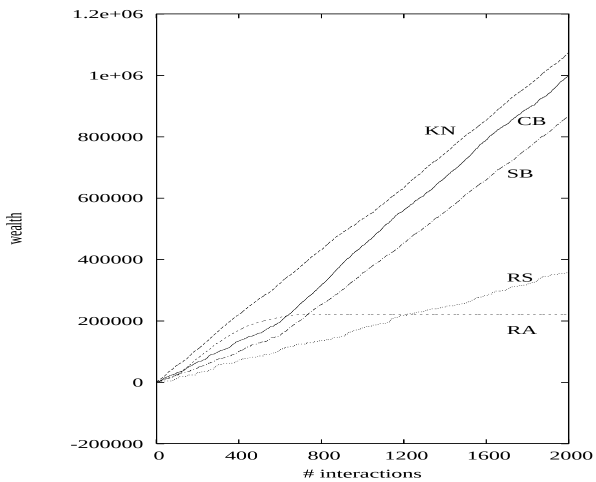

Figure 5: Noisy environment with Gaussian noise $\sim$ Gaussian(0,0.05)

experiment setting used is same as first experiment but here we add a Gaussian noise $\epsilon \sim N(0, 0.05)$ to the probability function, i.e., we use $F'(x) = F_1(x) + N(0, 0.05)$. The Figure 5 shows that CB buyers handle this noise more efficiently compared to the other buyers. After exploration it outperforms all the other adapting buyers and approaching close to the baseline KN buyer.

So we find that over a range of scenarios buyers using CB strategy are doing significantly better than other buyers using representative bargaining strategies when playing against the same opponent repeatedly. Now we turn to evaluating the CB buyers under dynamic environmental conditions.

Comparing Figure 2 and Figure 3, we concluded that CB needs to explore enough in order to closely approximate the decision function and increases profit. We now present experiments with a seller which becomes less inclined to accept the same offer in subsequent interactions. After every 200 iterations the seller decreases its probability of acceptance by replacing the previous $F(x)$ by $F(x - 20)$. Results from this scenario (see Figure 6) show that the CB strategy is performing much worse compared to the KN buyer. This is because of the fact that after the initial exploration phase, the buyer continues to use an outdated seller model. To rectify this problem the buyer needs to explore at some regular intervals to capture the changes in the seller policy. But the exploration process involve a cost due to sub-optimal offer presentations. This gives rise to two crucial questions: a) how frequently the agent should explore and b) how much the agent should explore in each exploration period. We use the following exploration schedule: a buyer explores for τ consecutive offers (*exploration-frequency*) every β offers (*inter-exploration interval*).

Figure 6 we present plots of performance of CB for different values of β when $\tau = 10$. We find that too frequent explorations adversely affect performance because then the agent does not have enough interactions in between exploration periods to recoup losses during exploration. We see that the agent even performs worse for $\beta = 50$ compared to $\beta = \infty$, i.e., not exploring at all. But it manages to do better if $\beta = 200$. In the next set of experiments (see Figure 7) we fix $\beta = 200$ and vary τ from 5 to 50. We find that for long exploration sequences, e.g., $\tau = 50$, the buyer performs even worse than when it does no periodic exploration. This is because of the high cost of exploration. But with small exploration frequencies, e.g., 5, the buyer is able to keep pace with changes in the seller decision function. Of course, the optimal inter-exploration interval depends on how often the seller changes its decision function and the optimal exploration-frequency depends on the rate of change of the seller decision function.

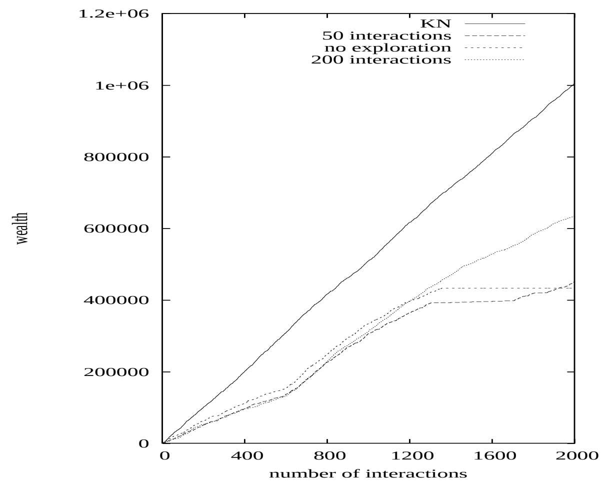

Figure 6: Wealth earned by the CB buyers using different *inter-exploration interval*

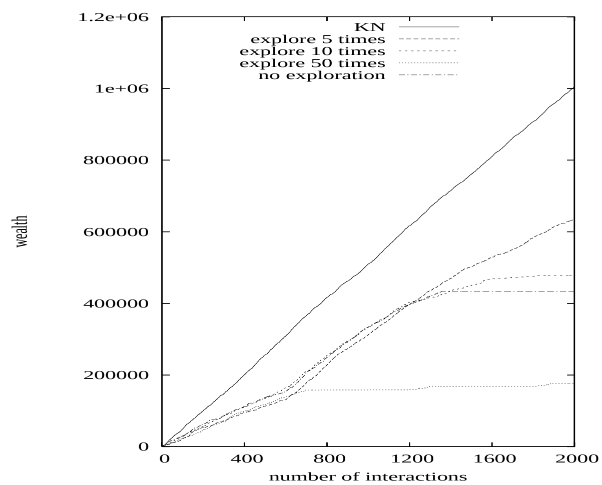

Figure 7: Wealth earned by the CB buyers using different *exploration-frequency*

7. RELATED WORK

Negotiation has long been studied [13] in Economics, Business, Law and Computer Science. In the past few years, automated negotiation has been studied extensively in the multiagent community [9, 10]. Research in multiagent systems addresses several negotiation issues like time constraints [10], deadline [15], outside options [11], different incomplete information settings [5], etc. Our research is different from all of these as we focus on modeling opponent decision function in repeated one-shot negotiation. We approximated opponent decision model using Chebychev's polynomials and use this to determine the optimal offer.

Carmel and Markovitch have used restrictive assumptions about the behavioral strategies being used by other agents to derive a simple finite automata model of another agent from its input/output behavior [3]. They then proceed to derive an optimal interaction strategy with that opponent. The work presented in this paper can develop models of more complex decision policies. Bui *et al.* [1] uses past responses about meeting scheduling requests to develop a probability distribution of open time slots in the calendar of other agents. Gmytrasiewicz and Durfee presents a decision-theoretic approach to updating recursive models of other agents [7]. Their model updating procedure, however, is based more on assumptions of rationality of the other agent and is not dependent on observed behavior of other agents. Zeng and Sycara present a learning mechanism by which agents can learn about payoff structures of other agents in a sequential negotiation framework [16]. The knowledge acquired by these agents if more of a threshold of acceptance rather than a general decision procedure over the range of input values.

8. CONCLUSION AND FUTURE WORK

We have experimentally evaluated the relative effectiveness of different buyer strategies for modeling the seller decision function under a repeated one-shot negotiation framework. We study a particularly interesting learning problem of inferring a probabilistic decision function over an input space based only on sampled yes/no responses from the input space using that probability function. Hence the learning problem studied here is very different from traditional function approximation problem where actual sampled function values are available as inputs to the model learner. We propose a novel learning approach using Chebyshev's polynomials and show its performance using extensive experimental settings. For most environmental settings, our modeling approach outperforms representative bargaining heuristics while repeatedly negotiating against a seller with a probabilistic decision function. We also evaluated the robustness of this modeling approach against noisy decisions.

We plan to extend this work in several directions. A very promising area of research would be to develop modeling of opponent decision functions in multi-step, instead of one-shot negotiation scenarios. Strategic information revealing during early negotiation stages can be used both to guide the negotiation process and signal preferences to a learning opponent. We also plan to extend this work to multi-issue bargaining and use this and other learning techniques to learn the preference structure of the agents in multi-issue bargaining.

Acknowledgements: This work has been supported in part by an NSF award IIS-0209208.

9. REFERENCES

[1] H. H. Bui, D. Kieronska, and S. Venkatesh. Learning other agents' preferences in multiagent negotiation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 114–119, Menlo Park, CA, 1996. AAAI Press.

[2] D. Carmel and S. Markovitch. Incorporating opponent models into adversary search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 120–125, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[3] D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Thirteenth National Conference on Artificial Intelligence*, pages 62–67, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[4] P. O. W. Curtis F. Gerald. *Applied Numerical Analysis*. Addison-Wesley Publishing Company, Menlo Park, CA, 1992.

[5] S. S. Fatima, M. Wooldridge, and N. Jennings. Optimal negotiation strategies for agents with incomplete information. In *Intelligent Agents VIII*, Lecture Notes in AI, 2333, pages 377–392. Springer-Verlag, March 2002.

[6] L. Y. Geronimus. *Orthogonal Polynomials*. Consultants Bureau, New York, NY, 1961.

[7] P. J. Gmytrasiewicz and E. H. Durfee. A rigorous, operational formalization of recursive modeling. In *First International Conference on Multiagent Systems*, pages 125–132, Menlo Park, CA, 1995. AAAI Press/MIT Press.

[8] M. Huber and E. H. Durfee. An initial assessment of plan-recognition-based coordination for multi-agent teams. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 126–133, Menlo Park, CA, 1996. AAAI Press.

[9] N. Jennings, P. Faratin, A. R. L. S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[10] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

[11] C. Li, J. A. Giampapa, and K. Sycara. Bilateral negotiation decisions with uncertain dynamic outside options. In *The First IEEE International Workshop on Electronic Contracting*, pages 54–61. IEEE Computer Society, July 2004.

[12] A. Muthoo. *Bargaining theory with applications*. Cambridge University Press, New York, NY, USA, 1999.

[13] J. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[14] P. Riley and M. Veloso. Planning for distributed execution through use of probabilistic opponent models. In *Sixth International Conference on AI Planning and Scheduling*, pages 72–81, 2002.

[15] T. Sandholm and N. Vulkan. Bargaining with deadlines. In *AAAI/IAAI*, pages 44–51, 1999.

[16] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human Computer Studies*, 48, 1998.

APPENDIX

Proof of Theorem 1

We know, that any function can be represented as a combination of Chebychev polynomials as,

$$f(x) = \frac{a_0}{2} + \sum_{i=1}^{\infty} a_i * T_i(x)$$

Next, we are going to prove that the function $\hat{f}(x)$ tends to f(x) as more and more samples of the function, of greater resolution is received. We have

$$C_i = \frac{}{\pi} * \{\sum_{j=1}^{k} \frac{(f(x_j) + \triangle f(x_j))}{\sqrt{1 - x_j^2}} * T_i(x_j) + \sum_{j=1}^{k} \frac{1 - (f(x_j) + \triangle f(x_j))}{\sqrt{1 - x_j^2}} * (-T_i(x_j))\},$$

where is a function of no of interactions and can be given by $= \frac{C}{k}$ and $\triangle f(x) = |f(x) - \hat{f}(x)|$ and $\triangle f(x)$ is in the order of $\frac{1}{k}$. So,simplifying we get,

$$C_i = \frac{}{\pi} * \{2 * \sum_{j=1}^{k} \frac{f(x_j) + \triangle f(x_j)}{\sqrt{1 - x_j^2}} * T_i(x_j) - \sum_{j=1}^{k} \frac{T_i(x_j)}{\sqrt{1 - x_j^2}}\}$$

Now, in case of infinite sampling and infinite resolution, we have,

$$k \to \infty \Rightarrow \quad \to 0 \Rightarrow \triangle f(x) \to 0.$$

So for $i = 0$, i.e. the coefficient becomes,

$$C_0 = \lim_{\phi \to 0} \frac{}{\pi} * \{2 * \sum_{j=1}^{k} \frac{f(x_j)}{\sqrt{1 - x_j^2}} * T_0(x_j) - \sum_{j=1}^{k} \frac{T_0(x_j)}{\sqrt{1 - x_j^2}}\}$$

$$C_0 = \lim_{\phi \to 0} \frac{1}{\pi} * \{2 * \sum_{j=1}^{k} \frac{f(x_j)}{\sqrt{1 - x_j^2}} * T_0(x_j) * \quad - \sum_{j=1}^{k} \frac{T_0(x_j)}{\sqrt{1 - x_j^2}} * \quad \}$$

$$C_0 \ = \ \frac{}{\pi} * \{2 * \int_x \frac{f(x)}{\sqrt{1 - x^2}} * T_0(x)dx - \int_x \frac{T_0(x)}{\sqrt{1 - x^2}} dx\}$$

As, the x range is [-1:1] and in case of infinite sampling and resolution we have information at all the values of x, the integral limits becomes,

$$C_0 \ = \ \frac{1}{\pi} * \{2 * \int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}} * T_0(x)dx - \int_{-1}^{1} \frac{T_0(x)}{\sqrt{1 - x^2}} dx\}$$

$$\Rightarrow C_0 \ = \ \{\frac{2}{\pi} * \int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}} * 1dx - \frac{1}{\pi} * \int_{-1}^{1} \frac{1}{\sqrt{1 - x^2}} dx\}$$

$$\Rightarrow C_0 \ = \ a_0 - \frac{1}{\pi} * \pi = a_0 - 1$$

As, $T_0(x) = 1$. Similarly, $\forall i = 1, 2, \ldots, n$,

$$C_i \ = \ \frac{1}{\pi} * \{2 * \int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}} * T_i(x)dx - \int_{-1}^{1} \frac{T_i(x)}{\sqrt{1 - x^2}} dx\}$$

$$\Rightarrow C_i \ = \ \{\frac{2}{\pi} * \int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}} * T_i(x)dx - \frac{1}{\pi} * \int_{-1}^{1} T_i(x)dx\}$$

$$\Rightarrow C_i \ = \ a_i$$

as from orthogonality property,

$$\int_{-1}^{1} \frac{T_i(x)}{\sqrt{1 - x^2}} dx = 0, \ \ \forall i = 1, 2, \ldots, n$$

Given,

$$\hat{f}(x) = \frac{C_0}{2} + \sum_{i=1}^{\infty} C_i * T_i(x)$$

$$\hat{f}(x) = \frac{a_0}{2} - 0.5 + \sum_{i=1}^{\infty} a_i * T_i(x)$$

$$\hat{f}(x) \ = \ f(x) - 0.5,$$

and therefore in the last step of the algorithm we increment $\hat{f}(x)$ by 0.5.