

Learning Social Preferences in Games

Content areas: Cognitive modeling, negotiation, game playing

Tracking number: 376

Abstract

This paper presents a machine-learning approach to modeling human behavior in one-shot games. It provides a framework for representing and reasoning about social preferences which affect people's play. The model predicts how a human player is likely to react to different actions of a computer player, and these predictions are used to determine the best possible strategy for the computer player. Data collection and evaluation of the model was performed on a negotiation game in which humans played against each other, and against computer models playing various strategies. When playing with people, our model performs better than Nash equilibrium and Nash bargaining computer players, as well as humans. It can also generalize to players and situations it had not seen before.

Introduction

Modern technology is opening up vast opportunities for computer agents to interact with people. Many of these interactions can be modeled as games, in which the human and computer players each have their own choices of actions and their own goals. Classical game theory has been widely adopted as the basis for designing computer agents that interact with humans. However, behavioral economics has shown that people actually play quite differently from classical game theory predictions. A variety of psychological factors, different from the pure self-interest of classical game theory, influence people's play. To design computer agents that interact successfully with humans, these factors must be taken into account.

In this paper, we use machine learning to learn models of how humans actually play games. We explicitly model the different social preferences that a person might have, ranging from pure self-interest to altruism, and including notions of fairness. To capture these principles, we assume that players use a social utility function that is a combination of these social preferences. The model learns the social utility function used by different types of people, and also a probability distribution over which utility function a given player uses. Since each utility function provides a guideline for behavior,

learning the utility functions is key to developing a model that can predict which action a player will take.

The learned models were incorporated into a computer agent that interacts with humans and is used to predict how a human player will likely react to different actions of the computer player. These predictions are used to determine the best possible strategy for the computer player.

We have implemented our methodology in a two-player negotiation game. The game takes place in the Colored Trails framework, developed by Grosz and Kraus (2004), an environment in which each player has a goal and certain resources are needed to reach it. The players can trade resources, leading to interesting negotiation scenarios. We study a scenario in which one player proposes a trade to the other, who then has the opportunity to accept or reject. Our model learns to predict whether the second player accepts or rejects the offer. This model is then used to help the first player decide what offer to make.

We tested our methodology in experiments involving human subjects. In the first phase of the experiments, data was collected from observing human play. This data was used to learn a model of human play. In the second phase, a computer player using the learned model was compared against two computer players and against the performance of human players themselves. Our computer player was able to perform better than all the other players.

Our approach to learning has been to generalize from the behavior of some human players to the behavior of other players. In contrast, most of the work on learning in games (Fudenberg & Levine 1998) has been on learning the strategy of a particular opponent from repeated play against that opponent. For example, opponent modeling has been applied successfully to poker (Davidson *et al.* 2000).

Common to the techniques in this approach is that they focus on learning and evolution within a single repeated game, in which the agent keeps playing against the same opponents. In contrast, we aim to develop agents that interact well with human agents, even those they have never seen before. The agents should also play reasonably in game situations they have never encountered before.

There has been work in AI with regard to learning utility functions. both in MDP (Ng & Russell 2000) and probabilistic frameworks (Chajewska, Koller, & Ormoneit 2001). Our work differs from these in fundamental ways. First, they

focus on learning the utility function of a single player by observing actions of that player. Their approach does not provide a prediction for players that have never been encountered before. In contrast, our model considers several types of players, and generalizes to playing new people for the first time. Second, they learn a utility function directly in terms of outcomes of the game. There are no factors that are external to the game that enter into the utility function, and there is no consideration of behavioral aspects in their model. In contrast, we learn utility functions that are based on social preferences. Third, their approaches assume that agents automatically follow their utility functions without exceptions. Our framework is more flexible. We allow for the possibility that their actions will sometimes contradict their supposed utility function. This approach allows us to handle noise in people's decisions.

Behavioral Decision-Making

Behavioral economics work expresses parsimonious "social utility" functions, representing an individual's level of satisfaction with an outcome. It has been shown that social norms play a crucial part in players' reasoning, both in repeated and in one-shot interactions. For example, reciprocal behavior has been shown to appear in the ultimatum negotiation game (G. Werner 1982), contradicting the predictions of traditional game theoretic models. The unique sub-game perfect Nash equilibrium of the game is to offer the smallest amount possible to the other player, and for the other player to agree to the proposal. However, numerous experiments have shown that individuals behave differently; most offers consist of half of the goods, and most rejections occur for offers consisting of less than a quarter of the goods.¹ These results imply that players engage in a tradeoff between *interpersonal preference* and strategic considerations. Recently, several social factors have been given formal definitions in the literature, and referred to as *social preferences*.

Individual Benefit A key motivation for players is to maximize their individual utility, as specified by the rules of the game. This is the sole motivation considered by classical game theory.

Altruism A player may be interested in the welfare of the group as a whole, as well as her own utility. Such a player might be willing to sacrifice some individual utility for the sake of social welfare.

Equality of Outcome A player who cares about fairness may be concerned that the outcome be as equal as possible for all players. Such a player might be willing to sacrifice individual utility to ensure a more balanced outcome, or to reject an otherwise beneficial trade if it leads to an imbalance outcome.

For example, Bolton (1991) assumes people care about the difference between their own payoff and that of their opponent, in addition to caring about their own payoff, when making a decision. Charness and Rabin(2002) present a theory in which players care about their own payoff (selfish-

¹In some societies, exceptionally large offers of the goods were rejected.

ness), the minimum payoff (guilt), and the maximum payoff (envy).

Other work includes analysis of the equilibrium in which each player is maximizing her own social utility function. Loweinstein et.al (1989) compared several social utility forms and found that a utility function including terms for own payoffs, as well as a separate term for positive and negative discrepancies between the parties' payoffs, closely matched data corresponding to one-shot dispute type negotiation.

Nash, who addressed the closely related topic of bargaining (J.Nash 1971), gave several axioms which every reasonable bargaining solution must follow. He then showed that the deal which maximizes the product of the agents' utilities is the sole solution to the set of axioms. An interesting point is that the Nash bargaining solution is always Pareto optimal, meaning that some solution must exist in which both parties are no worse off than not negotiating. In contrast, social preference models are able to describe situations in which altruistic players agree to deals which leave the other player better off, but they are left worse off.

However, there has been little work to date on *learning* social preferences through repeated observation of human play. If human behavior presents particular regularities such as described above, then it should be possible to identify the factors influencing their behavior through the use of computational modeling. Rather than have the structure of a social utility function reflect the bias of the modeler, it could be shown to evolve through learning, and social norms such as reciprocity and fairness would be given justification based on actual data. Moreover, a sound computational framework of learning would be able to generalize to new situations and examples of interaction that were not seen before.

Colored Trails

Our study used the game Colored Trails(CT), designed by Grosz and Kraus. CT is played on an NxM board of colored squares with a set of tiles in colors chosen from the same palette as the squares. One square is designated as the "goal square" and each player has a piece on the board, initially located in one of the non-goal squares. The players also have a set of colored tiles. To move a piece into an adjacent square a player must turn in a chip of the same color as the square. Tiles may be exchanged by the players, and the conditions of exchange may be varied to model different decision-making situations.

A player's performance in CT is determined by a scoring function. This function may depend on many factors, such as the player's distance from the goal-square, the number of moves made, and the number of tiles the player possesses at the end of the game. In addition, a player's performance in the game can be made to depend on the performance of other players, by including the score of other players in her own scoring function.

For our study, we use a version of CT which includes two players playing on 4x4 boards and a palette consisting of 4 colors. Each player has full view of the board, as well as the other player's tiles. At the beginning of the game, the two players are randomly placed at two locations on the CT

board. Each player is allocated four tiles at random, which could include any color in the palette. The distribution of tiles is designed such that it is likely that the game is “interesting”. A game is considered to be interesting if (1) at least one of the players can reach the goal after trading with the other player; (2) it is not the case that both players can reach the goal without trading.

The scoring function for the players was set as follows.

- 100 points bonus for reaching the goal.
- 5 points for each tile left in a player’s possession.
- 10 points reduced for any square in the path between the player’s final position and the goal-square. This path is computed by the Manhattan distance.

The parameters were chosen so that while getting to the goal is by far the most important component, if a player cannot get to the goal it is preferable to get as close to the goal as possible. Note that a player’s outcome is determined solely by her own performance.

Each player is designated a role, which determines the possible actions that are available to her during the game. One player is the *allocator*. She is allowed to propose an offer for exchange of tiles to the other player, who is the *deliberator*. In turn, the deliberator can either accept or reject the allocator’s offer. If the allocator does not make any offer, then both players are left with their initial allocation of tiles. We do not allow for a deliberator to counter the allocator’s offer with another proposal. The score that each player receives if no offer is made is identical to the score each player receives if the offer is rejected by the deliberator. We refer to this event as the *no negotiation alternative*. The score that each player receives if the offer is accepted by the deliberator is referred to as the *proposed outcome score*.

Under the conditions specified above, each game consists of a one-shot negotiation deal between the two players, and a deliberator’s reply to the exchange proposed by the allocator completely determines the final outcome of the game.

Model Construction

In this work, we attempt to model human deliberators in the CT game. Our task is to learn a predictive model, that will predict whether a deliberator will accept a given proposal. The inputs to the model are NN_A and NN_D , the no-negotiation alternative score for the allocator and deliberator, and PO_A and PO_D , the proposed outcome for the allocator and deliberator.

In order to develop the model, we introduce the following features, which represent possible social preferences for the deliberator:

- Individual Benefit $PO_D - NN_D$
- Aggregate Utility $PO_D + PO_A$
- Inequality of Outcome $PO_D - PO_A$
- Fair Trade $(PO_D - NN_D) - (PO_A - NN_A)$

As discussed in the section on behavioral economics, the features individual benefit, inequality and aggregate utility match social norms that have been shown to contribute to

players’ reasoning in the behavioral economics literature. We distinguish between two types of inequality. The first is inequality in the final outcome achieved. The second is inequality in the gains received from trade. Since players may conceivably care about either, we include both in our model. Note that the Inequality of Outcome and Fair Trade features are not symmetric. The assumption is that the deliberator will care about inequality when she receives less than the allocator, but will not mind when she receives more than the allocator.

Given any proposed exchange x , we assume that a particular deliberator’s utility u is a weighted sum of features. The utility function is determined by the weights $\mathbf{w} = \langle w_{NN_A}, w_{NN_D}, w_{PO_A}, w_{PO_D} \rangle$. The weights measure the relative importance of each of the social preferences to the deliberator. The utility function is normalized around 0, so a utility of 0 corresponds to indifference between accepting the proposal and rejecting it. We interpret the utility to be not only an indication of which decision to make, but also the degree to which one decision is preferred. Thus accepting a proposal is more strongly preferred when the utility is a large positive number than a small positive number.

While a particular deliberator may use some particular weighting of features, it is unlikely that all deliberators behave the same way. We could try to learn a single utility function representing the aggregate preferences of all deliberators, but this would be unlikely to fit many particular deliberators well. On the other hand, if we try to learn a separate utility function for each deliberator, we would need to play against the same deliberator for many rounds before we could learn about them, and we would not be able to generalize from one deliberator to another. Therefore we assume that there are several *types* of deliberator, where each type has its own utility function. We use a mixture model over types, with a probability distribution $P(t)$ over the set of types. Each type t is associated with its own set of social preference weights \mathbf{w}^t , defining a utility function u^t .

In addition, while a utility function corresponds to a decision rule, we do not assume that deliberators implement the decision rules perfectly. This is for two reasons. First, the deliberators’ play may be noisy — they may make mistakes and act capriciously. Therefore we can expect there to be noise in the data that could not be explained if the deliberators played perfectly. Second, even if a deliberator falls under a certain type, it is unlikely that her utility function will exactly match that of the type. Rather, it may differ in small ways, which cause her to take decisions that are contrary to the type’s utility function.

To capture the fact that a decision rule might be implemented noisily, we use a sigmoid function. We define the probability of acceptance $P(\text{accept}|x, t)$ for a particular exchange x by deliberator of type t to be $\frac{1}{1+e^{-u^t(x)}}$. This function has the right properties for modeling the play resulting from a utility function. The probability of acceptance converges to 1 as the utility becomes large and positive, and to 0 as the utility becomes large and negative. Meanwhile, when the decision is less clear-cut, i.e. the utility is close to zero, “mistakes”, or decisions that are contrary to the utility function, are more likely to happen.

Given that we have a model describing the deliberator’s behavior, the next step is to incorporate this model in to a computer agent that plays with humans. We will assume that the computer agent plays the allocator and that a human is playing the deliberator. The goal of this step is to maximize the score of the allocator, by exploiting the model of how the deliberator acts. The strategy is to propose the deal that maximizes the expected utility to the allocator. The expected utility is the sum of the allocator’s utility of the proposal times the probability the proposal is accepted, and the allocator’s no-negotiation alternative score times the probability the proposal is rejected.

We take the expectation of this sum with respect to all of the deliberator utility functions. Formally, for any game G , let X be the set of all possible exchanges in G that are available to the allocator. Let T be the set of deliberator types. The computer model will choose exchange x s.t.

$$e = \operatorname{argmax}_{x \in X} \sum_{t \in T} P(t) \cdot (P(\text{accept}|x, t) \cdot PO_A + (1 - P(\text{accept}|x, t) \cdot NA_A))$$

Learning

The goal of the learning task is to complete the model of the deliberators by estimating the parameters from collected data. Since we are learning a mixture model, the task is two-fold; we must learn the distribution $P(T)$ over deliberator types, and for each type $t \in T$, we must learn the feature weights $\mathbf{w}^t = (w_{NN_A}^t, w_{NN_D}^t, w_{PO_A}^t, w_{PO_D}^t)$, corresponding to the contribution of each social preference. To solve this problem, we interleaved two optimization procedures, a version of the EM algorithm (A.P. Dempster 1977) and the gradient descent technique. We began by placing an arbitrary distribution over deliberator types and setting the feature weights with particular parameter values. We varied the number of types and the initial feature weights for each type. For details, see the Results section.

Each observation d consists of inputs $\mathbf{x}^d = (NN_A^d, NN_D^d, PO_A^d, PO_D^d)$, and response y^d , which equals 1 or 0 for “accept” or “reject”. The inputs are scaled to lie in the interval $[-1, 1]$, by setting -1 to be the smallest value of the feature and 1 to be the largest. This is done only to speed up learning.

The deliberator type t^d is unobserved, but we can compute the probability of each type for each data case using the current parameter settings:

$$P(t^d|d) = \frac{1}{Z^d} P(y^d|t^d, x^d) \cdot P(t^d)$$

where Z^d is a normalizing factor. Note that $P(y^d|t^d, x^d)$ is the likelihood of the deliberator’s response in the game at data point d , according to type t^d . This can be computed by plugging in the social utility function u_t for the observed game into the logistic function $\frac{1}{1+e^{-x}}$.

Computing the new values of the $P(t)$ parameters is straightforward. We simply compute $E(N_t|D) = \sum_{d \in D} P(t^d | d)$ and normalize. The maximization for the feature weights for each type is more interesting. Note that the model for each type is a simple sigmoid belief network (Neal 1992). (In fact, it is a particularly simple sigmoid

belief network, because it has no hidden layers.) The maximum likelihood problem for such a network can be solved by gradient descent, using a delta rule.

Since these networks participate in a mixture model, each one makes a contribution to the final output of $P(t|d)$. Therefore the gradient for the model associated with type t is proportional to $P(t|d)$. In other words, the degree to which a training example can be used to learn the weights in a network is proportional to the probability that the network actually generated the data. In the delta rules for these networks, therefore, we multiply the learning rate by $P(t|d)$. We obtain the following delta rule for each feature j :

$$w_j^t = w_j^t + \alpha P(t|d) \sum_{d \in D} x_j^d (1 - P(y^d|t)).$$

The $1 - P(y^d|t)$ error term comes from assuming that if the network was a perfect predictor, it would have predicted the outcome y^d with probability 1. The difference between 1 and $P(y^d|t)$ is the degree to which the network is not a perfect predictor.

Experimental Setup

A total of 42 subjects participated in the experiment, which was advertised as “an experiment in coordination”, offering monetary reward. 28 of the subjects were students, both undergraduates and graduates; 14 of the subjects were adults living in the area. Participants were given a 20 minute tutorial of the game, consisting of an explanation of the rules, the scoring function and a practice game.

Each subject was identified by a serial number, and was seated in front of the same terminal for the entire length of the experiment, consisting of a number of rounds of Colored Trails. A central server was responsible for matching up the participants at each round and for keeping the total score for each subject in all of the rounds of the experiment. No subject was paired up with any other subject more than once in the same role capacity. Subjects could not observe the terminals of other subjects, and were not informed about the identity of their partner. Participants were paid in a manner consistent with the scoring function in the game. For example, a score of 130 points gained in a round earned a \$1.30 payment. We kept a running score of each of the subjects, throughout the experiment.

Our experiment consisted of two separate phases: data-collection and evaluation. In the data-collection study, 16 subjects played consecutive CT games against each other. Each subject played 12 CT rounds, making for a total of 96 games played. The initial settings (board layout, tile distribution, goal and starting point positions) were different in each game. For each round of the game, we recorded the board and tile settings, as well as the proposal made by the allocator, and the response of the deliberator. We ran two instances of the data-collection phase, each one with different subjects, collecting 192 games. The data obtained from the data-collection phase was then fed into the learning model.

The evaluation study consisted of two sessions, each involving 5 human subjects and 3 computer players. The computer players, only playing allocators, were automatic agents

capable of mapping any CT game position to a proposed exchange, according to a specified strategy. Agent *SP* consisted of an allocator who proposed the exchange with the highest expected utility, according to our learned social preferences model. Agent *NE* consisted of an allocator who proposed the exchange corresponding to the Nash equilibrium strategy for the allocator. Agent *NB* consisted of an allocator who proposed the exchange corresponding to the Nash bargaining strategy for the allocator.

At each round, eight concurrent games of CT were played, in which members of the same group played each other. The set-up was as follows. One of the human subjects, designated as an allocator, played another human subject, designated as deliberator; each computer player, designated as an allocator, played another human subject, designated as deliberator. The game settings, including board layout, start and goal positions, and initial tile distributions, were the same for all of the games played by members of the same group. Therefore, at each round there were 4 matching CT games being played by the eight members of each group.

Participants were given the same instructions and tutorial as in the data-collection experiment. We feared that telling the participants that they will be playing computer agents in some of the rounds would alter their behavior. Therefore, participants were deceived, and led to assume that they were playing a human at each round. Approval was obtained from the Human Subjects Committee for this procedure.

As before, for each round of CT, we recorded the settings of each game, the proposals being offered by the allocators, and the deliberators' response. The first group played 7 games, and the second group played 14 games, for a total of 21 games, where each game was replicated 4 times with different allocators.

Results and discussion

We attempted to learn separate models for one, two and three possible types of deliberators, henceforth referred to as Model1, Model2 and Model3 respectively. For all models, we used random initial values for the distribution over deliberator types. For Model1 we also used random values for the feature weights. For Model2 and Model3, we assigned each deliberator type with initial feature values that corresponded to different points in feature space. We did this by "highlighting" some features and giving them significantly higher initial value than others. In Model2, one of the deliberator types highlighted inequality and fair trade, while the other highlighted aggregate utility. In Model3, deliberator types highlighted inequality, aggregate utility, and fair trade separately.

We ran each model on the data obtained in the data-collection phase. Model1, which had a single deliberator type, learned feature weights (7.00, 5.42, 0.40, 4) for features (individual utility, inequality, aggregate utility, fair trade). Model1 described a single "selfish" deliberator who only cared about its own outcome and favored advantageous inequality. In Model2, the distribution over deliberator types was (0.36, 0.63) and feature weights were (3.00, 4.61, 5.13, 0.46) and (3.13, 0.47, 4.95, 3.30) for each type respectively. Model2 described two partially altruistic

deliberators, since they both have high weights for aggregate utility, while one of the types cares about making more, and one of the types cares about fair trade. In Model3, the distribution over deliberator types assigned miniscule probability for the third type, and resembled Model2 in all other parameter values. We therefore decided to use Model2 in the validation study.

The following table presents the results of the evaluation phase, with respect to the number of accepted exchanges and overall utility for each of the models used in the experiment.

Model	Total Utility	Proposals Accepted	Proposals Declined	No Offers
SP	2880	16	5	0
NE	2100	13	8	0
NB	2400	14	2	5
HU	2440	16	1	4

For each allocator model, we list the total utility, the number of proposals accepted, the number of proposals rejected, and the number of times no offer was proposed. The computer allocator labeled "NE" always proposed the exchange that corresponded to the allocator's strategy in the (unique) sub-game perfect Nash equilibrium of each CT game. In essence, this resulted to offering the best exchange for the allocator, out of the set of all of the exchanges that are not worse off to the deliberator. As a consequence, many of the exchanges proposed by this agent were declined, since they were not judged as fair by the deliberator. This result closely follows the findings of behavioral game theory. The performance of NE was the worst of the four. The computer allocator labeled "NB" always proposed the exchange that corresponded to the allocator's strategy in the Nash Bargaining profile. In particular, this always corresponded to an exchange which was Pareto Optimal, i.e. the proposed outcome was not worse off than the no-negotiation alternative for both players. These offers were consistently better off for the deliberator, when the board and tile distribution enabled it, and were therefore accepted very often. Note that if a Pareto optimal offer did not exist, the Nash bargaining agent made no offer. Because NB tended to offer quite favorable deals to the deliberator, its utility suffered.

The allocator labeled "HU" aggregates the data corresponding to proposals that were made by human subjects. Human offers were almost always accepted, when they were made. The computer allocator that followed our expected utility model, labeled "SP", achieved a significantly higher utility than the other computer agents, and even did better than the human players. It also had the highest number of accepted proposals, along with the allocations proposed by humans. Interestingly, our model proposed the same offer as the human proposal in 4 of the games, whereas the Nash equilibrium player did not match a human proposal in any game, and the Nash bargaining player matched human proposals in 2 games. This suggests that our computational model would be perceived to be relatively more reasonable, or "human like", by other people. T-test comparisons between the mean utility of "SP" and "NE" yielded a p value of 0.02 for a confidence interval of 95%. T-test comparison between "SP" and "NE" and "SP" and "HU" yielded a p value of 0.02 for a confidence interval of 90%. We are

planning to run a larger evaluation experiment soon.

We conclude by describing some interesting behavior displayed by our program. The “NNA” heading in the following tables represents the no-negotiation alternative situation.

First, we show a round in which SP proposed an exchange which was accepted by an altruistic human deliberator. Interestingly, there was only one observation in which an altruistic deliberator agreed to such an exchange, yet it was used 4 times by the allocator in the evaluation phase. This behavior, of asking for a favor when the other player is much better off, seems reasonable.

Model	Allocator Score	Deliberator Score
NNA	45	170
SP	70	150

A second example, in which the proposed outcome of the exchange proposed by NE, while beneficial for the deliberator, was lower than the exchange proposed by SP. The NE exchange was rejected, while the SP exchange was accepted. This seems to indicate that the responders in this game cared about the equality of outcomes. Note that in this exchange, the SP exchange and the exchange proposed by the human were equal.

Model	Allocator Score	Deliberator Score
NNA	75	150
SP	170	170
NE	180	155
NB	150	190
HU	170	170

Finally, an example in which the offer made by SP was rejected, while the proposal made by NE was accepted. The expected utility for SP in the no negotiation alternative for the allocator is already very high. This results in the allocator offering advantageous exchanges to himself, which stand a low probability of acceptance.

Model	Allocator Score	Deliberator Score
NNA	180	35
SP	200	15
NE	190	35

Conclusion and Future Work

We have presented a computational framework for representing and learning about social preferences of people in one-shot games. Our model successfully learns social preferences of humans from data, and can generalize to people and games that were not seen before. In the future, we plan to model additional social preference features. One feature that may be particularly relevant is the regret that a player might feel from accepting one trade when a more preferable trade was available. A player who feels regret might reject a trade that she would have accepted otherwise. Unlike the four features that we use, regret cannot be defined in terms of the no-negotiation alternative and proposed-outcome pay-offs. It depends not only on the proposed outcome, but on the space of possible outcomes. Regret must be measured relative to a particular “default” deal in the space of possible deals. A natural choice is the Nash bargaining solution.

The CT framework provides a natural test-bed for learning in more complex scenarios, such as repeated games. In repeated games, additional factors are at work, including dynamics of reciprocity and punishment. We would also like to study games involving more than two allocators, where issues of competition between the allocators arise.

While we have focused on one particular game for practical reasons, the learned models we use are cast in terms of general social preferences and do not depend on the specific features of the game. Therefore the learned models should be immediately applicable to other games in which the same social preferences play a role. There is reason to hope that the model learned for one type of negotiation game will generalize to another type of game. It will be interesting to see whether it does indeed do so.

For our performance measure, we have used the score obtained by our computational agent while playing against humans. One can also imagine setting the goal of trying to play as much like humans as possible. Our learned models would now be used to generate computer play, rather than simply as predictors of human reaction. With the approach described in this paper, it is conceivable that we could build a program to pass a limited Turing test in a negotiation game.

References

- A.P. Dempster, N.M. Laird, D. R. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39(1).
- B.Grosz; S.Kraus; Talman, S.; and B.Stossel. 2004. The influence of social dependencies on decision-making. In *submission*.
- Bolton, G. 1991. A comparative model of bargaining. *American Economic Review* (81).
- Chajewska, U.; Koller, D.; and Ormoneit, D. 2001. Learning an agent’s utility function by observing behavior. *ICML*.
- Charness, G., and Rabin, M. 2002. Understanding social preferences with simple tests. *Quarterly Journal of Economics* (117).
- Davidson, A.; Billings, D.; Schaeffer, J.; and Szafron, D. 2000. Improved opponent modeling in poker. In *International Conference on Artificial Intelligence*.
- Fudenberg, D., and Levine, D. K. 1998. *The Theory of Learning in Games*. MIT Press.
- G. Werner, R. Schmittberger, B. S. 1982. An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior and Organization* (3).
- G.F. Loewenstein, M.H. Bazerman, L. T. 1989. Social utility and decision making in interpersonal contexts. *Journal of Personality and Social psychology* (57).
- J.Nash. 1971. The bargaining problem. *Econometrica* 18.
- Neal, R. 1992. Connectionist learning of belief networks. *Artificial Intelligence* 56.
- Ng, A., and Russell, S. 2000. Algorithms for inverse reinforcement learning. *ICML*.