

Simultaneously Modeling Humans' Preferences and their Beliefs about Others' Preferences

Sevan G. Ficici
School of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts USA
sevan@eecs.harvard.edu

Avi Pfeffer
School of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts USA
avi@eecs.harvard.edu

ABSTRACT

In strategic multiagent decision making, it is often the case that a strategic reasoner must hold beliefs about other agents and use these beliefs to inform its decision making. The behavior thus produced by the reasoner involves an interaction between the reasoner's beliefs about other agents and the reasoner's own preferences. A significant challenge faced by model designers, therefore, is how to model such a reasoner's behavior so that the reasoner's preferences and beliefs can each be identified and distinguished from each other. In this paper, we introduce a model of strategic reasoning that allows us to distinguish between the reasoner's utility function and the reasoner's beliefs about another agent's utility function as well as the reasoner's beliefs about how that agent might interact with yet other agents. We show that our model is uniquely identifiable. That is, no two different parameter settings will cause the model to give the same behavior over all possible inputs. We then illustrate the performance of our model in a multiagent negotiation game played by human subjects. We find that our subjects have slightly incorrect beliefs about other agents in the game.

Categories and Subject Descriptors

I.2.6 [Learning]: Knowledge acquisition; I.6.5 [Model Development]: Modeling methodologies; J.4 [Social and Behavioral Sciences]: Economics

General Terms

Experimentation, Human Factors, Performance, Design

Keywords

Human models, Uncertainty, Reasoning, Negotiation

1. INTRODUCTION

Many multiagent domains involve both human and computer decision makers that are engaged in collaborative or competitive activities. Examples include online auctions, financial trading, scheduling, and computer gaming (online and video). To construct computer agents that can interact successfully with human participants, we need to under-

stand several things about human reasoning in multiagent domains. Behavioral economics [17, 3] has shown that people employ social utility functions that deviate from rational game-theoretic prescription. Learning about the social utilities humans use has been shown by Gal et al. [8] to be beneficial for the design of computer agents. Psychological theories of mind explore people's reasoning about others [10, 4, 13, 2]; one aspect of a theory of mind concerns *beliefs* about others. This aspect of modeling human reasoning for use in computer agents is left open by Gal et al. [8].

We are interested to investigate beliefs that human reasoners hold about other agents. Do people hold beliefs about another agent's preferences or intentions, and use these beliefs to inform decision making? If so, then what are these beliefs? Are these beliefs correct? How do the beliefs about others' preferences or intentions relate to the preferences or intentions of the reasoner? Do people believe others to be the same as themselves? If people use beliefs to reason, then their behavior is the result of an interaction between their beliefs about others and their own utility function; can we distinguish between the two and untangle a person's beliefs from her preferences? For example, the negotiation experiments of Gal et al. [8] indicate that human players often make offers that are more generous than necessary to be accepted. This result may indicate a gap between the proposer's beliefs about the responder and the responder's actual behavior. Alternatively, the human player may have a strong preference to be generous. If we are interested to identify an explanation, we must construct a model such that we can distinguish the reasoner's preferences from its beliefs about another agent. Otherwise, model parameters will represent some amalgam of these various factors.

In this paper, we use AI techniques to address the above questions. We introduce a model of strategic human reasoning that allows us to distinguish between three factors: the human's own utility function, the human's beliefs about another agent's utility function, and the human's beliefs about how that other agent may interact with yet other agents. To support our claim that the model allows us to untangle these factors, we show that the model is uniquely identifiable; that is, no two different parameters sets can produce the same model behavior over all possible inputs. We provide a learning algorithm for our model and analyze how well learned models fit data obtained from human-subjects trials of a multiagent negotiation game. In addition to investigating our general model, we also examine constrained versions that correspond to particular belief patterns. For example, it may be that a human player believes other agents

Cite as: Title, Author(s), *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

to share the same utility function as the player. In another example, we consider the case where different types of humans have different personal preferences, but share the same beliefs about the preferences of other agents. Our analyses provide insight into whether modeling a person’s beliefs about others’ preferences separately from the person’s own preferences yields a better model for use in computer agents.

Our work is significantly different from most work in multiagent systems (MAS) [19, 14]. For example, MAS research often focuses on environments comprised of only computer agents; thus, agents tend to be viewed as rational actors [9]. The bounded-rational agents of [18] do not specifically address human boundedness. Research on theories of mind and emotion [16, 11, 1] typically involves no learning at all or no learning from real human data. Finally, investigations of theories of mind do not address the construction of computer agents that are to interact with humans.

2. NEGOTIATION GAME

To investigate the role of beliefs about others in human reasoning, we require certain elements in a multiagent environment. First, we require a domain where agents’ preferences matter for decision making. The domain should provide the possibility for agents to reason strategically about each other, and this reasoning may entail the need for beliefs. Agents in the domain should be *situated* [15], such that behavior requires interaction within and with the environment; Gal et al. [6] show that situated task activity elicits stronger concern with social factors such as fairness, whereas the same underlying game presented in a more abstract payoff-matrix form engenders behavior more in line with rational Nash equilibrium play. Finally, we require the domain to be simple enough for modeling and decision making to be tractable. The Colored Trails (CT) framework [12] meets our requirements.

Using the CT framework, we construct a negotiation game in which players must negotiate with each other to obtain resources needed to complete a task. The player we are interested to model is called a *proposer*; a proposer formulates an offer to exchange resources with another player who is called the *responder*. The responder may also receive an outside offer to exchange resources. The responder can accept only one offer, either the proposer’s or some outside offer, or the responder can reject all received offers. Our domain can be viewed as a general model for one-shot negotiation that is situated in a particular task.

Our CT game is played on a 4x4 board of colored squares; each square is one of five colors. Each player 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 a role in the game (i.e., the proposer, responder, or the originator of an outside offer) and given a random assortment of chips. Each player knows the chips that she possesses and the state of the board (board colors, goal location, locations of all player pieces). The proposer also knows

the chips possessed by the responder, but not by any other player that may make an outside offer; this is a source of uncertainty in the game. The responder knows the chips possessed by all other players. The proposer is allowed to exchange chips only with the responder. Any redistribution of chips between the 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 space of possible proposals depends upon the specific chips possessed by the proposer and responder, and may range in size from approximately forty to four hundred. The responder then chooses either to accept the proposer’s offer, accept an outside offer, or decline all offers.

After the responder’s decision is made, the CT system 1) informs the proposer of the outcome, 2) executes any accepted proposal, and 3) automatically moves all players’ pieces to obtain the maximal possible score for each player, given the chips possessed after the negotiation. Landing onto the goal square earns a player 100 points; a player unable to reach the goal pays a penalty of 25 points for each square she is away from the goal. Each chip not used for moving earns a player 10 points.

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. Finally, because the responder can accept no more than one proposal, there exists a competitive relationship between the proposer and the outside offer. The behavior of the proposer explores the tension between fulfilling its own utility function and that of the responder in the face of unknown competition. We are interested to explore whether and how the proposer reasons by using beliefs about the responder’s preferences.

2.1 Example Game

To make our game more concrete, we provide the following example. Say that the goal square happens to be blue; thus, for a player to reach the goal, the player must possess a blue chip. Next, say that the proposer’s initial position is one square away from the goal; the proposer possesses four chips, but the proposer lacks a blue chip. Thus, regardless of what other chips the proposer has, it makes no sense for the proposer to use chips to move, since the proposer is already as close to the goal as it can currently get. Consequently, the best score the proposer can obtain with the chips it currently has is $40 - 25 = 15$ (10 points for each unused chip minus the penalty for being one square away). Say that the responder’s initial position is two squares away from the goal; the responder also has four chips, and while the responder has one blue chip, it lacks chips for the squares that happen to surround the goal square. Thus, the responder is also already as close to the goal as it can get. The best score the responder can obtain with the chips it currently has is $40 - 50 = -10$.

Let us say that the proposer asks for the responder’s blue chip in exchange for a chip that will allow the responder to move one square closer to the goal. If the responder accepts this offer, then the proposer’s score changes to $30 + 100 = 130$ (three unused chips plus the bonus for landing on the goal square); this is an increase of 115 points over the best score obtainable without exchanging chips. The responder’s score changes to $30 - 25 = 5$ (three unused chips minus the penalty

for being one square away), an increase of 15 points. Thus, a chip exchange may improve a player’s score even if it does not allow the player to reach the goal.

3. PLAYER MODELS

We are interested to model the behavior of proposer agents in our game. In particular, we model the proposer as maintaining beliefs about how the responder will behave and using these beliefs to reason about what offer to make. Thus, our proposer model contains parameters that facilitate reasoning about its own preferences as well as other parameters that facilitate reasoning with beliefs about the responder. Specifically, the proposer uses its beliefs about the responder to calculate the expected utilities of the possible offers it can make. In this section, we introduce our general model for representing a proposer’s beliefs about the responder.

Our models 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. 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. For example, in the game illustrated in Section 2.1, we consider a proposal where, from the point of view of the proposer, the self-benefit is SB = 115 and other-benefit is OB = 15.

Thus, let each proposal O be represented by a vector of feature values $O = \langle \text{SB}, \text{OB} \rangle$; let $\mathbf{w} = \langle w_{\text{SB}}, w_{\text{OB}} \rangle$ and $\mathbf{v} = \langle v_{\text{SB}}, v_{\text{OB}} \rangle$ be vectors of feature weight parameters. The parameters in \mathbf{w} are those which the proposer uses to reason about its own preferences; the parameters in \mathbf{v} are those which the proposer uses to reason about the responder’s believed preferences. Preference is expressed by a utility function $U : \mathcal{O} \rightarrow \mathbb{R}$ on the space \mathcal{O} of offers, which computes a linear combination of feature values using a set of weights.

Let ϕ denote the *status-quo*, which for the responder represents the option of rejecting the proposer’s offer as well as the outside offer, and for the proposer represents the proposal that no resources change hands. Note that $U(\phi) = 0$, since SB = OB = 0.

Since not all humans will likely share the same preferences and beliefs about others’ preferences, we use mixture models to cluster human play into different behavioral types. Let ρ^{s_i} be the proportion of proposers of type s_i . A proposer of type s_i uses the utility function U^{s_i} with weight vector \mathbf{w}^{s_i} to reason about its own preferences:

$$U^{s_i}(O) = w_{\text{SB}}^{s_i} \cdot O_{\text{SB}} + w_{\text{OB}}^{s_i} \cdot O_{\text{OB}}. \quad (1)$$

A proposer of type s_i may believe that different types of responders exist. Let ρ^{s_i, t_j} be the proportion of responders of type $\{s_i, t_j\}$ that a proposer of type s_i believes to exist. A proposer of type s_i uses the utility function U^{s_i, t_j} with weight vector \mathbf{v}^{s_i, t_j} to reason about the preferences it believes a responder of type $\{s_i, t_j\}$ has:

$$U^{s_i, t_j}(O) = v_{\text{SB}}^{s_i, t_j} \cdot O_{\text{SB}} + v_{\text{OB}}^{s_i, t_j} \cdot O_{\text{OB}}. \quad (2)$$

This is different from other work [8, 7] because the type $\{s_i, t_j\}$ of the responder is embedded in the type s_i of the proposer, who is holding a belief about the preferences of the responder.

Humans select offers non-deterministically; we are prone to make errors. Further, since our models will not be per-

fect, 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 multinomial logit function. The probability that a responder of type $\{s_i, t_j\}$ accepts an offer O is

$$\Pr(\text{accept} | O, \phi, \{s_i, t_j\}) = \frac{e^{U^{s_i, t_j}(O)}}{e^{U^{s_i, t_j}(O)} + e^{U^{s_i, t_j}(\phi)} + e^z}. \quad (3)$$

Since we do not know the outside offer, we cannot compare with the utility of a specific offer; rather, we use a parameter z to represent the believed utility of a generic, unknown outside offer. Taking an expectation over all responder types $\{s_i, t_j\}$ that a proposer of type s_i believes to exist gives

$$\Pr(\text{accept} | O, \phi, s_i) = \sum_j^{\text{Responder}} \Pr(\text{accept} | O, \phi, \{s_i, t_j\}) \cdot \rho^{s_i, t_j}. \quad (4)$$

For the proposer, let $\mathcal{O} = \{O_1, \dots, O_M\}$ be the set of possible offers, where M varies from game to game due to the particulars of the game state. The probability that a proposer of type s_i will select the m -th proposal in \mathcal{O} is a function of the expected utility of O^m :

$$EU^{s_i}(O^m) = U^{s_i}(O^m) \cdot \Pr(\text{accept} | O^m, \phi, s_i). \quad (5)$$

The expected utility to the proposer of an offer O is the proposer’s utility for O times the probability, according to the proposer’s beliefs, that the responder will accept O . With expected utilities in hand, the probability that a proposer of type s_i will select the m -th proposal in \mathcal{O} is

$$\Pr(\text{selected} = O^m | \mathcal{O}, s_i) = \frac{e^{EU^{s_i}(O^m)}}{\sum_k e^{EU^{s_i}(O^k)}}. \quad (6)$$

Taking an expectation over proposer types gives

$$\Pr(\text{selected} = O^m | \mathcal{O}) = \sum_i^{\text{Proposer}} \Pr(\text{selected} = O^m | \mathcal{O}, s_i) \cdot \rho^{s_i} \quad (7)$$

The general model’s structure, for two proposer types and two embedded responder types in each proposer type, is illustrated in Figure 1.

4. HYPOTHESES AND MODEL VARIATIONS

We explore four hypotheses with our model. Crucial to being able to test these hypotheses is the fact that our models are identifiable, meaning that no two sets of parameter settings can yield the same model behaviors over all possible feature inputs. Section 5 provides our proof of identifiability.

Hypothesis 1 The first hypothesis (H1) is that proposer beliefs about responders’ preferences are correct, and agree with the actual observed behavior of human responders. To explore this possibility, we pre-learn a model of responder behavior directly from *responder* data; we regard this responder model to have the correct values for parameters $v_{\text{SB}}, v_{\text{OB}}$, and z , as well as the correct mixture distribution over responder types. (In building the responder model, we determined that a mixture model with two responder types fits the responder data best.) We then fix these responder

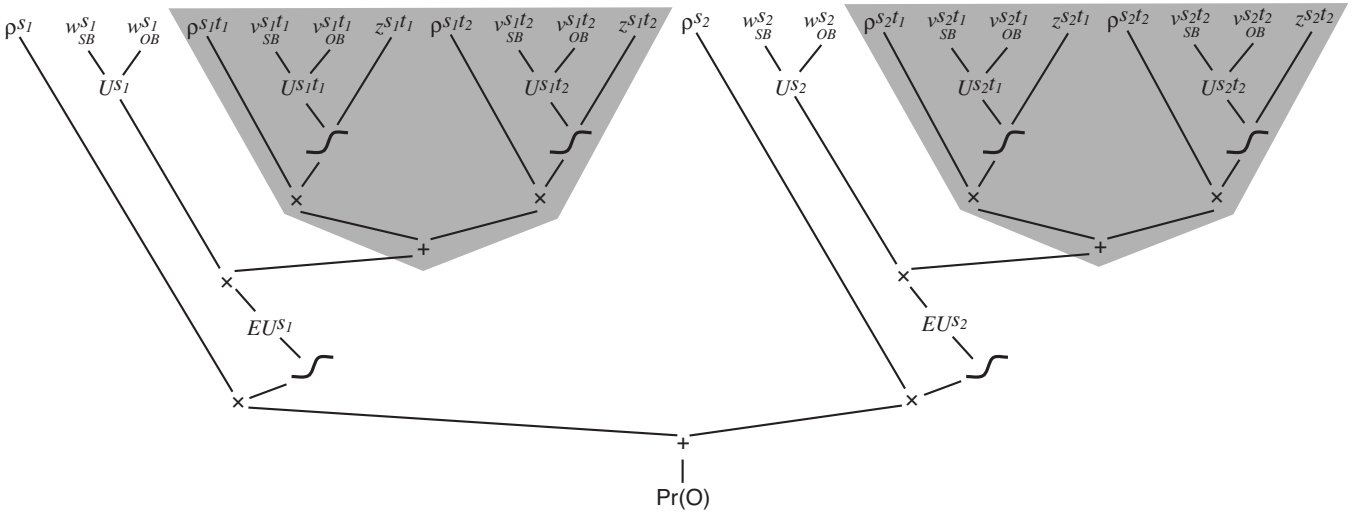


Figure 1: Structure of general model with two proposer types. Each proposer type has an embedded responder model (shaded portion) with two types. Model parameters are located at the leaves of the diagram.

parameters in our general model (Figure 1) and learn only the proposer parameters $w_{SB}^{s_i}$, $w_{OB}^{s_i}$, and ρ^{s_i} for each proposer type s_i from proposer data.

Hypothesis 2 The second hypothesis (H2) is that proposer beliefs about responders' preferences are incorrect. To explore this possibility, we learn all of the parameters of our general model simultaneously; now, the belief parameters v_{SB} , v_{OB} , and z are learned from *proposer* data—not responder data. If we obtain a better fit of proposer data under this approach than in H1, then proposer beliefs are incorrect. Our general model is shown in Figure 1. The leaves indicate the model parameters. We have two proposer types, s_1 and s_2 . Each proposer type has an embedded responder model that is used to calculate expected utility EU. Each responder model is itself a mixture of two responder types.

Hypothesis 3 The last two hypotheses concern patterns of belief. Our third hypothesis (H3) is that all proposer types believe the same thing about responders. Thus, the parameter value $v_{SB}^{s_i}$ is the same for all proposer types s_i ; similarly, $v_{OB}^{s_i}$ is the same for all proposer types s_i , as is z^{s_i} . Despite the shared beliefs, the different proposer types may still have different utility functions to express their own preferences. The structure of this model is illustrated by Figure 2. We learn all parameters of this model from proposer data.

Hypothesis 4 Our fourth hypothesis (H4) is that people believe other people are like themselves. Specifically, each proposer type believes that the responder has the same utility function as itself. In this case, each proposer type s_i believes that there exists one responder type $\{s_i, t_1\}$ and that $v_{SB}^{s_i, t_1} = w_{SB}^{s_i}$ and $v_{OB}^{s_i, t_1} = w_{OB}^{s_i}$. This structure is illustrated by Figure 3. We learn all parameters in this model from proposer data.

5. IDENTIFIABILITY

Both the proposer's preferences and beliefs about the responder have an effect on proposer behavior. Can we untangle the effects of one from the other? In other words, are the subjective beliefs proposers have about responders identifiable? It is not obvious that they are. Nevertheless, in this section, we show that, at least theoretically, preferences can

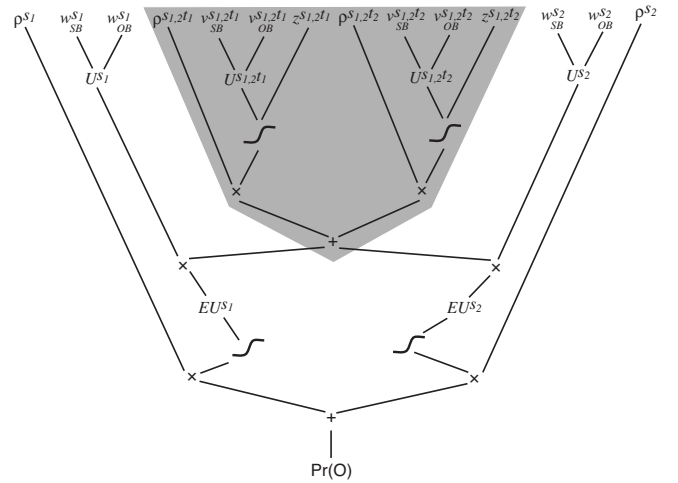


Figure 2: Structure of constrained model where different proposer types share beliefs about the responder types. Embedded responder model is shaded.

be distinguished from beliefs in each proposer type s^i . We are only interested to show this within one proposer type. It may be possible that different proposer mixtures produce the same behavior; nevertheless, in all models, the effects of the proposer's preferences, her beliefs about the responder's preferences, and her beliefs about the outside offer can be distinguished. No more than one set of parameter values can produce the same model behavior over all possible inputs.

To begin, let us use a minimal model with one responder type, and one generic proposal feature O_x and associated parameters w , v , and z . Our proof requires only that we look at the proposer's expected utility (5), which is

$$\begin{aligned}
 EU(O) &= (w \cdot O_x) \cdot \frac{e^{v \cdot O_x}}{e^{v \cdot O_x} + e^{v \cdot \phi_x} + e^z} \\
 &= (w \cdot O_x) \cdot \frac{1}{1 + e^{-v \cdot O_x} + e^{z - v \cdot O_x}}.
 \end{aligned} \tag{8}$$

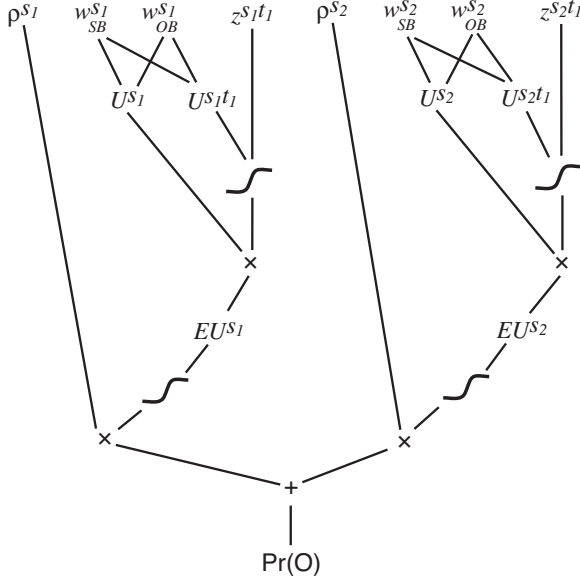


Figure 3: Structure of constrained model where each proposer type believes the responder type has the same preferences as itself. Notice shared weights.

Let us first consider parameter values $w > 0, v > 0$. Let us assume that two parameter sets $\langle w_0, v_0, z_0 \rangle$ and $\langle w_1, v_1, z_1 \rangle$, where $w_0 \neq w_1, v_0 \neq v_1$, or $z_0 \neq z_1$, produce the same value in (8) for all feature values O_x . Let h denote the probability that the responder will accept an offer (3). Since (3) is a sigmoid function, $\lim_{O_x \rightarrow \infty} h(\text{accept}|O, \phi) \rightarrow 1.0$. Thus, for sufficiently large values of O_x , Equation (8) approximates a linear equation with slope w . Clearly, two different values of w will create different lines with different slopes for large O_x . Thus, two parameter sets with different values for w cannot produce identical behavior in (8) over all O_x ; as a result, it must be that $w_0 = w_1$. This reasoning holds even when we have multiple responder types: for any set of types, with each type having some non-zero parameter, we can construct an offer that will take the types' sigmoid functions to some combination of 0.0 and 1.0 outputs; if a type has all-zero parameters, then it already outputs a constant. At this point, the mixture outputs a constant between 0.0 and 1.0. In either case, the expected utility function becomes a line.

Next, we examine the denominator of the exponent in (8). Parameters v and z interact in the denominator, but $z-v \cdot O_x$ is a line. No two pairs of $\langle v_0, z_0 \rangle$ and $\langle v_1, z_1 \rangle$ can describe the same line, unless $v_0 = v_1$ and $z_0 = z_1$; thus, the effects of parameters v and z can be distinguished from each other.

Because h is a sigmoid function, the parameters v and z together determine the width of the domain interval over which h spans the range interval $[0 + \epsilon, 1 - \epsilon]$. Given different v and z , we will obtain different transitions in the sigmoid. Therefore, it must also be that $v_0 = v_1$ and $z_0 = z_1$. This completes our proof for $w, v > 0$. For parameter values where one or both of w or v is less than zero, our reasoning is identical. When $v < 0$, we instead have $\lim_{O_x \rightarrow -\infty} h(\text{accept}|O, \phi) \rightarrow 1.0$. When $w < 0$, we instead have a negative slope.

Now let us assume that proposals have multiple features. For any two vectors \mathbf{w}_0 and \mathbf{w}_1 , the proposer's utility func-

tion U can only produce the same values over all possible offers if the corresponding elements of \mathbf{w}_0 and \mathbf{w}_1 are identical. If \mathbf{w}_0 and \mathbf{w}_1 are not identical, then we can construct an offer with value zero at each feature except for where the corresponding weights in \mathbf{w}_0 and \mathbf{w}_1 are different, and we will obtain different utilities. Identical reasoning applies to the responder feature vector \mathbf{v} and z .

6. LEARNING

Models are trained by gradient descent. Let $g(\text{selected} = O^*|\mathcal{O})$ be the probability that a proposer model assigns to the proposal O^* that was actually selected by a human proposer, given the set of options \mathcal{O} . The error function F that we minimize measures negative log likelihood of the data (N instances), given a model:

$$F = - \sum_{n=1}^N \ln \left(g(\text{selected} = O^{*n}|\mathcal{O}^n) \right). \quad (9)$$

The derivative of the error function with respect to some model parameter $w_l^{s_i}$ (or $v_l^{s_i, t_j}, z^{s_i, t_j}$, or ρ^{s_i, t_j}) is

$$\frac{\partial F}{\partial w_l^{s_i}} = - \sum_{n=1}^N \frac{\frac{\partial g}{\partial w_l^{s_i}}}{g(\text{selected} = O^{*n}|\mathcal{O}^n)}. \quad (10)$$

Equation (10) requires that we further calculate the partial derivative of function g . The full set of learning equations is given in [5]. Here, we provide illustrative examples. The partial derivative of g with respect to the proposer's feature weights w^{s_i} is

$$\begin{aligned} \frac{\partial g(\text{selected} = O^*|\mathcal{O})}{\partial w_l^{s_i}} &= \\ &g(\text{selected} = O^*|\mathcal{O}, s_i) \cdot \\ &\left(W(O^*) - \sum_k W(O^k) \cdot g(\text{selected} = O^k|\mathcal{O}, s_i) \right), \end{aligned} \quad (11)$$

where

$$W(O) \equiv O_l \cdot \overset{\text{Responder}}{\text{Pr}}(\text{accept}|O, \phi, s_i) \quad (12)$$

The partial derivative of g with respect to the proposer's beliefs about the responder's feature weights v^{s_i, t_j} is

$$\begin{aligned} \frac{\partial g(\text{selected} = O^*|\mathcal{O})}{\partial v_l^{s_i, t_j}} &= \\ &g(\text{selected} = O^*|\mathcal{O}, s_i) \cdot \\ &\left(Q(O^*) - \sum_k Q(O^k) \cdot g(\text{selected} = O^k|\mathcal{O}, s_i) \right) \end{aligned} \quad (13)$$

where

$$\begin{aligned} Q(O) &\equiv \\ &U^{s_i}(O) \cdot \overset{\text{Responder}}{\text{Pr}}(\text{accept}|O, \phi, \{s_i, t_j\}) \cdot \rho^{s_i, t_j} \cdot \\ &\left(O_l - O_l \cdot \overset{\text{Responder}}{\text{Pr}}(\text{accept}|O, \phi, \{s_i, t_j\}) \right) \end{aligned} \quad (14)$$

To calculate gradients for the embedded responder parameters, we compute $Q(O)$. The first line on the right-hand side of (14) is the contribution a responder of type $\{s_i, t_j\}$ makes to the expected utility of a proposer of type s_i ; the second line is an adjustment factor equal to the difference, given the responder type, between the feature value O_l if accepted and its expected value.

7. RESULTS

We ran two sessions of human-subjects trials to collect data on how humans play our negotiation game. A total of 69 subjects were recruited from a large pool (one to two thousand) maintained by the Harvard Business School (HBS) for their experimental economics research. Subjects reflected a diversity in age (min 18, max 58), gender, and profession. Subjects played 15 rounds of our game, with over half of their total compensation determined by the scores they accumulated over the rounds. A subject’s total score after all rounds played was converted to a dollar amount, with each point corresponding to one cent, e.g., a score of 1250 points gives \$12.50 USD (see Section 2 for details on how our game is scored). Subjects received a \$10 USD show-up fee. Subjects were free to leave at any point during the experiment, but received a bonus of 50 score points (i.e., \$0.50 USD) for each completed round. Average compensation was \$25 USD.

Our experiments took place at the HBS Computer Lab for Experimental Research. Subjects occupied a single room. Each subject operated a computer terminal with a privacy screen. Subjects could not see each other, and were randomly and anonymously matched to interact in each round. To emphasize that they were playing a sequence of one-shot games and not an iterated game, subjects performed an unrelated activity between rounds. We collected 536 data instances for training our proposer models; each data instance represents one proposer decision (along with a corresponding responder decision).

We used cross-validation to determine how well the model variations described above fit human proposer data; to speed training, we initialized all models with weights from our model of Hypothesis 1, which was trained first on responder data to learn beliefs (i.e., \mathbf{v} , z , and ρ^{s_i, t_j}), then proposer data to learn preferences (i.e., \mathbf{w} and ρ^{s_i}). Results are summarized in Table 1. Columns labeled 1-6 summarize model behavior for our four hypotheses along with two other models, one that behaves randomly and another model that is *reflexive* (i.e., does not explicitly maintain beliefs about the responder). Rows labeled 1-7 present different measurements of model behavior; each row is an average over five test sets. The models listed in the table are in order of decreasing negative log likelihood (Row 1).

7.1 Negative Log Likelihood

Row 1 gives the average negative log likelihood (NLL) of the data, given the model, divided by the size of the test data; lower numbers indicate better fit. Our learning process sought to minimize this measurement. The best fit is obtained under the model constructed to test hypothesis H2, which asks whether proposers’ beliefs about the responders are incorrect. Since Hypothesis 2 is investigated by using the general model, it should fit our data no worse than any other model, and this is indeed the case.

Next, we have hypotheses H3 and H1, which ask whether proposers have shared beliefs about responders, and whether proposers have correct beliefs about responders, respectively. The model used to investigate Hypothesis 3 (Figure 2) is a generalization of the model used to investigate H1 (correct beliefs imply shared beliefs). Thus, the model in Figure 2 should fit our data no worse than the model in H1, which is also what our data show.

The difference in fit between H2 and H1 is small but has a

p -value of 0.0707 (single-tailed t-test). Thus, there is a suggestion that proposers have slightly incorrect beliefs about responders. The p -value when comparing the fit of H2 and H3 is 0.1418. This is weaker evidence that proposers do not share the same beliefs.

Hypothesis H4 asks whether proposers believe that responders share the same preferences as themselves. This hypothesis clearly has the worst fit of our hypotheses and so appears false. Column 3 of Table 1 gives the performance of a reflexive model of proposer behavior; this model makes a decision based only upon the proposal options it has, and does not explicitly reason about the responder at all. The reflexive proposer model is a mixture model identical to our general model, except that it contains none of the parameters that relate to the responder. Interestingly, the model of hypothesis H4 fits our data worse than the reflexive model. That is, having a model with poor beliefs can be worse than a model with no beliefs at all.

Finally, column 1 of Table 1 gives the performance we may expect from a model that uses a uniform distribution over all proposals in \mathcal{O} , and so corresponds to random guessing. All of our models clearly fit our data better than random guessing.

7.2 Other Measurements of Behavior

While our learning procedure was to minimize negative log likelihood, one problem with this measurement is that it does not provide a very intuitive reflection of how good a model is. Rows 2-7 in Table 1 present a number of other ways to measure the behavior of the models we examine. Row 2 indicates the highest probability assigned by the models, on average, to an option \mathcal{O} ; higher values indicate a more peaked distribution, which may suggest a higher confidence. Here, we see that the peaks increase in height as the models improve their fit to the data with respect to NLL.

Row 3 indicates the proportion of options in \mathcal{O} that are given a higher probability by the model than the option actually chosen by the human in the data instance; zero would indicate that the actually chosen option was always most preferred by the model. This figure is not applicable to the random model, since each option is equally preferred. Here, we see that columns 2-5 show improvement in this measurement corresponding to improved fit with respect to NLL. The general model (column 6), however, has the third-best performance in this regard. The p -value obtained when comparing the proportion measurements of H2 and H3 is 0.2222, giving fairly weak evidence that the model of H3 actually fits better than H2. The difference between H2 and H1 is not statistically significant.

When the option in \mathcal{O} that is most preferred by a model is different from the option selected by a human, we can further see how dissimilar these two choices are. Let O^* be the proposal in \mathcal{O} actually made by a human proposer in a game; let \hat{O} be the proposal in \mathcal{O} that is most preferred by a proposer model. Row 4 of Table 1 gives the mean absolute difference between O_{SB}^* and \hat{O}_{SB} ; row 5 gives the mean absolute difference between O_{OB}^* and \hat{O}_{OB} . These measurements are not applicable to the random model, since it prefers all options in \mathcal{O} equally. Moving left to right in columns 2-4, we clearly see the differences in benefits shrink as the models improve their fit with respect to NLL. The differences in benefits in columns 4-6 are much smaller in magnitude. The difference between H2 and H3 with respect to $|\Delta_{OB}|$ (row 5) is not

Table 1: Fit of learned models to data test-sets.

	1	2	3	4	5	6
	Random	H4	Reflexive	H1	H3	H2
1 Negative Log Likelihood	5.0106	4.4991	4.4220	4.0873	4.0803	4.0698
2 Max Pr	0.0082	0.0334	0.0508	0.1566	0.1610	0.1834
3 Proportion	N/A	0.2658	0.2290	0.1922	0.1906	0.1928
4 $\overline{ \Delta SB }$	N/A	50.2016	48.4586	33.6408	33.1160	33.6422
5 $\overline{ \Delta OB }$	N/A	66.8316	55.4660	49.0256	49.5312	49.6272
6 $\overline{E[\Delta SB]}$	50.1894	34.0936	27.5698	25.2460	25.5444	25.4482
7 $\overline{E[\Delta OB]}$	45.0558	39.1614	41.6058	42.5416	42.6192	42.3976

statistically significant; with respect to $\overline{|\Delta SB|}$ (row 4), we obtain a p -value of 0.1598—a fairly weak indication that the model in H3 fits better than H2 with respect to $\overline{|\Delta SB|}$. The difference between H2 and H1 with respect to $\overline{|\Delta SB|}$ is not statistically significant; with respect to $\overline{|\Delta OB|}$, we obtain a p -value of 0.0724, which suggests that H1 does fit better than H2 on this measurement.

Rows 6 and 7 of Table 1 give the average expected differences in benefits between O^* and \hat{O} . For these measurements, we use a model’s entire distribution over \mathcal{O} to calculate an expectation, rather than look only at the model’s most preferred option. For $\overline{E[|\Delta SB|]}$ (row 6), we see clear improvement as we move from column 1 to 4; differences amongst the last three columns are again small in magnitude. With respect to $\overline{E[|\Delta OB|]}$ (row 7), we are surprised to find that fit becomes worse as we move from column 2 to 4; again columns 4-6 show very similar performance.

7.3 Learned Belief Weights

Table 2 gives some of the actual parameter values learned in our models. The column labeled ‘H1’ gives the weights of the correct responder model, which was learned directly from responder data. This model is a mixture of two responder types, t_1 (rows 1–4) and t_2 (rows 5–8). The next two columns give the weights that concern proposer beliefs about the responder, from a representative learning run of our general model. Here, we have two proposer types, s_1 and s_2 . The column for type s_1 , for example, gives parameters representing the beliefs s_1 has about the responder. Thus, the entry in row 1 of this column tells us the probability with which proposer type s_1 expects to encounter a responder of type $\{s_1, t_1\}$.

Rows 1 and 5 give the probabilities with which the responder types are believed to be encountered by proposers. Rows 2 and 6 give the believed SB weights for the responder types; rows 3 and 7 do the same for OB. Rows 4 and 8 give the believed generic utility for the outside offer for each responder type.

The most dramatic changes concern the mixture probabilities ρ^{t_1} and ρ^{t_2} . The correct responder model identifies two responder types, appearing with frequencies of about 0.67 and 0.33, respectively. After learning the general model to test hypothesis H2, we find some dramatic changes in the beliefs. Proposer type s_1 believes that the two responder types are almost evenly distributed in the population, rather than 0.67 and 0.33. Much more extreme is proposer type s_2 , which believes that about 97% of responders are of type t_1 . The distribution of the two proposer types, according to the learned general model, is about $\rho^{s_1} = 0.74$ and $\rho^{s_2} = 0.26$. Thus, neither proposer type has correct beliefs

about the responder distribution. Beyond the beliefs about the responder distribution, we also find some of the beliefs about responder preferences to shift. For example, proposer type s_2 believes that type t_1 responders care less about their own self benefit (row 2) and more about depriving proposers of benefit (row 3) than type t_1 actually does.

Despite the parameter changes shown in Table 2, rows 4–7 of Table 1 show that the average behavior of the learned general model (H2) is virtually identical to that of the correct model (H1). When we contrast these models on a game-by-game basis, however, we are able to find some substantive differences in behavior. In particular, we find games for which the preference orderings of the two models is different for certain proposal options. That is, where the correct model prefers option A over B, the general model prefers B over A. Such differences between the models in H1 and H2 are certainly more meaningful than the very small changes in average behavior shown in Table 1.

Table 2: Learned Belief Weights of General Model (Representative Run).

		H1	H2 (Type s_1)	H2 (Type s_2)
1	ρ^{t_1}	0.6734	0.4913	0.9673
2	$v_{SB}^{t_1}$	2.5243	2.4262	1.9768
3	$v_{OB}^{t_1}$	-0.1580	-0.2755	-0.3935
4	z^{t_1}	-2.1229	-2.1146	-2.1052
5	ρ^{t_2}	0.3266	0.5087	0.0327
6	$v_{SB}^{t_2}$	0.2932	0.3397	0.2481
7	$v_{OB}^{t_2}$	-0.5269	-0.4001	-0.4653
8	z^{t_2}	1.7508	1.7193	1.7398

7.4 Discussion

We trained our models to minimize negative log likelihood. Improvements in model performance with respect to NLL are often, but not always, accompanied by improvements with respect to other measures of performance. Exceptions are usually found where differences in NLL are small, but row 7 of Table 1 gives an example where substantial improvements in NLL do not necessarily lead to better fit with respect to $\overline{E[|\Delta OB|]}$. An alternative error function that directly measures differences in benefits might prove more effective in our domain.

Nevertheless, such ambiguities in the data are small in extent. Hypothesis H4 is convincingly shown to be false; thus, proposers do *not* believe that responders use the same utility function as themselves. Measurements of NLL and detailed examination of model behavior in games suggest that proposers have slightly incorrect beliefs about responder preferences.

Quite aside from the human data obtained in our particular negotiation game, we have shown how our models can be used to explore the effects of preferences and beliefs on human decision making in a multiagent setting. We have shown that our models are identifiable, meaning that no two different parameters settings can yield the same model behavior over all possible inputs. Our approach gives a way to detect false beliefs that are reflected in data. The features used by our models are quite generic and should easily work in other negotiation settings. Thus, testing in different domains may give sharper distinctions between hypotheses H1, H2, and H3. This is beneficial future work.

Gal et al. [8] studied a simple two-player negotiation game with human subjects and found that proposers often made offers to the responder that were more generous than what the responder would actually have required to accept the offer. Our modeling framework points to a way to understand the origin of this gap between what the responder requires and what the proposer gives. Our current models allow us to consider how the proposer's decision making is influenced by the combination of its own preferences and beliefs about the responder. Thus, it might be the case that the proposer is generous to the responder because the proposer's utility function prefers generous offers. Alternatively, it might be the case that the proposer erroneously believes that the responder will not accept offers below a certain threshold. Our model allows us to disentangle these two influences. A third possibility, however, is that the proposer is *risk averse* and makes a generous offer to avoid uncertainty about whether the responder will accept. We are now expanding our modeling approach to also take this factor into account.

8. CONCLUSIONS

Human decision making in multiagent scenarios is the product of several factors, such as individual preference, beliefs about others' preferences, and beliefs about how others interact with third parties. We have introduced the first model of human reasoning that allows us to identify and distinguish these factors; we show that our model is identifiable. Using a simple multiagent negotiation game, we conduct human-subjects trials to obtain data about human reasoning. We then use cross validation to determine how well each of several variations of our general model fit our data; each model variation corresponds to a particular hypothesis we investigate. These hypotheses ask whether people form correct beliefs about others' preferences or not, and whether one's beliefs relate to one's preferences in particular ways. We find that, in our negotiation game, players form slightly incorrect beliefs about each other's preferences. Our results have implications for agent designers who want to build computer agents that interact with people in strategic situations.

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.

9. REFERENCES

- [1] T. Bosse, Z. Memon, and J. Treur. A two-level BDI-agent model for theory of mind and its use in social manipulation. In *AISB 2007 Workshop on Mindful Environments*, 2007.
- [2] J. T. Cacioppo, P. S. Visser, and C. L. Pickett, editors. *Social neuroscience: People thinking about people*. MIT Press, 2005.
- [3] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.
- [4] M. Davies and T. Stone, editors. *Folk Psychology: The Theory of Mind Debate*. Blackwell Publishers, 1995.
- [5] S. G. Ficici and A. Pfeffer. Simultaneously modeling humans' preferences and their beliefs about others' preferences. Technical Report TR-02-07, Harvard University School of Eng. and Applied Sciences, 2007.
- [6] Y. Gal, B. J. Grosz, A. Pfeffer, S. M. Shieber, and A. Allain. The influence of task contexts on the decision-making of humans and computers. In *Proc. Sixth International and Interdisciplinary Conference on Modeling and Using Context*, 2007.
- [7] Y. Gal and A. Pfeffer. Predicting people's bidding behavior in negotiation. In *AAMAS*, 2006.
- [8] Y. Gal, A. Pfeffer, F. Marzo, and B. J. Grosz. Learning social preferences in games. In *National Conference on Artificial Intelligence (AAAI)*, 2004.
- [9] P. Gmytrasiewicz and E. H. Durfee. Rational coordination in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 3(4):319–350, 2000.
- [10] R. Gordon. Folk psychology as simulation. *Mind and Language*, 1:158–171, 1986.
- [11] J. Gratch and S. Marsella. Evaluating a computational model of emotion. *Autonomous Agents and Multi-Agent Systems*, 11(1):23–43, 2005.
- [12] B. J. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. In *AAMAS*, 2004.
- [13] S. Hurley. The shared circuits hypothesis: A unified functional architecture for control, imitation, and simulation. In S. Hurley and N. Chater, editors, *Perspectives on Imitation: From Neuroscience to Social Science*, volume 1. MIT Press, 2004.
- [14] S. Kraus. *Strategic Negotiation in Multiagent Environments*. MIT Press, 2001.
- [15] C. Lueg and R. Pfeifer. Cognition, situatedness, and situated design. In *Conf. on Cognitive Tech.*, 1997.
- [16] S. Marsell, D. Pynadath, and S. Read. Psychsim: Agent-based modeling of social interactions and influence. In *ICCM 2004*, 2004.
- [17] M. Rabin. Psychology and economics. *Journal of Economic Literature*, 36:11–46, 1998.
- [18] J. M. Vidal and E. H. Durfee. Recursive agent modeling using limited rationality. In *International Conference on Multi-Agent Systems*, 1995.
- [19] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 2000.