An Agent Design for Repeated Negotiation and Information Revelation with People

Noam Peled
Gonda Brain Research Center
Bar Ilan University, Israel

Ya'akov (Kobi) Gal
Faculty of Engineering
Ben-Gurion University, Israel

Sarit Kraus *
Dept. of Computer Science
Bar Ilan University, Israel

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. Results show that the agent was able to outperform people. In particular, it learned (1) to make offers that were beneficial to people while not compromising its own benefit; (2) to incrementally reveal information to people in a way that increased its expected performance. The approach generalizes to new settings without the need to acquire additional data. 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.

Introduction

In many negotiation settings, participants lack information about each other's resources and preferences, often hindering their ability to reach beneficial agreements (Sarne and Kraus 2003; Sarne and Grosz 2007). In such cases, participants can choose whether and how much information to reveal about their resources to others. This paper presents a novel agent design for repeated negotiation with people in settings where participants can choose to reveal information while engaging in a finite sequences of alternating negotiation rounds. For example, consider two agents representing airlines that negotiate over a codeshare agreement for sharing seating allocations on their flights. This process comprises separate agreements about how many seats to allocate in different flights. For each of these agreements, revealing private information can improve the outcomes of participants. One of the airlines may disclose that a flight is sparsely booked, which can help the other airline by accommodating their over-booking fares for the same destinations. However, revealing information may be costly, in that the other airline can demand a lower price for the sparsely booked seats.

Our study is conducted in an experimental framework called a "revelation game" in which people and agents repeatedly negotiate over scarce resources, there is incomplete information about their resources and preferences and they are given the opportunity to reveal this information in a controlled fashion during the negotiation. Although revelation games are akin to many real world negotiation scenarios, constructing effective agent strategies for such settings is challenging (Peled, Kraus, and Gal 2011).

The proposed agent design 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. In particular, it learned (1) 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; (2) to incrementally reveal information to people in a way that increased its expected performance. Moreover, the agent had a positive effect on people's strategy, in that people playing the agent performed significantly higher than people playing other people. Lastly, we show how to generalize the agent-design to different settings that varied rules and situational parameters of the game without the need to accumulate new data.

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, including its ability to generalize to new settings.

Our work is related to studies in AI that use opponent

*Sarit Kraus is also affiliated with the University of Maryland Institute for Advanced Computer Studies.

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 (Oshrat, Lin, and Kraus 2009). It used density estimation to model people's behavior (following the method proposed by Coehoorn and Jennings for modeling computational agents (2004)) 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 (Hindriks and Tykhonov 2008) or approximation heuristics (Jonker, Robu, and Treur 2007) to estimate people's preferences in negotiation and integrated this model with a pre-defined concession strategy to make offers. Bench-Capon (2009) provide an argumentation based mechanism for explaining human behavior in the ultimatum game. We extend these works by considering more challenging settings with incomplete information and large strategy spaces, as well as explicitly modeling the effect of revelation of information on people's negotiation behavior in the agent-design.

Peled et al. (2011) present an agent-design that makes revelation decisions in negotiation settings that include people and computer agents. Our work differs in several ways. First, their model only supports all-or-nothing revelation decisions, while our model allows agents to reveal their resources incrementally, more like the real world. We show later that this was a key feature of our agent's strategy. Second, their setting was significantly more constrained than ours, consisting of a single negotiation and revelation round. Our game allowed several consecutive rounds of negotiation and revelation, making for an exponentially larger state space. Third, they tested the model on the same games for which they collected data, while our approach is shown to generalize to new settings (for which prior data was not available). Lastly, they used pre-defined rules (tit-for-tat) to predict people's behavior, while we used a principled machine learning approach.

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" was implemented using Colored Trails (CT), an established test-bed for studying task-settings that involve people and computers making decisions together (Gal et al. 2010; Haim et al. 2012; Gal et al. 2011). The game is played on a board of colored squares. One square on the board is designated as 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.

Each round in our CT game progresses in three phases

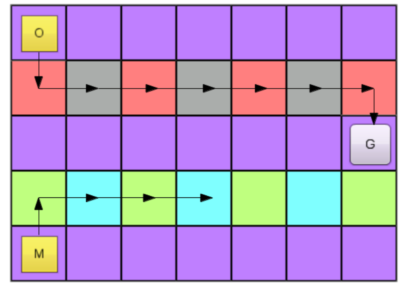

Figure 1: A snapshot showing the revelation game from the point of view of a person (the "M" player) playing against a computer agent (the "O" player).

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.[1] 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 responder cannot 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. In the next round the players' roles are switched: the first proposer in the previous round becomes the responder for the first proposal. The game ends after both players reach the goal, or after 5 rounds.[2] 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. Note that players' motivation in the game is to maximize their score, not to beat the other participants.

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 to reveal information incrementally, or reveal all their information on the onset of them game. Lastly, it is akin to the real world in that it combines revelation with negotiation. For example, the airline code-

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

[2]This was not arbitrary. The majority of games played by people ended before the 5-round deadline.

sharing agreement from the introduction can be modeled as a revelation game.

The MERN Agent

The agent designed for the study, termed MERN (the Maximal Expectation-based Revelation and Negotiation agent) developed uses a decision-theoretic approach to negotiate in revelation games. It is based on a model of how humans make decisions in the game. We begin by making the following definitions. The pair $\psi^n = \langle \psi_a^n, \psi_p^n \rangle$ contains the revelation decision ψ_a^n for the agent and ψ_p^n for the person at round n. Let $C^n = \langle C_a^n, C_p^n \rangle$ represent the set of chips in the possession of MERN and the person at round n. A proposal $\omega^n = \langle c_a^n, c_p^n \rangle$ includes chips $c_a^n \subseteq C_a^n$ sent by the agent and chips $c_p^n \subseteq C_p^n$ sent by the person at round n. A player can only propose to send chips that it already revealed. For the agent, this means that $c_a^n \subseteq \bigcup_{i=1}^n \psi_a^i$ (and similarly for the person). The set $\Omega_a^n(h^n)$ includes all possible proposals for the agent in round n (and similarly for the person). Let h^n denote the history of the game up to round n, where h^0 contains the board layout and the players location on the board on the onset of the game.

MERN makes decisions in the game by using Expectimax search. Due to the large action space and the exponential increase in the size of the tree, spanning the entire game is not feasible. Thus, a key challenge to designing strategies for MERN is how to assign values to intermediate states in the game. To address this challenge MERN uses a heuristic value function to assign utilities to intermediate rounds of the game. The value function is an estimate of the score that MERN will receive at the end of the game. Specifically, for any non-terminal round k in the game, the function $f_a(h^k)$ returns the estimate of MERN's score in the game.[3]

The results of this process are as follows. If round n is the end of the game, then the utility for MERN is simply its score in the game as described in the previous section. For intermediate nodes ($n = k$), the utility for any decision by MERN is computed using the estimator $f_a(h^n)$. We do assume the existence of stochastic models for predicting people's behavior at a given round n in the game. Specifically, there exist probability distributions of how people reveal chips, respond to proposals from MERN, and make proposals. We will show how we derive these models from data in the next section.

Suppose that MERN is making the decision to reveal chips in round n. The expected utility to MERN from revealing ψ_a^n chips, denoted $EU_a\left(\psi_a^n \mid h^{n-1}\right)$ depends on its model $p\left(\psi_p^n \mid h^{n-1}\right)$ of how people reveal chips in round n.

$$EU_a\left(\psi_a^n \mid h^{n-1}\right) = \sum_{\psi_p^n \in \Psi_p^n} p\left(\psi_p^n \mid h^{n-1}\right) \cdot \max_{\omega_a^n \in \Omega_a^n} EU_a\left(\omega_a^n \mid \psi^n, h^{n-1}\right) \quad (1)$$

where the pair $\psi^n = \langle \psi_a^n, \psi_p^n \rangle$ contains the revelation decision for both players and $EU_a\left(\omega_a^n \mid \psi^n, h^{n-1}\right)$ is the expected utility for the agent for making the proposal ω_a^n given the revelation decisions and the history. The set $\Psi_a^n(h^n)$ includes all the chips that MERN can choose to reveal at round n. This set includes all possible subsets of the intersection between the chips in the possession of MERN in round n and the chips it already revealed until this round. (we omit the history h^n when it is clear from context). Similarly, Ψ_p^n is the set of all the chips that the person can choose to reveal at round n.

Suppose that MERN is making the proposal ω_a^n at round n. Its utility from the proposal depends on its model $p\left(r_p^n \mid \psi^n, \omega_a^n, h^{n-1}\right)$ of how people respond to proposals in the game. If ω_a^n is accepted, then the game proceeds to round $n+1$, in which MERN will make the decision ψ_a^{n+1} of how many chips to reveal. If ω_a^n is rejected, then the person will make a counter-proposal in round n. In this case, MERN's utility depends on its model $p\left(\omega_p^n \mid \psi^n, \omega_a^n, r_p^n = F, h^{n-1}\right)$ of how people make proposals in the game, and choosing the best response to the proposal ω_p^n. Formally, we write

$$\begin{aligned} EU_a\left(\omega_a^n \mid \psi^n, h^{n-1}\right) &= p\left(r_p^n = T \mid \psi^n, \omega_a^n, h^{n-1}\right) \cdot \\ &\max_{\psi_a^{n+1} \in \Psi_a^{n+1}} EU_a\left(\psi_a^{n+1} \mid h^n\right) + p\left(r_p^n = F \mid \psi^n, \omega_a^n, h^{n-1}\right) \cdot \\ &\sum_{\omega_p^n \in \Omega_p^n} p\left(\omega_p^n \mid \psi^n, \omega_a^n, r_p^n = F, h^{n-1}\right) \cdot \\ &\max_{r_a^n \in \{T,F\}} EU_a\left(r_a^n \mid \psi^n, \omega_a^n, r_p^n = F, \omega_p^n, h^{n-1}\right) \end{aligned} \quad (2)$$

where $EU_a\left(\psi_a^{n+1} \mid h^n\right)$ is the utility of MERN for revealing ψ_a^{n+1} chips in round $n+1$, the history h^n is $h^{n-1} \cup \left\{\psi^n, \omega_a^n, r_p^n = T\right\}$ and $EU_a\left(r_a^n \mid \psi^n, \omega_a^n, r_p^n = F, \omega_p^n, h^{n-1}\right)$ is the utility for MERN for response r_a^n to the counter-proposal ω_p^n.

Suppose MERN is making the decision r_a^n to respond to a counter-proposal ω_p^n. Its utility from responding to the proposal depends on making the best possible revelation decision ψ_a^{n+1} at round $n+1$ given its response. Formally, we write

$$\begin{aligned} &EU_a\left(r_a^n \mid \psi^n, \omega_a^n, r_p^n = F, \omega_p^n, h^{n-1}\right) = \\ &\max_{r_a^n \in \{T,F\}} \max_{\psi_a^{n+1} \in \Psi_a^{n+1}} EU_a(\psi_a^{n+1} \mid h^n) \end{aligned} \quad (3)$$

where h^n is $h^{n-1} \cup \left\{\psi^n, \omega_a^n, r_p^n = F, \omega_p^n, r_a^n\right\}$. Suppose that MERN is making the decision ψ_a^{n+1} to reveal chips at round $n+1$. In this round, it is the person who makes the first proposal. Therefore, the expected utility of MERN's revelation decision depends on the models $p\left(\omega_p^{n+1} \mid \psi^{n+1}, h^n\right)$ and $p\left(\psi_p^{n+1} \mid h^n\right)$ of how people make proposals and reveal chips in round $n+1$. Formally,

$$\begin{aligned} EU_a\left(\psi_a^{n+1} \mid h^n\right) &= \sum_{\psi_p^{n+1} \in \Psi_p^{n+1}} p\left(\psi_p^{n+1} \mid h^n\right) \cdot \\ &\sum_{w_p^{n+1} \in \Omega_p^{n+1}} p\left(\omega_p^{n+1} \mid \psi^{n+1}, h^n\right) \cdot \\ &\max_{r_a^{n+1} \in \{T,F\}} EU_a\left(r_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, h^n\right) \end{aligned} \quad (4)$$

[3] We detail how to compute this function in the next section.

where $EU_a\left(r_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, h^n\right)$ denotes the expected utility for MERN for making response r_a^{n+1} to the person's proposal ω_p^{n+1} in round $n+1$. If MERN accepts the proposal ω_p^{n+1}, the game will proceed to round $n+2$, and MERN will decide how many chips to reveal. If MERN rejects the proposal, it will make the best counter proposal at round $n+1$. MERN will choose the response that is associated with the best utility out of these two options. Formally,

$$EU_a\left(r_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, h^n\right) = \max\Big[EU_a\left(\psi_a^{n+2} \mid h^{n+1}\right), \max_{\omega_a^{n+1}\in\Omega_a^{n+1}} EU_a\left(\omega_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, r_a^{n+1}=F, h^n\right)\Big] \quad (5)$$

where $EU_a\left(\psi_a^{n+2} \mid h^{n+1}\right)$ is defined in Equation 1, the history h^{n+1} is $h^n \cup \left\{\psi^{n+1}, \omega_p^{n+1}, r_a^{n+1}=T\right\}$ and $EU_a\left(\omega_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, r_a^{n+1}=F, h^n\right)$ is the utility to MERN from making the counter-proposal ω_a^{n+1} in round $n+1$. In turn, the utility for MERN for this proposal depends on its model $p\left(r_p^{n+1} \mid \psi^n, \omega_p^{n+1}, r_a^{n+1}=F, \omega_a^{n+1}, h^n\right)$ of how people respond to proposals

$$EU_a\left(\omega_a^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, r_a^{n+1}=F, h^n\right) = \sum_{r_p^{n+1}\in\{T,F\}} p\left(r_p^{n+1} \mid \psi^{n+1}, \omega_p^{n+1}, r_a^{n+1}=F, \omega_a^{n+1}, h^n\right) \cdot \max_{\psi_a^{n+2}\in\Psi_a^{n+2}} EU\left(\psi_a^{n+2} \mid h^{n+1}\right) \quad (6)$$

where $EU\left(\psi_a^{n+2} \mid h^{n+1}\right)$ is defined in Equation 1.

Learning in the Game

In this section we detail how MERN learned the probabilistic models used to make its decisions in the previous section. For people, this consisted of predicting the set of chips people reveal, the proposals they make, and how they responded to a given proposal. For MERN, this consisted of predicting its final score in the game.

Because the number of possible proposals and revelations is exponential in the total number of chips given to both players we defined equivalence classes over proposals and revelations in the game. For brevity, we describe these features from the point of view of the person, but they can also be defined from the point of view of the agent. Because players cannot observe each other's chips, these classes are based on the chips players need to get to the goal square, but not on the chips in their possession. Specifically, we say that the *needed chips* for the person, denoted NC_p^n, is the set of chips the person needs to get to the goal square given its location on the board at round n (regardless of the chips in the person's possession). For instance, for the person playing the role of the "M" player in Figure 1, this set equals 4 green, 3 cyan chips, and one purple chip. The needed chips for the agent NC_a^n (playing the "O" role) include 4 red chips, 3 gray chips and one purple chip. We denote the set of needed chips for both players as the pair $NC^n = \langle NC_p^n, NC_a^n\rangle$. We say two revelations at round n are equivalent if they agree on the following characterization functions given NC^n:

- $\alpha_1(\psi_p^n) = \left|\psi_p^n \cap NC_a^n\right|$ The number of chips revealed by the person that are useful to the agent. For example, in the game described in Figure 1, suppose that the "M" player has revealed one red and one gray chip. Both of these chips are needed by the "O" player, and therefore this function equals 2.
- $\alpha_2(\psi_p^n) = \left|\psi_p^n \cap NC_p^n\right|$ The number of chips revealed by the person that are useful to the person. In the board described in Figure 1, this feature equals 0 given that the "M" player revealed a red and gray chip.
- $\alpha_3(\psi_p^n) = \left|\psi_p^i \setminus (NC_a^i \cup NC_p^i)\right|$ The number of chips revealed by the person that were not useful for both the agent and the person. In the board described in Figure 1, this feature equals 0 given that the "M" player revealed a red and gray chip.

These functions can also be used to characterize the agent's revelations. An example for two equivalent revelations, given the players' positions in Figure 1, is a revelation of one red and one gray chip and a revelation of two red chips by player "M". In a similar fashion, we also defined a similar equivalence relation over proposals at round n.

Feature Selection

We trained a separate classifier for each action using a data set consisting of people's play in revelation games.[4] Each classifier was fitted with a different subset of features, those for which it achieved the best performance on a held out set of data instances using ten-fold cross-validation. We describe the set of features that were useful for all of the learning tasks, termed the "common feature set":

1. The current round n in the game.
2. The total number of accepted and rejected proposals by the person in rounds 0 through $n-1$. This feature reflects the person's willingness to accept offers throughout the game.
3. $\left|C_a^n\right| - \left|C_a^0\right|$ The difference between the number of chips in the possession of the agent in round n and the chips it was given at the onset of the game. This feature measures the "chips balance" for the agent.
4. $\left|NC_p^n\right|$ This feature measures the number of chips needed by the person in round n. It estimates its current performance in the game without relying on information that is not observed by players in the game.
5. $\sum_{i=o}^{n} \alpha_1\left(\psi_p^i\right)$ This feature measures the sum of the number of chips revealed by the person in rounds 0 through n that were useful to the agent.
6. $\sum_{i=o}^{n} \alpha_2\left(\psi_p^i\right)$ This feature measures the sum of the number of chips revealed by the person in rounds 0 through n that were useful to the person.

[4] The data set consisted of 97 games. We provide additional details on how we collected this data in the next section.

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

Table 1: Predictors and accuracy (LG for Logistic Regression)

7. $\sum_{i=o}^{n} \alpha_3 \left(\psi_p^i\right)$ This feature measures the sum of the number of chips revealed by the person in rounds 0 through n that were not useful for both the person and the agent.

Note that features (4)-(7) are described from the point of view of the human player. We also included these features from the point of view of the agent. The common feature set was incorporated into all of the classifiers we used in the study. For brevity, we do not list features added for specific tasks: predicting whether a proposal made by the agent ω_a^n will be accepted by the person, predicting players' revelations and proposals, and predicting whether the agent would reach the goal in the end of the game.

Prediction Accuracy

Table 1 summarizes the chosen predictor and the resulting accuracy for each learning task, comparing between Support Vector Machines, and linear and logistic regression methods. As shown by the Table, an SVM classifier with a linear kernel was found to be the most accurate for predicting whether a proposal ω_a^n will be accepted by the human responder (71%). A logistic regression model was found to be most accurate for predicting players' revelations (72%) and proposals (68%). A logistic regression classifier was found most accurate (82%) for predicting whether the agent would reach the goal at the end of the game. Lastly, a linear regression was chosen to estimate the agent's final score in the game. This estimate is used by the MERN agent to assign values to its actions in intermediate states in the game. The best prediction was obtained when learning separate regression models for cases in which the agent reached the goal, and did not reach the goal. The expected score was computed as follows:

$$f_a(h^n) = \sum_{G \in \{T,F\}} p\left(G \mid h^{n-1}\right) \cdot \hat{y}\left(h^{n-1} \mid G\right) \quad (7)$$

where $p\left(G \mid h^{n-1}\right)$ is the agent's probability for reaching (for $G = T$) and not reaching (for $G = F$) the goal. $\hat{y}\left(h^{n-1} \mid G\right)$ is the linear regression result, where the features are taken from games where the agent has reached the goal (for $G = T$) and hasn't reached the goal (for $G = F$).

Empirical Methodology

We recruited 410 subjects using Amazon Mechanical Turk(Kittur, Chi, and Suh 2008). Subjects received an identical tutorial and needed to answer a series 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 at least one proposal which allows both players to reach the goal.

Participants were paid according to their performance in the game and divided into several pools. The first pool consisted of people playing other people (45 games). The second pool consisted of people playing a computer agent using a predefined strategy (39 games). The purpose of this pool was to collect reactions to diverse situation, such as their response to proposals and chips revelations that were never made by other people. The third pool was consisted of people playing the MERN agent (24 games). We used two-thirds of the first pool and the complete second pool to train models of human behavior, as described in the previous section. We use the remaining third of the first pool and the complete third pool to test the performance of the MERN agent. In addition, we collected more data for the generalizations process, as described in the next section. MERN used the prediction model described in Section and the Expectimax search described in Section to compute its strategy. We set $k = 2$, meaning the tree was grown for two ply before estimating its score in the game using the function $f_a(h^n)$.

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 using t-test and Mann Whitney tests as appropriate.

We first present a comparison of the average performance of MERN and people (when interacting with other people). The average score of MERN ($101.88 \pm sd = 25.45$, $n = 24$) was significantly higher than people (63.97 ± 47.97, $n = 30$). In addition, MERN was able to get to the goal (96%) significantly more often than people (60%). MERN also had a positive influence on people's individual performance and social welfare in the game: The performance of people interacting with MERN (85.83 ± 41.45, $n = 24$) was significantly higher than the performance of people interacting with other people (63.97 ± 47.97). The aggregate utility for both players (the social welfare) was higher when interacting with MERN (187.71 ± 28.61) than with other people (127.17 ± 75.4).

Strategic Analysis

To facilitate the description of MERN's negotiation strategy, we make the following definitions. Let i, j be the two players in the game (whether agent or person). We define the score to i from a j's proposal ω_j^n in round n, denoted $\pi_i(\omega_j^n \mid h^{n-1})$ as the score to i given that the proposal is realized (the chips are transferred for both sides), and players are moved as close as possible to the goal. We use the associated score to define the following measures, which are computed from the point of view of an oracle (that is, they assume knowledge of players' chips in the game): We define the *helpfulness* of a proposal ω_i^n made by player i at round n as $\pi_j(\omega_i^n \mid h^{n-1})$, that is, the score to j that is associated with the offer made by i. We define the *competitiveness* of a proposal ω_i^n made by player i at round n as $\pi_i(\omega_i^n \mid h^{n-1}) - \pi_j(\omega_i^n \mid h^{n-1})$, that is, the difference in the scores to player i and player j associated with the offer.

The results revealed two opposing patterns in MERN's negotiation strategy. The first pattern included cooperative behavior: MERN learned to make proposals that were significantly more helpful to people (83.52 ± 31.17, $n = 44$), than proposals made by people to other people (37.52 ± 43.49, $n = 204$). The second pattern was competitive: The proposals made by MERN to other people were significantly more competitive (24.89 ± 37.44) than proposals made by people to other people (10.47 ± 57.11). To explain this discrepancy, we need to show how MERN evolved its negotiation strategy over time. In early rounds of the game (rounds 1-4), MERN made highly competitive offers (32.70 ± 33.18, $n = 36$), while in the last round of the game (round 5) it was significantly more generous (-42.50 ± 5.00, $n = 8$). An example of a proposal made by MERN in round 1 consists of asking for 4 red and 4 gray chips in return for 1 green and 4 cyan chips. This proposal is associated with a score of 115 points for MERN and 25 points for the person. Conversely, the proposal made by MERN in round 5 consists of asking for 1 gray in return for 3 green and 1 cyan chips. This proposal is associated with a score of 80 for MERN and 125 for the person.

Playing this "hard-headed" strategy affected people's behavior, in that people's average acceptance rate (per proposal) when interacting with MERN (19%) was significantly lower than when interacting with other people (36%). However, for those games in which MERN rejected peoples' proposals in rounds 1-4, peoples' average acceptance rate for the competitive offers made by MERN in those rounds was very high (88%). Thus, MERN was able to learn to make competitive offers to people that were eventually likely to be accepted. This strategy also affected the efficiency of the offers made by MERN in the game in that 78% of MERN's proposals were pareto optimal, while only 17% of peoples' proposals.

Interestingly, MERN's strategy had a positive effect on people's behavior. People playing MERN offered significantly more pareto optimal proposals (42%), and their proposals were significantly more helpful (68.10 ± 53.26, $n = 79$). This result is striking, in that people's success in the game is attributed not only due to the beneficial offers made by MERN, but also to the change in their strategy.

Generalizing the model

In this section we show that our approach was able to generalize across different number of rounds. We tested MERN's performance on a two rounds interactions, rather than five.[5]. We found, unsurprisingly, that a straightforward deployment of MERN in the modified game (using predictors of human behavior trained on a game with 5 rounds) resulted in poor performance (64.44 ± 51.15, $n = 18$, as compared to 101.88 ± 25.45, $n = 24$ on the original settings). MERN was still able to perform significantly better than people (33.12 ± 36.46, $n = 32$). The social welfare in MERN's games (112.78 ± 90.37, $n = 18$) was also significantly higher than the social welfare in people's games (66.25 ± 70.48, $n = 16$).

We hypothesized that MERN's performance could be improved by separating the data from the original 5-round games into segments of two rounds, without collecting new data. To this end, we recomputed the value of the aggregate features on each of these two-round segments. All other aspects of MERN's decision-making remained the same. Using the new features, MERN significantly improved its performance (84.41 ± 27.33, $n = 17$, $p = 0.015$) on games with two rounds, compared to MERN's play using the original five-round features. The social welfare obtained for both players was also significantly higher (173.82 ± 60.14, $n = 17$, $p = 0.001$) using this approach. An interesting facet of MERN's behavior in these two-round settings was that the adaptation process of its "hard-headed" strategy was very quick. Specifically, when people rejected its offer in the first round, it immediate gave generous offer in the second round. In comparison for the low acceptance rate in the five-rounds games (19%), here people accepted 52% of MERN proposals.

[5]85% of the games were ended before the fifth round, so increasing the rounds number is not likely to change the results significantly.

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. In empirical investigations, the agent was able to outperform people playing other people. It was also able to generalize to different settings that varied rules and situational parameters of the game without the need to accumulate new data. The agent learned to play a "hard-headed" strategy, which was shown to be very effective when playing revelation games with people. This work is a first step towards a general argumentation system in which agents integrate explanations and justifications within their negotiation process.

Acknowledgments

This work was supported in part by ERC grant #267523, Marie Curie reintegration grant 268362, the Google Inter-university center for Electronic Markets and Auctions, MURI grant number W911NF-08-1-0144 and ARO grants W911NF0910206 and W911NF1110344.

References

Bench-Capon, T.; Atkinson, K.; and McBurney, P. 2009. Altruism and agents: an argumentation based approach to designing agent decision mechanisms. In *Proceedings of AAMAS*.

Coehoorn, R. M., and Jennings, N. R. 2004. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of EC*.

Gal, Y.; Grosz, B.; Kraus, S.; Pfeffer, A.; and Shieber, S. 2010. Agent Decision-Making in Open-Mixed networks. *Artificial Intelligence*.

Gal, Y.; Kraus, S.; Gelfand, M.; Khashan, H.; and Salmon, E. 2011. An adaptive agent for negotiating with people in different cultures. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(1):8.

Haim, G.; Gal, Y. K.; Gelfand, M.; and Kraus, S. 2012. A cultural sensitive agent for human-computer negotiation. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 451–458. International Foundation for Autonomous Agents and Multiagent Systems.

Hindriks, K., and Tykhonov, D. 2008. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proc. of AAMAS*, 331–338.

Jonker, C. M.; Robu, V.; and Treur, J. 2007. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems* 15(2):221–252.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with Mechanical Turk. 453. ACM Press.

Oshrat, Y.; Lin, R.; and Kraus, S. 2009. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proc. of AAMAS*.

Peled, N.; Kraus, S.; and Gal, K. 2011. A study of computational and human strategies in revelation games. In *Proc. of AAMAS*.

Sarne, D., and Grosz, B. J. 2007. Estimating information value in collaborative multi-agent planning systems. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, 48–55.

Sarne, D., and Kraus, S. 2003. The search for coalition formation in costly environments. In *Proceedings of the Seventh International Workshop on Cooperative Information Agents (CIA'03)*, 117–136.