Learning to Reveal Information in Repeated Human-Computer Negotiation

Noam Peled[1], Kobi Gal[2], and Sarit Kraus[1,3]

[1] Bar Ilan University, Israel
[2] Ben-Gurion University, Israel
[3] Institute for Advanced Computer Studies. University of Maryland, MD 20742

Abstract. Many negotiations in the real world are characterized by incomplete information, and participants' success depends on their ability to reveal information in a way that facilitates agreement without compromising the individual gains of agents. This paper presents a novel agent design for repeated negotiation in incomplete information settings that learns to reveal information strategically during the negotiation process. The agent used classical machine learning techniques to predict how people make and respond to offers during the negotiation, how they reveal information and their response to potential revelation actions by the agent. The agent was evaluated empirically in an extensive empirical study spanning hundreds of human subjects using the Colored Trails test-bed. Results show that the agent was able to outperform people as well as an alternative agent that was tailored by experts to play the game. In particular, the agent learned to exploit the fact that revealing its resources during the game in an incremental fashion makes people more willing to make beneficial proposals to the agent. This work demonstrates the efficacy of combining machine learning with opponent modeling techniques towards the design of computer agents for negotiating with people in settings of incomplete information.

1 Introduction

In many negotiation settings, participants lack information about each other's preferences, often hindering their ability to reach beneficial agreements. In such cases, people can choose whether and how much information to reveal about their preferences to others. This paper presents a novel agent design for repeated negotiation with people in settings in which participants can choose to reveal their private information while engaging in a finite sequence of alternating negotiation rounds.

People commonly choose to reveal information to others in an incremental fashion when they negotiate. For example, consider a scenario in which two diplomats representing their respective countries are negotiating a peace treaty. The peace treaty is comprised of several agreements spanning economical, diplomatic and military issues. Revealing private information can facilitate the negotiation. For example, diplomat A may disclose that her country is suffering from an acute drought in order to justify a request for additional access to water resources under the jurisdiction of country B. Diplomat A may also have information regarding a recent discovery of gas resources by her country. However, she may choose to refrain from revealing this information until securing the water rights so diplomat B will not ask to claim this resource.

Our study is conducted in an experimental framework in which people and agents repeatedly negotiate over scarce resources, there is incomplete information about their preferences and they are given the opportunity to reveal these preferences in a controlled fashion during the negotiation. Constructing effective agent strategies for such settings is challenging. On the one hand, behavioral economics work has shown that people often follow equilibrium strategies [2] when deciding whether to reveal private information to others. On the other hand, people's bargaining behavior does not adhere to equilibrium [6, 15], and computers cannot use such strategies to negotiate well with people [13].

This paper describes a new agent design that uses a decision-theoretic approach to negotiate proficiently with people in repeated revelation games. The agent explicitly reasons about the social factors that affect people's decisions whether to reveal private information, as well as the effects of people's revelation decisions on their negotiation behavior. It combines a prediction model of people's behavior in the game with a decision-theoretic approach to make optimal decisions. The parameters of this model were estimated from data consisting of human play. The agent was evaluated in an extensive empirical study that spanned hundreds of subjects. The results showed that the agent was able to outperform human players as well as an agent using an alternative strategy that was designed by experts to play the game. It learned to make offers that were significantly more beneficial to people than the offers made by other people while not compromising its own benefit, and increased the social welfare of both participants as compared to people and the expert-designed agent. In particular, it learned to reveal information to people in a way that increased its expected performance according to its model.

The contributions of this paper are threefold. It presents a formal model of how people reveal private information in repeated negotiation settings. Second, it shows how to incorporate this model into a decision-making paradigm for an agent design that is able to incrementally reveal information. Lastly, it demonstrates the efficacy of this model empirically, showing that the agent is able to outperform people as well as the expert-designed agent.

2 Related work

Our work is related to studies in AI that use opponent modeling to build agents for repeated negotiation in heterogeneous human-computer settings. These include the KBAgent that made offers with multiple attributes in settings which supported opting out options, and partial agreements [17]. It used density estimation to model people's behavior [4] and approximated people's reasoning by assuming that people would accept offers from computers that are similar to offers they make to each other.

Other works employed Bayesian techniques [10] or approximation heuristics [11] to estimate people's preferences in negotiation and integrated this model with a pre-defined concession strategy to make offers. Bench-Capon [3] provide an argumentation based mechanism for explaining human behavior in the ultimatum game. We extend these works in two ways, first in developing a partially strategic model of people's negotiation behavior and second in formalizing an optimal decision-making paradigm for agents using this model. Gal and Pfeffer [9] proposed a model of human reciprocity in a setting consisting of multiple one-shot take-it-or-leave-it games, but did not evaluate a computer agent or show how the model can be used to make decisions in the game. Our work augments these studies in allowing players to reveal private information and in explicitly modeling the effect of revelation on people's negotiation behavior.

Work in interest-based negotiation has studied different protocols that allows players to reveal their goals in negotiation in a controlled fashion [7, 18, 8]. Our work complements these studies in deriving a learning model for the revelation decision. Lastly, our results align with past studies showing that social preferences such as fairness and competitiveness affect people's strategies in negotiation studies. It shows that in revelation games, this tendency overbears cooperation when interacting with computers.

Peled et al. [20] presented an agent for making all-or-nothing revelation decisions in negotiation settings by combining decision theoretic models with classical machine learning algorithms. We extend this work to more complicated settings in which participants' success depends on how much information they reveal at each stage of the game. Furthermore, Peled et al did not allow for repeated revelation.

Lastly, our empirical setting, which incorporates both signaling and bargaining, was inspired by canonical studies showing that people learn to play equilibrium strategies when they need to signal their private information to others [2]. On the other hand, people's bargaining behavior does not adhere to equilibrium [6, 15], and computers cannot use such strategies to negotiate well with people [13]. Our work shows that integrating opponent modeling and density estimation techniques is an effective approach for creating agents that can outperform people as well equilibrium strategies in revelation games.

3 Repeated Revelation Games

We designed a game in which players need to negotiate over resources in an incomplete information setting and make repeated decisions about whether to reveal salient information to their partners. This "repeated revelation game" is played on a board of colored squares. One square on the board is designated as

[4] following a method suggested by Coehoorn and Jennings for modeling computational agents [5]

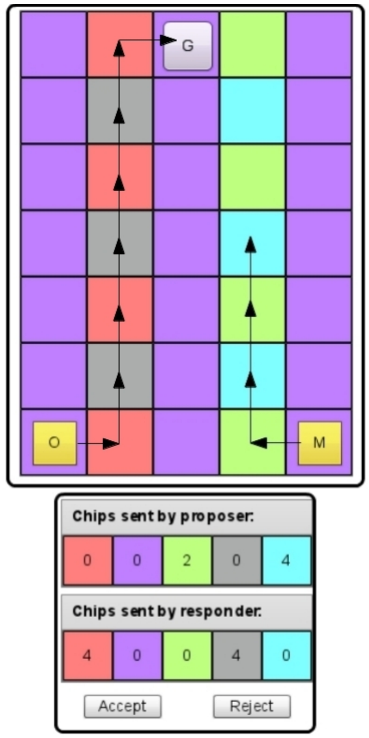

Fig. 1. A snapshot showing the revelation game from the point of view of a person (the "M" player) playing against a computer agent (the "O" player). Both players have only one possible path to the goal. Their closest position to the goal with their current chip sets is outlined on the board. The bottom panel shows an example of a proposal given by the "O" player.

the players' goal. The goal of the game is to reach the goal square. To move to an adjacent square requires surrendering a chip in the color of that square. Each player starts the game with a set of 16 chips. The allocation of the chips was chosen such that no player can reach the goal using only his chips, but there are some chip exchanges that let both players reach the goal. Players have full view of the board, but cannot observe the other player's chips. An example of a CT revelation game is shown in Figure 1. The gray icon "G" represent the goal.

Each round in our CT game progresses in three phases with associated time limits. In the first "revelation" phase, both players can choose to reveal a subset of their chips. This decision is performed simultaneously by the players, and the chips are only revealed at the end of the phase[5]. Figure 2 shows a snapshot of a revelation panel from the point of view of the "M" player, who has revealed a

[5] The revelation decision is truthful, that is, players cannot reveal chips that are not in their possession.

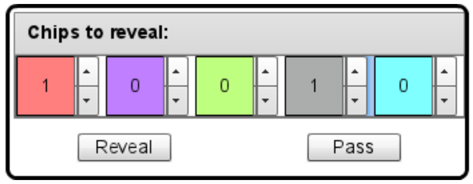

Fig. 2. A snapshot showing an example for revelation decision from the point of view of the "M" player, who has revealed a red and gray chip to the other player.

red and gray chip to the other player. Players communicate using an alternating offers protocol. In the "proposal phase", one of the players can offer to exchange a (possibly empty) subset of its chips with a (possibly empty) subset of the chips of the other player. The proposer cannot include chips that it did not reveal in the proposal, but can ask for any set of chips from the other player. The game server does not allow the responder to accept proposals that require it to give more chips than it has. Following an accepted proposal, the chips are transferred automatically. If the responder rejects the proposal (or no offer was received following a three minute deadline), it will be able to make a counter-proposal. In the "movement phase", the players can move towards the goal using the chips they have. The game ends after both players reach the goal, or after 5 rounds. At the end of the game, both players are moved towards the goal according to their chips, and their score is computed as follows: 60 points bonus for reaching the goal; 5 points for each chip left in a player's possession and 10 points deducted for any square in the path between the players' final position and the goal-square. This path is computed by the Manhattan distance. Suppose, as the example in Figure 1, player "O" offers to send player "M" 2 green and 4 cyan chips, and in return he asks for 4 red and 4 gray chips. Without using the chips the players already have, accepting this proposal will move the players as the black arrows point out: Player "O" will be able to reach the goal surrendering 8 chips, while player "M" will be able to move only until the cyan square, 4 squares from the goal, by surrendering only 4 chips. Let's assume this is the first accepted proposal in the game, so player "O" gains 2 chips from the transfer and will remain with $16+2-8 = 10$ chips, while player "M" looses 2 chips from the transfer and will also remain with $16-2-4 = 10$ chips. The score of player "O" from this proposal is $60 + 10 * 5 = 110$, and the score for "M" is $10 * 5 - 4 * 10 = 10$. On the other hand, if the game ends without an agreement, both players will be moved one square toward the goal (player O lacks gray chips and player M lacks cyan chips) and get $15 * 5 - 7 * 10 = 5$ points. The player's only motivation in the game is to maximize its own points, without caring about the other participant's score. There are several advantages towards using the revelation game described above to study people's revelation strategies. First, the game is challenging to play. Players have limited information about each other's resources. There is a tradeoff between revealing information to the other party to facilitate the negotiation, and the other party exploiting this information. Second, it offers a rich space of revelation strategies for players, who can choose never to reveal information to each other, or reveal information incrementally. Lastly, it combines revelation strategies with negotiation in a natural way.

CT provides a realistic analog to task settings, highlighting the interactions among goals, tasks required to achieve these goals, resources needed for completing tasks, and the revelation of these resources. Chips correspond to agent capabilities and skills required to fulfill tasks. Different squares on the board represent dierent types of tasks. A player's possession of a chip of a certain color corresponds to having the skill available for use at a time. Traversing a path through the board corresponds to performing a complex task whose constituents are the individual tasks represented by the colors of each square.

4 The MERN Agent

The Maximum Expectation Revelation and Negotiation (MERN) agent developed for this study uses a decision-theoretic approach to negotiate in revelation games. It is based on a model of how humans make decisions in the game. The agent uses a decision theoretic approach towards generating its actions in the game. To maximize its expected score, the agent grows and traverses a game tree to a constrained depth, and uses backward induction to choose the strategy that is associated with maximal score. The constrained depth can be set according to the avaiable computational power[6]. There are two types of nodes in the tree. Decision nodes represents actions for MERN, such as deciding how many chips to reveal, what proposals to make, and how to respond to offers. Chance nodes in the tree represent people's actions, which are modeled probabilistically and described in the next section. The state of the game includes salient information that MERN uses to make decisions in the game. For each round, the state includes the chips in its possession, its position on the board, and the history for all previous rounds.

If MERN reaches a node associated with an terminal state of the game, it computes its score based on the scoring function of the game given this state. If MERN reaches the constrained depth of the tree and generates a leaf that is not associated with a terminal state of the game it estimates the final score using the prediction model described in the next section. Otherwise, MERN proceeds as follows. If the node represents a proposal decision, it makes the offer that maximizes its expected score over two events. If the person accepts, then MERN updates its state to include the realization of the offer, and it grows the tree one ply to compute its expected score in the next round. If the person rejects, then MERN predicts the possible counter proposals the person can make, and computes its expected score from accepting and rejecting the offers.

[6] In this study we used constrained depth of 0: The agent computes all the decision nodes in the current round, and than for each node predicts its final score in the game. The only exception is the first round, where the agent can calculates the decision nodes offline and uses a constrained depth of 1.

If the node represents a response decision, then MERN accepts the offer if it maximizes its expected score over the following two events. If MERN accepts, then it updates its state to include the realization of the offer, and it grows the tree one ply to compute its expected score in the next round. If MERN rejects then MERN grows the tree one ply and predicts the expected score from making a proposal in the next round.

If the node represents a revelation decision, then MERN computes its expected score from revealing every possible chip sub-set from the set of chips in each possession. This computation depends on its model of how people reveal their own chip set and on its expected score from making its decision in the next round after growing the tree one ply.

The branching factor of the tree is very large. For example, the set of all the human player's possible chips revelations is exponential in the size of his chips set and the set of all possible offers the human player can make is exponential in the size of the agent and the human player chip sets. MERN addresses this challenge as follows. First, when reasoning about people's decisions, it limits the branching factor by only considering actions that are most likely according to its model (associated with a probability that is higher than a constant threshold). Second, when reasoning about what proposals to make, it constrains the set of possible chip sets to those it revealed, as required by the rules of the game. When reasoning about revelations, it uses a transformation function to constrain the chip set which is described in the next section.

5 Modeling Human Players

In this section we describe how we model people's behavior in the game for the following tasks: revealing chips, making and responding to offers. To this end we assume that there is a training set of games played by people. We used four types of classifiers: Support vector machine with linear, polynomial and radial basis kernels, as well as logistic regression. We trained each of the classifiers on a test set and tested on a held out test set using ten-fold cross validation. For binary classifiers, the accuracy was measured using the AUC method [1] for each fold. For multi-class classifiers we measured the accuracy for each class using AUC (5 folds) and then calculated the mean. All classifiers used the following set of features that provided the optimal performance, deemed the "common feature set".

- The current phase in the game (whether proposal or counter proposal).
- The current round in the game.

The following features are computed for both players.

- The number of chips needed to get from the current position of the player to the goal.
- The difference between the chips in the possession of a player and the chips it is given at the onset of the game.

– The number of accepted and rejected proposals of both players.
– The number of chips that were revealed by a player that it needs to get to the goal. Because players cannot observe each other's chips, we estimate the set of needed chips for a player as the chips comprising the shortest path from the current location of the player to the goal. In this way we include features for the chips that each player needs, and does not need, to get to the goal.

In addition to the common features described above, we included additional features which were added to each classifier for the prediction tasks we describe below. Table 1 summarizes for each task its predictor and the accuracy.

Model	Predictor	Accuracy
Accepting proposals	SVM (linear kernel)	71%
Proposals	multi-class logistic regression	68%
Revelations	multi-class logistic regression	72%
Reaching the goal	logistic regression	82%

Table 1. Predictors and accuracy

5.1 Accepting Proposals

For predicting whether a chosen proposal is accepted by the responder, we used an SVM classifier with a linear kernel found to be the most accurate (71%). In addition to the common features set, this classifiers also use the following features:

– The number of chips the responder needs and doesn't need to get to the goal (and similarly for the proposer).
– The sum of the above features from all prior proposals made to the responder.

We used Platt's probabilistic output method for SVM [14] to approximate the probability that a responder accepts a given offer[7].

5.2 Revelations and Proposals

We found the logistic regression model [16] to be most accurate for predicting revelations (72% accuracy) and proposals (68% accuracy) in the game. The number of different revelations and proposals is exponential in the number of chips, and the size of the state space becomes unwieldy. To represent the state space of possible chips more efficiently we created a mapping from proposals and revelations to the real numbers. Each proposal is defined as a tuple of 4 values:

[7] For calculation we used the packages LIBSVM [4] and Scikit-learn: Machine Learning in Python[19]

the number of chips the responder needs to get to the goal (λ_1^ω), doesn't need to get to the goal (λ_2^ω), the number of chips the proposer needs to get to the goal (λ_3^ω) and doesn't need (λ_4^ω). Let T_ω define a transformation function as follows: $T_\omega(\omega) = \lambda_1^\omega \cdot 10^3 + \lambda_2^\omega \cdot 10^2 + \lambda_3^\omega \cdot 10 + \lambda_4^\omega$. Similarly, each revelation is defined as a tuple of 3 values: number of chips the revealer needs (λ_1^ψ), number of chips the other player needs (λ_2^ψ), and number of chips no one needs (λ_3^ψ). According to this transformation we have a one-to-one mapping from chip sets to the real numbers (given that the feature values are between 0 and 9, which is always the case in our setting).

5.3 Reaching the Goal

A logistic regression classifier was found most accurate (82%) in predicting whether the agent would reach the goal at the end of the game. In addition to the common feature set, we also used the agent's current score in the game - the score it got from the last accepted proposal, and if no proposals was accepted yet, it's the non-negotiation score, the score the players get if no proposals was accepted.

5.4 Final Score in the Game

We used linear regression to predict a player's final score in the game. In addition to the common feature set we added the current score of the agent. The regression correlation for predicting a player's score given the common feature set was 0.36. We hypothesized that training a separate classifier dependent upon whether the player got to the goal will improve performance because scores in the game are highly dependent upon whether the player reaches the goal. Confirming this hypothesis, the regression correlation rose to 0.52 and 0.613 when separating the data test into these two cases. Therefore, we compute an expectation over a player's final score in the game given the probability that the player reaches the goal, as computed by the previous classifier.

6 Empirical Methodology

We recruited 121 subjects using Amazon Mechanical Turk[12]. Subjects received an identical tutorial on repeated revelation games that was exemplified on a board (not the board used in the study)[8]. Actual participation was contingent on successfully answering a set of basic comprehension questions about the game. Each participant played only one game. The board in the study fulfilled the following conditions at the onset of the game: (1) Every player lacks some of the chips needed to reach the goal; (2) Every player possesses the chips that the other needs to get to the goal; (3) There exists a set of chips exchange which allows both players to reach the goal. A central server (randomly) matched each participant

[8] The tutorial and the game can be found here: http://tinyurl.com/7w2cjxq

with a human or an agent counterpart for each game. The participants didn't know whether they are playing against a human player or an agent. The identity of each participant was not disclosed. Participants were paid according to their performance in the game.

7 Results and Discussion

In this section we demonstrate the efficacy of the MERN agent by comparing its performance to new people, as well as to an agent using an alternative negotiation strategy that was designed by experts. For each result, we list the mean, standard deviation and number of observations. All results reported in this section are statistically significant in the $p < 0.05$ range[9].

We first present a comparison of the average performance of MERN and people (when interacting with other people). As shown in Figure 3, the average score of MERN (101.88 ± 25.45, $n = 24$) was significantly higher than people (63.97 ± 47.97, $n = 58$). In addition, MERN was able to get to the goal (96%) significantly more often than people (60%). Lastly, MERN also had a positive influence on the individual performance and social welfare of both participants: The performance of people interacting with MERN (85.83 ± 41.45) was significantly higher than the performance of people interacting with other people (63.97 ± 47.97). The aggregate utility for both players was higher when interacting with MERN (187.71 ± 28.61) than with other people (127.17 ± 75.4).

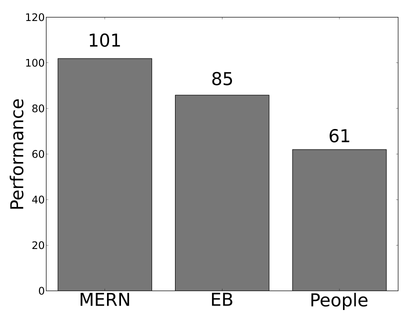

Fig. 3. Average Performance of the MERN agent, the EB agent and people.

7.1 Strategy Analysis

To explain how MERN was able to outperform people, we provide the following detailed analysis of its negotiation strategy. We first observed that people were

[9] To analyze the significance we've used T-test.

significantly more generous to MERN than to other people: the offers made by people were significantly more beneficial to MERN (61.96 ± 54.28, $n = 80$) than to other people (33.54 ± 44.45, $n = 227$). Further analysis of people's behavior supports this conjecture, in that there was a high positive correlation ($R^2 = 0.56$) between the potential score to people made by MERN and the offers made by people to MERN - as MERN offers higher score, the human player tends to be more generous in his offers to MERN. This correlation explains the generous offers MERN got from human player. Given that people made generous offers to MERN, it was surprising to discover that people accepted less offers when interacting with the MERN agent (19%) than when interacting with people (36%). To explain this, we need to analyze its negotiation strategy more closely. To begin with, MERN makes offers that are significantly more competitive than people. The difference in the proposed score to MERN and to people from offers made by MERN was 33 points, whereas this difference was just 12 points from offers made by people. This discrepancy highlights the negotiation strategy used by MERN in the game. MERN learned to make offers that are significantly more beneficial to itself and to the other player, than the offers made by people, however the offers were significantly more beneficial to MERN than to people. For example, suppose MERN is the "O" player and is the first proposer in the board shown in Figure 1. The proposal in the example in section 3 is an example of one of MERN's possible proposals as first proposer in the game[10]. We conjectured that this competitiveness is the reason for people's lower acceptance rates when playing with MERN. The associated utility to MERN is sufficiently high so that it is still able to outperform people given the lower acceptance rates. Another reason for MERN's success was its ability to adapt to individual participants during the game. When its negotiation partner rejects its offers in earlier rounds, it learns to make more beneficial offers in later rounds. For example, in a game in which a human participant rejected all of its past offers, MERN made a proposal that allowed both players to reach the goal while offering the other participant 4 more chips than to itself. This offer was associated with a 79% acceptance probability and an expected score of 65 points for MERN. In contrast, offering a more competitive deal in which both players get to the goal, but giving MERN two more chips than the other participant, is associated with an acceptance probability of 38% and an expected score of 46 points for MERN.

The fact that people's behavior in strategic negotiation is known to be noisy affected the prediction accuracy of our models. Here we show that in practice, our MERN agent was able to make good decisions in practice by combining these predictors with a decision theoretic model.

7.2 Analysis of Revelation Strategies

We've searched our learning games set for correlations between different types of chips revelation, and human behavior that the agent can exploit, such as

[10] If MERN is the first proposer in the game, its proposal depends only on the human player's revelation

generosity and the willingness to accept offers. We found a strong correlation suggesting that as a player reveals more chips that the other player needs, both the player ($R^2 = 0.67$) and the other player ($R^2 = 0.53$) become more generous in their offers. In addition, a player becomes less likely to accept proposals when it reveals more chips that the other player needs ($R^2 = 0.47$), and when the other player reveals chips that the first player needs ($R^2 = 0.51$). As a result, MERN tends to reveal more chips the human player needs, to exploit its generosity. Moreover, MERN takes into consideration how many chips the human player reveals that MERN needs: if it doesn't reveal many chips, the human player will be more willing to accept proposals, and it affects MERN's calculation of its expected score from proposals, such as asking for more chips.

We were interested in what manner human players revealed chips in the game. As described in Table 2, we calculated the histogram of the number of times human players have made a revelation act in the game. We found out that most people revealed their chips incrementally. The average number of chip revelations in a game was similar for both MERN (3.62 ± 1.35) and for people (3.8 ± 1.42).

	0	1	2	3	4	5
People	4%	21%	24%	17%	21%	13%
MERN	0%	17%	21%	58%	4%	0%

Table 2. Revelations histogram of people and MERN

7.3 Comparison against an Expert Design Agent

To demonstrate the performance of MERN, we compared its performance to an alternative agent that used hand-crafted rules of behavior. This agent used a predefined strategy designed by experts and represents an upper bound on performance in our game. We hypothesized that the Expert Based (EB) agent would be able to outperform people, and that the MERN agent would be able to play at least as well as the EB agent[11]. As shown in Figure 3, the results confirm the hypothesis, in that the EB agent obtained a significantly higher score (94.23 ± 37.95, $n = 39$) than people. In addition, there was no significant difference between the average score obtained by MERN and the EB agent. Interestingly, the EB agent was able to get to the goal significantly more times than people (79%), but significantly less than MERN. Second, the potential score offered by people to EB (49.2 ± 47.34, $n = 126$) was significantly higher

[11] In general, EB asks for 3 chips it needs and propose 2 chips that on the other player path to the goal. It accepts any beneficial proposal, as long it receives at least one more chip than it gives. It reveals no more than two chips, to be able to propose them. If only EB or the other participant has reached the goal, EB's strategy changes - EB turns to be more generous.

than proposed to other people, but significantly lower than proposed to MERN. Third, the social welfare in EB's games (157.05 ± 60.89) was significantly higher than the social welfare in people against other people games, but significantly lower than in MERN games. Lastly, the score obtained by people interacting with the EB agent was significantly lower (62.82 ± 40.25) than the score obtained by people playing with MERN.

8 Conclusion

This paper presented an agent-design for repeated negotiation with people in incomplete information settings where participants can choose to reveal private information at various points during the negotiation process. The agent used opponent modeling techniques to predict people's behavior during the game, based on a set of features that included players' social factors as well as game-dependent information and the effects of people's revelation decisions on their negotiation behavior. The parameters of the model were estimated from data consisting of people's interactions with other people and with an expert designed agent. In empirical investigations, the agent was able to outperform people playing other people and the expert designed agent. This work is a first step towards a general argumentation system in which agents integrate explanations and justifications within their negotiation process.

9 Acknowledgments

We thank the anonymous reviewers for their helpful remarks. This work is supported in part by the following grants: ERC grant #267523, MURI grant number W911NF-08-1-0144, ARO grants W911NF0910206 and W911NF1110344 and MOST #3-6797.

References

1. B. Andrew P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997.
2. J. Banks, C. F. Camerer, and D. Porter. Experimental tests of nash refinements in signaling games. *Games and Economic Behavior*, 6:1–31, 1994.
3. T. Bench-Capon, K. Atkinson, and P. McBurney. Altruism and agents: an argumentation based approach to designing agent decision mechanisms.
4. C. Chang and C. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
5. R. M. Coehoorn and N. R. Jennings. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of EC*, 2004.
6. I. Erev and A. E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, 88(4):848–881, 1998.

7. S. G. Ficici and A. Pfeffer. Simultaneously modeling humans' preferences and their beliefs about others' preferences. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 323–330, 2008.
8. Y. Gal, S. D'souza, P. Pasquier, I. Rahwan, and S. Abdallah. The effects of goal revelation on computer-mediated negotiation. In *CogSci*, 2009.
9. Y. Gal and A. Pfeffer. Predicting people's bidding behavior in negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 370–376, 2006.
10. K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proc. of AAMAS*, pages 331–338, 2008.
11. C. M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
12. A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. page 453. ACM Press, 2008.
13. R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Communications of the ACM*, 53(1):78–88, 2010.
14. C. Lin Hsuan-Tien and R. C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, Aug. 2007.
15. R. D. McKelvey and T. R. Palfrey. An experimental study of the centipede game. *Econometrica: Journal of the Econometric Society*, 60(4):803–836, 1992.
16. S. W. Menard. *Applied logistic regression analysis.* SAGE, 2002.
17. Y. Oshrat, R. Lin, and S. Kraus. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proc. of AAMAS*, 2009.
18. P. Pasquier, R. Hollands, F. Dignum, I. Rahwan, and L. Sonenberg. An empirical study of interest-based negotiation. In *Proc. of ICEC*, 2007.
19. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. *scikit-learn: Machine Learning in Python'.* 2010.
20. N. Peled, S. Kraus, and K. Gal. A study of computational and human strategies in revelation games. In *Proc. of AAMAS*, 2011.