

**Modeling how Humans Reason about Others
with Partial Information**

Sevan G. Ficici
and
Avi Pfeffer

TR-01-07



Computer Science Group
Harvard University
Cambridge, Massachusetts

Modeling how Humans Reason about Others with Partial Information

Sevan G. Ficici and Avi Pfeffer
Technical Report TR-01-07
Maxwell-Dworkin Laboratory
School of Engineering and Applied Sciences
Harvard University
Cambridge Massachusetts 02138 USA

March 30, 2007

Abstract

Computer agents participate in many collaborative and competitive multi-agent domains in which humans make decisions. For computer agents to interact successfully with people in such environments, an understanding of human reasoning is beneficial. In this paper, we investigate the question of how people reason strategically about others under uncertainty and the implications of this question for the design of computer agents. Using a situated partial-information negotiation game, we conduct human-subjects trials to obtain data on human play. We then construct a hierarchy of models that explores questions about human reasoning: Do people explicitly reason about other players in the game? If so, do people also consider the possible states of other players for which only partial information is known? Is it worth trying to capture such reasoning with computer models and subsequently utilize them in computer agents? We further address these questions by constructing computer agents that use our models; we deploy our agents in further human-subjects trials for evaluation. Our results indicate that people do reason about other players in our game and that the computer agents that best model human play obtain superior scores.

1 Introduction

With increasing frequency, computer agents are participating in collaborative and competitive multi-agent domains in which humans reason strategically to make decisions. Examples include online auctions, financial trading, scheduling, and computer gaming (online and video). The deployment of computer agents in such domains requires that the agents understand something about human behavior so that they can interact successfully with people; the computer agents

must be sensitive to both how people reason in strategic settings as well as the social utilities people employ to inform their reasoning. To date, these design requirements for computer agents have received relatively little attention.

Models of human reasoning have been shown to be helpful for agent design [Gal et al., 2004]. In their work, [Gal et al., 2004] use a situated full-information two-player negotiation game; in this game, one player proposes an exchange of resources and the other player responds by accepting or rejecting the proposal. [Gal et al., 2004] conducted human-subjects trials and subsequently learned a model of the responder’s utility function; this model was utilized to construct a computer proposer agent that took human responder behavior into account. Nevertheless, [Gal et al., 2004] leaves unexplored the questions of how humans reason strategically and under uncertainty: the responder of their game reasons only after the proposal is received, and so requires neither strategic reasoning (that is, reasoning by one player about what another player might do) nor reasoning under uncertainty about another player’s state. Here, we are interested to focus on these two aspects of human reasoning.

In this paper, we investigate questions about how people reason strategically about others under uncertainty and the implications of these questions for the design of computer agents. For example, is human reasoning *reflexive*, where the behavior of other players is accounted for implicitly, that is, without explicit consideration of other players’ possible actions or states? Or, do humans somehow try to reason strategically about other players by consulting models that they maintain about them? If so, then does such reasoning also consider the possible states of other players? And, if either of these possibilities are true, then is it worth trying to capture such reasoning with computer models and subsequently utilize them in computer agents?

Our investigation of these questions leads us to a variety of contributions. We construct a hierarchy of models, whereby models are differentiated not only by whether they include strategic reasoning, but also by whom that reasoning concerns. We provide learning algorithms for our models. The human-subjects experiments we conduct provide a wealth of data which we use to train and test our models and which can be used for further investigations. Finally, our analyses provide insight into whether and how humans behave strategically under uncertainty and the issues that surround engineering computer agents to interact with humans.

We find that a model’s ability to predict human behavior depends upon whether we model the human as using reflexive or strategic reasoning; further, if strategic reasoning is used, then the model’s performance also depends upon whom we model the human to be reasoning about. Beyond the pattern of reasoning, a model’s performance depends upon the utility functions that we model humans to be using. We also find that the benefit to be gained from increasingly sophisticated models diminishes, while the computational costs of such models increases.

While many fields relate to the goal of creating computer agents that take human strategic thinking into account, none have yet placed a spotlight on this goal. For example, classical game theory [Fudenberg and Tirole, 1998] precludes

modeling agents as anything other than rational actors, severely restricting the types of reasoning we can capture. Further, the rationality assumption is well known to be violated in many real-world domains. In recognition of the fact that human decision making often deviates from full rationality, the field of behavioral economics [Rabin, 1998, Camerer, 2003] seeks to explain the gap between actual human decision-making and that of classical game theory’s *homo economicus*. Nevertheless, the decision-making domains studied in behavioral economics are very abstract and lack *situatedness*; being situated entails interaction with and within an environment [Lueg and Pfeifer, 1997]. Further, behavioral economics does not concern itself with engineering computational agents that can interact successfully with human decision makers.

The field of multi-agent systems (MAS) [Weiss, 2000, Kraus, 2001] is concerned with engineering computer agents that operate and interact in environments containing other agents. Nevertheless, much MAS research focuses on environments comprised of only computer agents, and so agents tend to be viewed as rational actors; this assumption simplifies the task of *recursive modeling* [Gmytrasiewicz and Durfee, 2001], where an agent models another agent as an entity that itself models other agents. Recursive models involving bounded-rational agents are also examined in MAS [Vidal and Durfee, 1995], but such models are not generally intended to capture the peculiarities of bounded rationality in human reasoning. Research on modeling human emotion and its effect on behavior [Seif El Nasr et al., 2000, Gratch and Marsella, 2005] generally involves no learning at all or no learning from real human data. Thus, the enterprise of modeling strategic human decision-making for the purpose of engineering computer agents that are sensitive to human behavior stands apart from most MAS research.

2 Three-Player Negotiation Game

We require an environment that is appropriate for investigating how humans reason about others under uncertainty. The environment must be simple enough for analysis and agent engineering to be tractable, yet rich enough to reflect salient features of the real world. We desire an environment that is situated, can provide partial information, and promotes reasoning about other players. The Colored Trails (CT) framework [Grosz et al., 2004] meets our requirements.

CT is a highly configurable, situated multi-agent task environment that can be played by humans and computer agents. CT captures the important high-level features of decision-making found in many real-world environments; CT is sufficiently abstract to focus on high-level features, yet is simultaneously grounded in a situated task domain. The situated task activities presented by the CT environment distinguish CT from the games often used in behavioral economics, which tend to present highly abstract decision-making scenarios. [Allain, 2006] demonstrates a framing effect in which a game presented as a situated task activity elicits stronger concern with social factors such as fairness; the same underlying game presented in a more abstract payoff matrix form en-

genders behavior more in line with rational Nash equilibrium play. These results demonstrate the importance of eschewing highly abstract games in favor of situated activity if we are interested to learn about how people reason in real-world settings.

Using the CT environment, we construct a three-player partial-information negotiation game. Our game is played on a 4x4 board of colored squares; each square is one of five colors. Each of the three players has a piece on the board as well as a collection of colored chips that can be used to move her piece; a player may move her piece to an adjacent square only if she has a chip of the same color as the square. After the piece is moved, the chip is discarded by the player. The board also has a square that is designated as the *goal*. The objective of each player is to move her piece as close as possible to, and preferably onto, the goal square. We generate initial conditions such that players can usually improve their ability to approach the goal by trading chips.

Each game proceeds as follows. Each player is randomly assigned to one of the three roles in the game, denoted *proposer 1*, *proposer 2*, and *responder*. Each player knows the state of the board (board colors, goal location, locations of all player pieces) and the chips that she possesses. Proposers also know the chips possessed by the responder, but not by each other; this is the source of uncertainty in the game. The responder knows the chips possessed by both proposers. The proposers are allowed to exchange chips with the responder, but not each other. The proposers simultaneously formulate their proposals to exchange chips and submit them to the responder; any redistribution of chips between a proposer and the responder is valid, including giving away all chips, requesting all chips, or anything in between. A proposal may also leave the chips unchanged. The responder then chooses to accept no more than one of the two proposals, or declines them both. After the responder's decision is made, the CT system informs the proposers of the outcome and automatically moves all three players' pieces to obtain the maximal possible score for each player, given the chips possessed.

A number of factors may influence the offer a proposer ultimately makes. First, a proposer may need certain chips to improve its utility. But, the responder may also require certain chips, and these requirements may or may not be synergistic with the needs of the proposer. In addition, because the responder can accept no more than one proposal, there exists a competitive relationship between the proposers. Therefore, a proposer may want to reason about what the other proposer may offer. Since proposers have partial information about each other (they know each other's location on the board, but do not know about each other's chips), reasoning under uncertainty is required. The behavior of a proposer explores the tension between fulfilling its own utility function and that of the responder in the face of unknown competition.

3 Player Models

We construct several models of how humans reason in our game. By examining the performance of a variety of models, we hope to identify prominent features of human reasoning and engineer effective computer agents. We concentrate on modeling the proposer in our game, since our game invites strategic reasoning under uncertainty for this role. Nevertheless, most of our proposer models will require models of the responder to operate.

3.1 Reflexive Models

Our most basic models, the reflexive models, are based upon the architecture of [Gal et al., 2004, Gal and Pfeffer, 2006]. All of our models (reflexive and strategic) make use of only two simple features that quantify proposal properties; these features are rather general and can be applied to almost any negotiation game. Though our game involves three players, each proposal specifies a pair-wise interaction between two players. Let *self-benefit* (SB) quantify the change in score a player will receive if a proposal is accepted, and *other-benefit* (OB) quantify the change in score the other player will receive if the proposal is accepted. Other features were investigated with cross validation, but were found not to improve our models. In particular, we considered categorical discretizations of SB and OB to indicate whether benefit is positive, negative, or zero; these were believed to be useful to express “rational” play. We also considered two features that addressed the fairness of a proposal, one that corresponds to the Nash bargaining concept [Nash, 1950] and another that considers the context of alternative proposals that could have been made [Falk et al., 2003].

Let each proposal $O = \langle \text{SB}, \text{OB} \rangle$ be a vector of feature values; let $\mathbf{w} = \langle w_{\text{SB}}, w_{\text{OB}} \rangle$ be the vector of corresponding feature weight parameters. We define the linear utility function $U : \mathbb{O} \rightarrow \mathbb{R}$ on the space \mathbb{O} of offers to be

$$U(O) = \sum_l w_l \cdot O_l. \tag{1}$$

Let ϕ denote the *status-quo*, which for the responder represents the option of rejecting both offers, and for the proposer represents the proposal that no chips change hands. Note that $U(\phi) = 0$, since $\text{SB} = \text{OB} = 0$.

Since all humans will likely not share the same utility function, we use mixture models to cluster human play into different behavioral types. We have T types; an individual of type t_i uses utility function U^{t_i} with weight vector \mathbf{w}^{t_i} . Let ρ^{t_i} be the proportion of individuals of type t_i .

Humans also select offers non-deterministically; we are prone to make errors. Further, since our models will not be perfect, we care to have our models attach probabilities to different outcomes. To accommodate these factors, we convert proposal utilities to probabilities of selection with a soft-max function. Let \mathcal{O} be a set of options to choose from. The probability that an individual of type t_i will select the m -th proposal in \mathcal{O} is

$$\Pr(\text{selected} = O^m | \mathcal{O}, t_i) = \frac{e^{U^{t_i}(O^m)}}{\sum_k e^{U^{t_i}(O^k)}}. \quad (2)$$

Taking an expectation over all behavioral types gives

$$\Pr(\text{selected} = O^m | \mathcal{O}) = \sum_i \Pr(\text{selected} = O^m | \mathcal{O}, t_i) \rho^{t_i}. \quad (3)$$

A reflexive model of player behavior makes the assumption that the player does not explicitly reason about other players in the decision-making process. Such an assumption is clearly appropriate for modeling the responder of our game, since the responder’s decision-making requires neither strategic thinking nor reasoning under uncertainty; the responder simply reacts to the decisions that are made by the proposers. The responder has three choices $\mathcal{O} = \{O, \bar{O}, \phi\}$, where O and \bar{O} are the two proposer offers, and ϕ is the null offer, which the responder selects if it cares for neither proposer offer. Our model of responder behavior is a mixture model precisely of the form described by Equations (1)–(3); the $w_i^{t_i}$ are model parameters that are easily interpreted to represent the responder’s utility function.

Our simplest model of proposer behavior is also reflexive; this model assumes that a proposer does not explicitly reason about the responder and other proposer, even though such reasoning is meaningful in our game. Here, $\mathcal{O} = \{O^1 \dots O^M\}$ is the set of proposals that the proposer can make, given its game state. Like the responder model, the reflexive proposer model is described by Equations (1)–(3), and the $w_i^{t_i}$ are model parameters. Unlike the responder model, however, the $w_i^{t_i}$ are not so easily interpreted to represent the proposer’s utility function. This is because, to the extent that a proposer implicitly reasons about other players, model training will cause the w to represent, as best they can, some amalgam of the proposer’s utility function and implicit reasoning process. We denote our reflexive responder and proposer models $R\{\}$ and $P\{\}$, respectively. We use the empty curly braces to indicate a reflexive model. Below we introduce strategic proposer models that reason about other players using other models; these other models are indicated by being enclosed in the braces.

3.2 Strategic Models

The reflexive models explicitly reason about the options they have, but not about the other players in the game. Our game’s structure makes a reflexive model appropriate for the responder; but, it is possible that a reflexive model of proposer behavior can be improved upon. Here, we introduce new strategic models of proposer behavior.

More sophisticated than $P\{\}$, we can model a proposer as reasoning explicitly about the responder, but still not the other proposer. The existence of the other proposer is acknowledged, but the partial information about the other proposer that the game provides is ignored. Unfortunately, without explicit reasoning about the other proposer, we cannot utilize our responder model $R\{\}$,

because it requires that we specify both proposers' offers. To address this problem, we construct a modified responder model $R^- \{ \}$ that does not require two proposer offers. Our proposer model then utilizes this modified responder model to reason about the responder, specifically about how likely the proposer's offers are to be accepted by the responder.

The modified responder model $R^- \{ \}$ accepts two known choices $\mathcal{O} = \{O, \phi\}$ as input and represents the unknown \bar{O} with a fixed "generic" proposal that has utility v^{t_i} (v^{t_i} is a model parameter in addition to feature weights $w_i^{t_i}$):

$$\Pr(\text{selected} = O^m | \mathcal{O}, t_i) = \frac{e^{U^{t_i}(O^m)}}{e^{v^{t_i}} + \sum_k e^{U^{t_i}(O^k)}}, \quad (4)$$

$$\Pr(\text{selected} = \bar{O} | \mathcal{O}, t_i) = \frac{e^{v^{t_i}}}{e^{v^{t_i}} + \sum_k e^{U^{t_i}(O^k)}}, \quad (5)$$

where v^{t_i} is the generic utility given by responder type t_i to the unknown proposal \bar{O} . Our first strategic proposer model, therefore, is denoted $P\{R^- \{ \} \}$, to indicate that a pre-learned responder model of type $R^- \{ \}$ is embedded in the proposer model.

Proposer model $P\{R^- \{ \} \}$ uses $R^- \{ \}$ to reason about how the responder might react to possible offers, thus converting its utilities to expected utilities:

$$EU^{t_i}(O) = U^{t_i}(O) \cdot \Pr(\text{selected} = O | \mathcal{O}, \phi) \quad (6)$$

The expected utilities, rather than the plain utilities, are then plugged into (2) and (3). The utility functions U^{t_i} used by $P\{R^- \{ \} \}$ contain the parameters $w_i^{t_i}$ of the model that are to be learned.

Next in our hierarchy is a model of proposer behavior that asserts that a proposer explicitly reasons about both the responder and other proposer; this model takes into account the partial information available about the other proposer to reason about what that proposer might offer and how its offers might affect the responder's decision. To perform such reasoning, the proposer model must itself utilize models of the responder and other proposer. One such proposer model is $P\{R \{ \}, P \{ \} \}$. This model embeds our reflexive model of responder behavior; it also models the other proposer as being reflexive. Another such proposer model is $P\{R \{ \}, P\{R^- \{ \} \} \}$, which models the other proposer as reasoning about the responder but not the first proposer. We refer to $P\{R \{ \}, P \{ \} \}$ and $P\{R \{ \}, P\{R^- \{ \} \} \}$ as *level-one* models; the models of proposer behavior embedded in the level-one models do not model the other proposer ($P\{R^- \{ \} \}$ only models the responder), and so are level-zero proposer models. The parameters of a level-one model remain the $w_i^{t_i}$ feature weights of its utility functions; the embedded models are pre-learned, and so do not contribute any level-one parameters.

A *level-two* proposer model is still more sophisticated; it states that a proposer (for clarity named P1) reasons about both the responder and other proposer (for clarity named P2), and further states that P1 believes that P2 itself reasons about P1. In the level-two model $P\{R \{ \}, P\{R \{ \}, P \{ \} \} \}$, P1 believes

that P2 models P1 as being reflexive; in level-two model $P\{R\}, P\{R\}, P\{R^{-}\{\}\}\}$, P1 believes that P2 models P1 as reasoning about the responder, but not P2. The parameters of a level-two model are again the w of its utility function. In principle, we can create a level- N proposer model by recursively embedding a level- $(N - 1)$ model.

The expected utility $EU(O)$ of an offer O to a level- N model (say, $P\{R\}, P\{N-1\}$) is

$$EU(O) = \sum_{\bar{C} \in \bar{\mathcal{C}}} \sum_{\bar{O} \in \bar{\mathcal{O}}|\bar{C}} \Pr(\bar{C}) \cdot \Pr(\text{selected} = \bar{O}|\bar{C}) \cdot \Pr(\text{selected} = O|\bar{O}, \phi) \cdot U(O). \quad (7)$$

Equation (7) operates as follows. We consider all possible chipsets \bar{C} that the other proposer might have. For each such chipset \bar{C} , we consider all possible offers \bar{O} that the other proposer could make. To calculate our expected utility for offer O , we need to consider several factors. First, is the probability that the other proposer has a certain chipset \bar{C} . Then, given \bar{C} , we use our level $N - 1$ model of the other proposer to estimate the probability that it will make offer \bar{O} . We next use our model of the responder to estimate the probability that it will accept our proposal O over \bar{O} and ϕ . The product of these probabilities times our utility for O gives our expected utility for O . The expected utilities for all the offers in \mathcal{O} are then plugged into the soft-max equation (2) to obtain a probability distribution over offers for the level- N model.

3.3 Emulating and Strategizing Agents

We utilize the models described above in two ways to construct computer agents that play proposers in our game. Let \mathcal{O} be the set of possible offers that the computer agent can make in a game. Our first approach uses a human model to achieve a crude form of emulation; the computer agent queries the model to learn which offer in \mathcal{O} , according to the model, a human is most likely to make and makes that offer. Our second approach uses models of human responders and proposers to strategize in order to maximize the agent’s expected utility. Here, the computer agent uses a pattern of reasoning that is identical to a level- N proposer model, except that the computer agent’s utility function is entirely selfish (that is, $w_{SB} > 0$ and $w_{OB} = 0$). Thus, a strategic computer agent that uses a level- N proposer model is performing level- $N + 1$ reasoning.

4 Learning

Models are trained by gradient descent. Let $g(\text{selected} = O^*|\mathcal{O})$ be the probability that some model assigns to the proposal O^* that was actually selected

by a human player, given the set of options \mathcal{O} . The error function F that we minimize measures negative log likelihood of the data (containing D decision instances), given a model:

$$F = - \sum_{d=1}^D \ln(g(\text{selected} = O^{*d} | \mathcal{O}^d)). \quad (8)$$

The derivative of the error function with respect to some model parameter $w_l^{t_i}$ (or v^{t_i} in $R^- \{\}$) is

$$\frac{\partial F}{\partial w_l^{t_i}} = - \sum_{d=1}^D \frac{\frac{\partial g}{\partial w_l^{t_i}}}{g(\text{selected} = O^{*d} | \mathcal{O}^d)}. \quad (9)$$

Let α be our learning rate. The weight-update equation for some model parameter $w_l^{t_i}$ is

$$w_l^{t_i} \leftarrow w_l^{t_i} - \alpha \cdot \frac{\partial F}{\partial w_l^{t_i}}. \quad (10)$$

To update the probability ρ^{t_i} of type t_i in the mixture, we multiply by the negative of the gradient, which turns out to be equivalent to using Bayes' rule:

$$\begin{aligned} \rho^{t_i} &\leftarrow - \frac{\partial F}{\partial \rho^{t_i}} \cdot \rho^{t_i} \\ &= \sum_{d=1}^D \frac{g(\text{selected} = O^{*d} | \mathcal{O}^d, t_i) \cdot \rho^{t_i}}{g(\text{selected} = O^{*d} | \mathcal{O}^d)}. \end{aligned} \quad (11)$$

Equation (9) requires that we further calculate the partial derivative of function g , which represents the behavior of our model. Though g varies with each model, we find that the derivative of g has a similar structure over all models. The partial derivative of g with respect to some $w_l^{t_i}$ for our reflexive models (and $R^- \{\}$) is:

$$\frac{\partial g(O^* | \mathcal{O}, t_i)}{\partial w_l^{t_i}} = g(O^* | \mathcal{O}, t_i) \left(O_l^* - \sum_k O_l^k \cdot g(O^k | \mathcal{O}, t_i) \right) \quad (12)$$

The partial derivative of g with respect to v^{t_i} in $R^- \{\}$ is:

$$\frac{\partial g(O^* | \mathcal{O}, t_i)}{\partial v^{t_i}} = - \frac{e^{U^{t_i}(O^*)} \cdot e^{v^{t_i}}}{\left(e^{v^{t_i}} + \sum_k e^{U^{t_i}(O^k)} \right)^2} \quad (13)$$

The partial derivative of g with respect to some $w_l^{t_i}$ for model $P\{R^- \{\}\}$ is:

$$\frac{\partial g(O^*|\mathcal{O}, t_i)}{\partial w_i^{t_i}} = g(O^*|\mathcal{O}, t_i) \left(O_i^* \cdot \Pr^{R^-\{}}(O^*|\mathcal{O}) - \sum_k O_i^k \cdot g(O^k|\mathcal{O}, t_i) \cdot \Pr^{R^-\{}}(O^k|\mathcal{O}) \right) \quad (14)$$

The partial derivative of g with respect to some $w_i^{t_i}$ for a level- N model is:

$$\frac{\partial g(O^*|\mathcal{O}, t_i)}{\partial w_i^{t_i}} = g(O^*|\mathcal{O}, t_i) \left(O_i^* \cdot Z(O^*) - \sum_k O_i^k \cdot Z(O^k) \cdot g(O^k|\mathcal{O}, t_i) \right), \quad (15)$$

where

$$Z(O) = \sum_{\bar{c} \in \bar{c}} \sum_{\bar{o} \in \bar{O}|\bar{c}} \Pr(\bar{c}) \cdot \Pr^{P\{N-1\}}(\text{selected} = \bar{O}|\bar{O}) \cdot \Pr^{R\{}}(\text{selected} = O|\bar{O}, \phi). \quad (16)$$

Note that using Equation (15) entails the calculations performed in (7). For higher level models, this is recursive, making the cost of a level- N model grow exponentially with N . Thus, only small N are feasible; nevertheless, we do not expect human reasoning to correspond to large N .

5 Human-Subjects Trials

We recruited 69 human subjects to play our negotiation game for 15 rounds; over half of their total compensation was determined by the scores they accumulated over the rounds. Subjects were randomly matched in each round. To emphasize that they were playing a sequence of one-shot games, not an iterated game, subjects performed an unrelated activity between rounds. We obtained a total of 268 games in which all three players were human subjects. Another 221 games involved a human Responder deciding between two hand-crafted offers; the hand-crafted offers are not used to train proposer models.

Using cross validation, we determined that our model of responder behavior best fit the data with two types; we then used two types for our proposer models, as well. We trained two responder models $R\{\}$ and $R^-\{\}$, and two level-zero proposer models $P\{\}$ and $P\{R\{\}$. The high cost of training the level-one models $P\{R\{\}, P\{\}$ and $P\{R\{\}, P\{R^-\{\}\}$ (≈ 2 hours per epoch of gradient descent) required us to use an alternative; rather than learn, we chose to adopt the utility function of $R\{\}$ to be the utility function for the level-one proposer models. While necessitated by an impractical training time, this

approach allowed us to consider the hypothesis that responders and proposers share the same utility function. Using these models, we constructed six computer proposer agents. Two agents used models $P\{\}$ and $P\{R\{\}\}$, respectively, to “emulate” human behavior; these agents make the offer that the model says a human proposer is most likely to make. The remaining four proposer agents use the models to construct best replies to what the models say is the expected behavior of human proposers. We recruited an additional 61 human subjects to test the performance of our models and agents. Each game-state was used seven times. In one copy, all players were human. In the other six copies, Proposer 1 was one of our agents, and the other two players were human. This allows us to compare the performance of Proposer 1 across each copy.

6 Results

Table 1 gives the negative log likelihood of our test data given each proposer model; smaller numbers are better. Random guessing produces a negative log likelihood of 537. These results indicate that strategic reasoning fits human proposer behavior better than reflexive reasoning. Reasoning about both the responder and other proposer appears to fit human data about the same as reasoning only about the responder; as we discuss above, however, the level-one models were not trained, but used the utility function of $R\{\}$.

Table 2 summarizes our results from testing our models and computer proposer agents. First, we find that human proposers have many more of their offers accepted than computer proposers. Yet, we find that all of our best-response agents each obtain higher total benefits ($p < 0.01$, paired sign test) in trading than human proposers. The agent playing best-response to $P\{R^-\{\}\}$ obtains the highest total benefit. There is some statistical evidence that this agent outperforms the agent playing best-response to $P\{\}$ ($p < 0.12$); thus, it is insufficient for the computer agent to simply reason about the other proposer—it is important to reason with a good model. Nevertheless, playing best-response to $P\{\}$ outperforms the agent that emulates using $P\{R^-\{\}\}$ ($p < 0.12$). The responder is able to accumulate much higher total benefit when interacting with human proposers than with our computer agents. Human offers are much more generous than required in order to be accepted by the responder. This gap between what human proposers offer and what is required by human responders is exploited by our best-response agents to give them higher total benefit.

Our data show that human proposers model the responder to reason. Our data also show that modeling humans as also reasoning under uncertainty about the other proposer outperforms modeling humans as acting reflexively; nevertheless, the performance of these level-one models do not improve upon the simpler $P\{R\{\}\}$, which only models the responder, so the extent to which humans reason under uncertainty is unclear. What is clear, however, is that the high training cost for our level-one models raises the question of whether it is worth modeling people as hypothetically filling in missing state information about other players.

Table 1: Fit to data (negative log likelihood).

$P\{\}$	$P\{R\{\}\}$	$P\{R\{\}, P\{\}\}$	$P\{R\{\}, P\{R^-\{\}\}\}$
474	438	443	442

Table 2: Proposer performance. PB is expected total benefit to Proposer 1; RB is expected total benefit to responder when accepting offers of Proposer 1; AO is expected number of accepted offers by Proposer 1.

Player Type (Proposer 1)	PB	RB	AO
Human	1292	1609	22.10
Emulator: $P\{\}$	529	311	6.75
Emulator: $P\{R^-\{\}\}$	1212	667	16.00
Best-response: $P\{\}$	1324	379	14.53
Best-response: $P\{R^-\{\}\}$	1516	650	16.57
BR: $P\{R\{\}, P\{\}\}$	1325	365	14.90
BR: $P\{R\{\}, P\{R^-\{\}\}\}$	1353	398	15.15

7 Conclusion

Our paper concerns strategic human reasoning about others under uncertainty and its implications for the design of computer agents intended to interact with humans. We conduct human-subjects trials in which people play a three-player partial-information negotiation game. We then construct a hierarchy of models to investigate how people reason. We have shown that humans are not reflexive reasoners, but rather model other players to think strategically. We have demonstrated that models of human reasoning about others can be effectively leveraged to construct computer agents that interact successfully with humans; computer agents outperform human players. Similar demonstrations in other domains will be useful future work. Also, engineering issues associated with the execution time of level-one models need to be addressed. We are examining sampling schemes to make these models feasible to train and use in real time.

Acknowledgments

The research reported in this paper was supported in part by NSF grant CNS-0453923 and AFOSR grant FA9550-05-1-0321. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, AFOSR, or the U.S. Government.

References

[Allain, 2006] Allain, F. A. (2006). The effect of context on decision making in Colored Trails. Technical report, Harvard College. Undergraduate honors

thesis.

- [Camerer, 2003] Camerer, C. F. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. The Roundtable Series in Behavioral Economics. Princeton University Press.
- [Falk et al., 2003] Falk, A., Fehr, E., and Fischbacher, U. (2003). On the nature of fair behavior. *Economic Inquiry*, 41(1):20–26.
- [Fudenberg and Tirole, 1998] Fudenberg, D. and Tirole, J. (1998). *Game Theory*. MIT Press.
- [Gal and Pfeffer, 2006] Gal, Y. and Pfeffer, A. (2006). Predicting people’s bidding behavior in negotiation. In *Autonomous Agents and Multi-Agent Systems*.
- [Gal et al., 2004] Gal, Y., Pfeffer, A., Marzo, F., and Grosz, B. J. (2004). Learning social preferences in games. In *National Conference on Artificial Intelligence (AAAI)*.
- [Gmytrasiewicz and Durfee, 2001] Gmytrasiewicz, P. J. and Durfee, E. H. (2001). Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems Journal*, 4(3):233–272.
- [Gratch and Marsella, 2005] Gratch, J. and Marsella, S. (2005). Evaluating a computational model of emotion. *Autonomous Agents and Multi-Agent Systems*, 11(1):23–43.
- [Grosz et al., 2004] Grosz, B. J., Kraus, S., Talman, S., Stossel, B., and Havlin, M. (2004). The influence of social dependencies on decision-making: Initial investigations with a new game. In *Autonomous Agents and Multi-Agent Systems*.
- [Kraus, 2001] Kraus, S. (2001). *Strategic Negotiation in Multiagent Environments*. MIT Press.
- [Lueg and Pfeifer, 1997] Lueg, C. and Pfeifer, R. (1997). Cognition, situatedness, and situated design. In *Conference on Cognitive Technology*.
- [Nash, 1950] Nash, J. (1950). The bargaining problem. *Econometrica*, 18:155–162.
- [Rabin, 1998] Rabin, M. (1998). Psychology and economics. *Journal of Economic Literature*, 36:11–46.
- [Seif El Nasr et al., 2000] Seif El Nasr, M., Yen, J., and Ioerger, T. R. (2000). Flame—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3:219–257.

[Vidal and Durfee, 1995] Vidal, J. M. and Durfee, E. H. (1995). Recursive agent modeling using limited rationality. In *International Conference on Multi-Agent Systems*.

[Weiss, 2000] Weiss, G., editor (2000). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press.