Noname manuscript No.
(will be inserted by the editor)

Strategic Advice Provision in Repeated Human-Agent Interactions

Amos Azaria · Ya'akov (Kobi) Gal · Sarit Kraus · Claudia V. Goldman

the date of receipt and acceptance should be inserted later

Abstract This paper addresses the problem of automated advice provision in scenarios that involve repeated interactions between people and computer agents. This problem arises in many applications such as route selection systems, office assistants and climate control systems. To succeed in such settings agents must reason about how their advice influences people's future actions or decisions over time. This work models such scenarios as a family of repeated bilateral interaction called "choice selection processes", in which humans or computer agents may share certain goals, but are essentially self-interested. We propose a Social agent for Advice Provision (*SAP*) for such environments that generates advice using a social utility function which weighs the sum of the individual utilities of both agent and human participants. The SAP agent models human choice selection using hyperbolic discounting and samples the model to infer the best weights for its social utility function. We demonstrate the effectiveness of SAP in two separate domains which vary in the complexity of modeling human behavior as well as the information that is available to people when they need to decide whether to accept the agent's advice. In both of these domains, we evaluated SAP in extensive empirical studies involving hundreds of human subjects. SAP was compared to agents using alternative models of choice selection processes informed by behavioral economics and psychological models of decision-making. Our results show that in both domains, the SAP agent was able to outperform alternative models. This work demonstrates the efficacy of combining computational methods with behavioral economics to model how people reason about machine-generated advice and presents a general methodology for agent-design in such repeated advice settings.

Amos Azaria
Department of Machine Learning, Carnegie Mellon University, Pittsburgh PA
E-mail: azariaa@cs.cmu.edu

Ya'akov Gal
Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel
E-mail: kobig@bgu.ac.il

Sarit Kraus
Department of Computer Science, Bar-Ilan University, Ramat Gan, Israel
E-mail: sarit@cs.biu.ac.il

Claudia V. Goldman
General Motors Advanced Technical Center, Herzliya, Israel
E-mail: claudia.goldman@gm.com

1 Introduction

Computer systems are increasingly being deployed in platforms that enable interactions with people as well as with other computer agents. Many of these scenarios require computer agents to repeatedly generate advice to their human users about what choices to make. Such settings arise in application domains like coaching and rehabilitation, route-navigation and climate control systems. Although users and systems in these domains share some goals, such as completing the user's tasks, their goals may not overlap completely. For example, consider an environmentally-conscious route-selection system that advises drivers about their daily commuting routes. The system possesses information about traffic jams and road conditions that is not available directly to the driver who makes the decision which route to take. Both system and driver wish to reach the destination safely. However, the driver may prefer quicker routes, while the system cares also about reducing the driver's emissions to protect the environment or about some information it expects to collect from the driver (such as traffic conditions or patrol-car location). Another example involves a decision-support system for doctors for the purpose of recommending medical treatments to patients. The system may have knowledge of a new highly effective antibiotic, but will suggest a more traditional treatment for the patient in order to alleviate drug resistance in the population.

The focus of this paper is on the design of advice provision strategies for computer agents that repeatedly interact with people. We model these interactions as a family of repeated games of incomplete information called *choice selection processes* comprising a human and a computer player. Both of the participants in a choice selection process are self-interested. The computer possesses private information regarding the states of the world which influences both participants' rewards; this information is not fully known to the person. In our example, this corresponds to the person not knowing the traffic conditions in all of the roads. At each round, the computer suggests one of several choices to the person, and the person then selects his or her choice, which may or may not correspond to the computer's suggestion. The choice of the person affects the reward for both the person and the computer agent. The performance of both participants is measured by their aggregate reward that is accumulated over time.

For an agent to be successful in such interactions, it needs to generate advice that is likely to be accepted by people, while still fulfilling the agent's individual goals. The design of such advice provision strategies is computationally challenging for several reasons. First, the agent needs to reason about the potential effect of the proposed advice and resulting user-action on its future interactions. For instance, suggesting routes that are significantly more beneficial to the agent than to the person may cause the person to ignore future recommendations or to turn off the advice system. Second, it is difficult to predict people's behavior over time for different types of advice strategies, because people affected by a variety of social and psychological factors [10]. For instance, some people may prefer certain routes over other routes due to past experience that was satisfactory and may be reluctant to adopt new, possibly preferable alternative routes. Lastly, people have been shown to discount the advice they receive from experts when they need to make strategic decisions [9,48]. Lastly, people may not be familiar with the agent's costs and benefits, as in the case of a driver that is interacting with a route recommendation system. Thus choice selection processes are not amenable to traditional game theoretic analysis.

To address these challenges, we designed several models of human behavior in choice selection processes that incorporated quantal response, exponential smoothing, and hyperbolic discounting theories taken from behavioral economics [21,24]. We estimated the parameters of these models using maximum likelihood techniques based on data consisting of

hundreds of instances of human play in choice selection processes. The best model found for the human decision making process was a combination of hyperbolic discounting and quantal response. We implemented an intelligent agent named Social agent for Advice Provision (*SAP*) that provides an advice that maximizes a social utility function which is a weighted sum of the agent and human's utilities. The SAP agent uses the human model and runs simulations of repeated human-agent interaction to identify the weights that maximizes the agent's utility over time.

The agent behavior was evaluated in extensive empirical studies using hundreds of human subjects in two types of selection processes that varied in complexity and the type of interaction used between computer and person. The first domain was analogous to a route selection task in which users needed to choose one of several possible commuting routes (from a set of candidates) for each day. The travel time and the fuel consumption of each road varied due to traffic, and was known to a computer agent (but not to the person). At each round, the computer suggested one of the routes to the person. The person's individual goal was to minimize travel time while the agent's individual goal was to minimize fuel consumption.

The second domain was analogous to a climate control task in which users needed to set the level of the climate control system deployed in a fictional car. The comfort level of the person depended on the level of the climate control system as well as environmental conditions such as the heat load of each day. While the person's individual goal depended both on its comfort level as well as the power consumption of the climate control system, the agent's goal was solely to minimize the energy consumption. The person may choose to partially follow the computer's advice, by selecting comfort levels that are close to, but not equal to the computer's suggestion. The computer needs to reason about this fact when it considers the effect of its advice on the person's behavior.

In both of the domains, we compared several alternative agent designs for providing advice to people. We used several candidate agent models. We tested the performance of an agent that approximated the optimal strategy based on a Markov Decision Process (MDP). We also tested the performance of three baseline strategies. The first of which provides no advice, the second provided the advice which was best for the user and the third totally ignored the user and provided advice which was optimal for the agent's individual goal. Finally, we compared all these agents to the SAP approach, which considered the costs for both agent and person when making suggestions. We evaluated these different agent designs in studies comprising hundreds of people that interacted with the system we developed on Amazon's Mechanical Turk [2]. In both of the domains, the SAP agent was consistently able to outperform all other agent strategies.

This work is first to design a computer agent for generating advice to people in repeated settings, and demonstrates the efficacy of using behavioral economic models when generating advice. This paper extends previous work in Azaria et al. [8] by formalizing people's advice provision behavior in a general way, as presented in Section 3 and used throughout the paper. This paper also extends previous work by adding an additional domain, the climate control system (which appears in Section 5, along with its extensive experimental evaluation. This new domain is substantially different (as described in Section 5), and also allows us to directly compare our agent's performance with that of an agent using a formal decision-theoretic approach (MDP).

The rest of this paper is organized as follows. The following section presents related work on advice provisions from the computational and social science disciplines. Section 3 presents the general selection process model. Sections 4 and 5 presents the route selection and climate control domains, show how we used selection processes to design the SAP

agent for each domain, and present empirical evaluation of the agent. Section 6 concludes with discussing some of the limitations of our approach and provides pointers for future work.

2 Related Work

Past work on advice provision spans the computational and social sciences disciplines.

Work in Multi-Agent Systems has provided several agent designs for providing advice to human users. One of these is the Pocket Negotiator [29] which provides advice to users about when to accept bids in closed bi-lateral negotiation settings. Rovatsos and Belesiotis [42] provide a formal model of advice taking in Multi-agent reinforement learning that is evaluated on simulated data.

Recommender systems [1] advise users to take certain actions, usually from a large set of actions. Users may benefit from such recommendations since they may have difficulties in estimating their utility from each alternative. Several works in recommendation systems have predicted rating behavior by users in order to best provide them with recommendations. (See Ricci et al. [40] for a review of this topic). Most works in this realm have only considered the utility of the system and have not modeled the user's reactions to the system's recommendations over time. Although several works do consider the utility of the system [37, 12, 13, 5] neither of them explicitly deal with repeated interactions. One exception is the work by Shani et al. [43], which uses a discrete-state MDP model to maximize the system utility function taking into account the future interactions with their users. However, the model they use does not explicitly consider the possible effects of providing advice that turns out to be bad (or good) to the user on the way the user will perceive future advice. In our work, we show that providing advice which turns out to be bad to the user causes the user to follow future advice less

It is well known that the way information is presented may have an impact on the human decision-making process. Rosenberg et al. [41] study the effect that photographs of political candidates have on voters' perception and show indeed that these images significantly affect their votes. Fenster et al. [17] design an agent which influences human decision-making in a conversational setting. The work studied an environment where the human had to select a location for a school. The agent interacted with the human and attempted to convince her to choose a certain location. The agent tries to convince the human about a location by providing examples for her to emulate, or by providing justifications for a certain choice.

Azaria et al. [6] designed a system that provides partial information to a user in order to encourage the user to take a certain action. This information was presented to the user as a probability distribution on the state of the world. Based on these probabilities, the user had to choose a road among several options. Subsequent work proposed a method for advice-generation in path selection problems which are difficult for people to solve. In those settings, the agent and the person interact only once and both have full information about the roads network. The user's sole incentive was to choose the shortest path, while the agent's incentives also included the number of color changes in the path. Recent work in E-commerce that also considered information disclosure to people as a way to affect their performance [26, 25]. These works consider comparison shopping agents (CSAs) and suggest a set of methods for affecting users decisions based on selective disclosure of information and anchoring, aiming to influence users not to query additional CSAs. None of these works consider repeated interactions in which there is a need to model how the user reacts to the agents' advice. Elmalech et al. [15] show that an agent trying to maximize a

user's expected utility, should provide advice which is not based merely on the encapsulated utility, but rather also on the likelihood of its acceptance by the user.

Past work in user-modeling have generated advice in dialogue systems or collaborative office assistants [47]. These models have relied on probabilistic and decision-theoretic methods [28], plan recognition and logical approaches [3] for inferring users' goals and activities. We extend these works by incorporating features from behavioral economics in our models of human-decision-making, and showing the efficacy of this approach empirically.

Much of the work in human computer interaction related to encouraging certain behavior focuses on visualizations and feedback [30, 38, 39]. These works succeed in modifying human behavior by causing users to be more aware of the consequences of their actions. Froehlich et al. [19] survey many persuasive technologies with a goal of reducing environmental impact. However, in these works the system does not provide advice to the user nor builds a model of human behavior over time and use this model to provide advice to the user as we do.

Fogg surveys many technologies trying to persuade humans, and analyzes the main properties required for such persuasion technologies to be successful [18]. One example is an exercise bicycle connected to a TV ("Telecycle"). In this system, as you pedal at a higher rate the image on the TV becomes clearer. This way the Telecycle encourages humans to exercise at higher rates. This work also describes different methods for persuasive systems such as a social actor - an example is the Banana-Rama slot machine which has characters which celebrate every time the gambler wins. Fogg later states that in order to be persuasive, a system must be credible, i.e. both trustworthiness and expertise.

3 Choice Selection Processes

A choice selection process is a repeated interaction with incomplete information between a receiver and a sender. Each round, the sender observes the state of the world $v \in \mathcal{V}$, drawn from some distribution $P(\mathcal{V})$, and can advise the receiver to take one of the actions in a predefined set, $d \in A$. After observing the advice given by the sender (d), the receiver chooses one of the actions $a \in A$. The costs to the receiver and to the sender depend on the action chosen by the receiver and the state of the world, which are denoted $c_R(a, v)$ and $c_S(a, v)$, respectively. Both players, receiver and sender, can observe the outcome at the end of each round. In contrast to the sender, the receiver does not know the state of the world nor the costs for the sender. This interaction is repeated indefinitely and players' costs each round are discounted by a constant factor γ. The sender knows the distribution over $\mathcal{V}$ and the costs of both participants, for possible future interactions (note that the sender is revealed the exact state of the world, v, only once that round is reached).

A round t in a selection process is represented by a tuple $h^t = (a^t, c^t, d^t)$ where a^t is the receiver's action at round t, $c^t = (c_R^t, c_S^t)$ is the cost for the receiver and sender at t, and d^t is the advice provided by the sender at t (prior to the receiver choosing a^t) given the state v. Here, c_R^t denotes $c_R(a^t, v)$ and c_S^t denotes $c_S(a^t, v)$. We define the history from round 1 through t as $h^{1,t} = h^{1,t-1} \circ h^t$. For $t = 0$ the cost functions c_R^t and c_S^t are initialized to 0 and $h^{1,0}$ is initialized to an arbitrary a and d. $H^{1,t}$ defines the set of all possible history sequences (the set of all $h^{1,t}$). Table 1 provides a complete list of notation used in the paper.

The next sections show two different advice provision domains which were modeled using selection processes. In the first domain users do not have information about the state of the world and therefore each interaction can be considered a new interaction. In this case, users cannot increase their knowledge about the state of the world over time. In the second

Notation	Meaning
a	action.
A	action space.
$c_R(a, v)$	receiver's cost as a function of the action a and state v.
$c_S(a, v)$	sender's cost.
d	advice given by the sender.
$EC_S^t(v, h^{1,t-1}, d)$	optimal expected cost for sender at time t as a function of current state v, the history $h^{1,t-1}$ and advice d.
h^t	history at time t composed of (a^t, c^t, d^t).
$m_b(a, v, t)$	receiver's belief about its comfort level for action a state v and time t (CCS domain).
$o(v)$	observation obtained by receiver, depending on the state v (CCS domain).
$p(v)$	density function of the state space.
$P(a)$	probability that the receiver will take action a.
$HC^a(t)$	hypothetical cost for receiver for taking action a at time t.
$HC^F(t)$	hypothetical cost for receiver for following the advice at time t.
$HC^N(t)$	hypothetical cost for receiver for not following the advice at time t.
t	round number (time).
$tr(h^{1,t-1})$	trust rate given the history $h^{1,t-1}$(CCS domain).
v	state of the world.
$\mathcal{V}$	state space.
w	weight.
w_i	optimized parameters.
$AC^a(t, h^{1,t-1})$	aggregated hypothetical cost for receiver for action a from round 1 to round $t-1$ (route selection domain).
$AC^F(t, h^{1,t-1})$	aggregated hypothetical cost for receiver for following the advice from round 1 to round $t-1$.
$AC^N(t, h^{1,t-1})$	aggregated hypothetical cost for receiver for not following the advice from round 1 to round $t-1$ (CCS domain).
α, β	parameters used in cost function for receiver (CCS domain).
γ	discount factor in choice selection process.
δ	discount factor for aggregated hypothetical cost.
λ	parameter for logit quantal response.
$C(a, v, d, t, h^{1,t-1})$	sender's model of receiver's cost as a function of the action a, the state v, the advice d, the time t and the history $h^{1,t-1}$ (CCS domain).
$\pi(v, w)$	sender's advice (for SAP) assuming world state v and the use of the weight w.

Table 1 List of notations

domain, users receive observations about the state of the work prior to making a decision regarding and they can choose the extent to which to follow the advice that is provided by the agent.

4 Route Selection Domain

In this domain the driver (the player playing the receiver) can choose one of A roads for his or her commute. The state of the world $v = (v_1, \ldots, v_{|A|})$ is sampled from a continuous multivariate random variable that represents the traffic condition (travel time and fuel consumption from source to destination) for each of the roads. At each round, the system (the sender) observes the state of the world and suggests one of the roads in A to the driver. The outcome for both participants depends on the road $a \in A$ chosen by the driver as well as the road conditions v_a. Since the person does not know the actual state of the world, and in particular the costs of all actions in each round, we need to express his hypothetical costs when reasoning about which action to take.

We define the *hypothetical cost* the receiver incurs for taking action a at time t, denoted $HC^a(t, h^t)$ to equal the cost c_R^t when $a = a^t$ (i.e., the receiver chose action a^t at time t); if $a \neq a^t$ then the person did not choose action a^t, and its hypothetical cost equals some default value w_3. This is because the person does not know what cost would have been incurred by taking action a^t for rounds that it was not chosen. For example, suppose that the receiver chose to use route 66 on day 1 and incurred a 45 minute commute. The hypothetical cost of the receiver for using route 66 on day 1 equals 45 minutes, while the hypothetical cost for using any other route equals the default value.

The probability distribution that the receiver will take action a^t at round t given advice d, and behavior $h^{1,t-1}$ in past rounds is denoted $P(a \mid h^{1,t-1}, d, t)$. For a given world state v and history $h^{1,t-1}$, the sender's expected cost $EC_S(v, h^{1,t-1}, d)$ for advice d is an expectation over its future costs given it gives the best advice d' at each time step. The best advice is the one computed by the optimal policy π^* as follows:

$$EC_S^t(v, h^{1,t-1}, d) = \sum_{a \in A} P(a \mid h^{1,t-1}, d, t) \cdot \left(c_S(a, v) + \gamma \int_{v'} P(v') \text{min}_{d'} EC_S^{t+1}(v', h^{1,t}, d') dv'\right) \quad (1)$$

For a given world state v and history h^{t-1}, the advice d that minimizes the sender's cost is a policy $\pi^*(v, h^{1,t-1}, t)$ defined as follows:

$$\pi^*(v, h^{1,t-1}, t) = \text{argmin}_d EC_S^t(v, h^{1,t-1}, d) \quad (2)$$

As we later show, there is a natural mapping from this formalization to a Markov decision making problem for the sender agent.

4.1 Modeling Human Receivers

In this section we provide a model of a human receiver player in choice selection processes for the case in which the state of the world is not observed by the receiver.

Because a receiver cannot observe the state of the world nor its distribution, his decision problem can be analogously described as a Multi Armed Bandit Problem (MAB) [4], in which there are $|A| + 1$ arms (one for each action, and one for following the advice of the sender). We therefore assume that the receiver records the utility obtained from each of the actions (or arms) and is more likely to choose an action (or arm) that performed better in the past. If following the advice yielded a high performance for the user, he will be more likely to follow the advice in future actions.

The models of human receivers we present below differ in the way in which they compute the utility that is attributed by the receiver for each choice. Before presenting the model, we need to make the following definitions. First, we generalize the definition of *hypothetical cost* of the receiver for following the advice of the sender. We define the hypothetical cost incurred by the receiver for taking advice d at time t, denoted $HC^F(t, h^t)$, to equal the cost c_R^t when $a^t = d$ (i.e., the receiver followed the sender's advice), or a default value. Note that F (which stands for following the advice) is simply part of the function name and may not take any value (unlike HC^a, in which a may be any action). $HC^{(\eta)}$ will denote a general hypothetical cost function in which (η) may either be an action a (implying that $HC^{(\eta)}$ will compute HC^a for any action a) or a constant F (implying that $HC^{(\eta)}$ will compute HC^F).

Next, we generalize the notion of the receiver's hypothetical cost to include behavior over multiple rounds. Let $AC^a(t, h^{1,t-1})$ denote the aggregate hypothetical cost incurred by the receiver for taking action a, taking into account all rounds 1 through $t-1$, and $AC^F(t, h^{1,t-1})$ the aggregate hypothetical cost incurred by the receiver for following the advice, taking into account all rounds 1 through $t-1$. This aggregated hypothetical cost, AC^a, builds upon the hypothetical cost and thus, is well defined for all actions a even though, the receiver may have taken different actions at certain rounds. Similarly, AC^F builds upon HC^F, and thus is well defined even though the receiver may have not followed the advice in certain rounds.

We can now describe several models which differ in how they aggregate the receiver's hypothetical costs over time. We begin with two models in which receivers discount their past costs higher than their present costs. In the *hyperbolic discounting model* [11,14], the discount factor δ falls very rapidly for short delay periods, but falls slowly for longer delay periods. For example, consider a driver who took a new route to work on Monday which happened to take an hour longer than the route on Friday. According to hyperbolic theory, the relative difference between the commute times will be perceived to be largest during the first few days following Monday. However, as time goes by, the perceived difference between the commute times will diminish. Equation 3 models the accumulative cost in the *hyper* model:

$$AC^{(\eta)}(t, h^{1,t-1}) = \sum_{t'<t} \frac{HC^{(\eta)}(t, h^{t'})}{\delta \cdot (t - t')} \tag{3}$$

Where (η) may either be an action a, or F for following the advice.

In the *Exponential Smoothing* model [21], the discount factor δ is constant over time, meaning the perceived difference between the commute times will stay the same over time. The hypothetical cost for the receiver is defined as follows. If $a^{t-1} = a$ (the receiver took action a at time $t-1$) or $a^{t-1} = d$ (the receiver followed the advice specified in h^{t-1} of $h^{1,t-1}$) then we have:

$$AC^{(\eta)}(t, h^{1,t-1}) = \delta \cdot HC^{(\eta)}(t, h^{t-1}) + (1-\delta) \cdot AC^{(\eta)}(t-1, h^{1,t-2}) \tag{4}$$

If $a^{t-1} \neq a$ or $a^{t-1} \neq d$ the receiver does not update his aggregate hypothetical cost for action a or the advice respectively, and we have

$$AC^{(\eta)}(t, h^{1,t-1}) = AC^{(\eta)}(t-1, h^{1,t-2}) \tag{5}$$

If $t = 1$ then $AC^{(\eta)}(t, h^{1,t-1})$ equals the default value w_3 for any (η).

In the *Short Term Memory* model, the receiver's valuation is limited to the past 7 rounds, (the number of items commonly associated with human short term memory capacity [34, 31]). The aggregated hypothetical cost for the receiver is defined as follows:

$$AC^{(\eta)}(t, h^{1,t-1}) = \sum_{t-7 \leq t' < t} HC^{(\eta)}(t', h^{1,t'-1}) \cdot \frac{1}{7} \tag{6}$$

If $t < 7$, then the summation only spans rounds 1 through t, and the denominator is replaced by t (the receiver is assumed to remember all utilities obtained if there were less than 7 rounds in total).

Lastly, as a baseline, we consider the *Soft Max* model [46] in which the aggregate hypothetical cost of the receiver for any action is simply the average true cost (as opposed to the hypothetical cost) of taking this action in past rounds, with no discount factor:

$$AC^{(\eta)}(t, h^{1,t-1}) = \frac{\sum_{1\leq t'<t} c_R^{t'} \cdot \mathbf{1}\{(\eta) = a_{t'} \vee \big((\eta) = F \wedge d_{t'} = a_{t'}\big)\}}{\sum_{1\leq t'<t} \mathbf{1}\{(\eta) = a_{t'} \vee \big((\eta) = F \wedge d_{t'} = a_{t'}\big)\}} \tag{7}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. In order to avoid division by 0, some default value is assigned to actions which were never performed.

When estimating the probability of the action a, one should reason about the past experience of the receiver from taking this action ($AC^a(t, h^{t-1})$) and the experience of the receiver from following the advice of the sender ($AC^F(t, h^{t-1})$). The probability of choosing a certain action should increase if that action was advised by the sender. Therefore, for all the four suggested models for the aggregated hypothetical cost we adopted the quantal response theory from behavioral economics [24] for choice of actions. This theory assigns a probability of choosing an action a that is inversely proportional to the aggregate hypothetical cost of that action given the history (i.e. $AC^{(\eta)}(t, h^{t-1})$). The receiver is modeled to prefer actions associated with lower hypothetical costs. However, with some probability, the receiver may still choose actions that are more costly.

Formally, the probability that the receiver will take action a^t at round t given behavior in past rounds $h^{1,t-1}$ depends on the benefit $AC^F(t, h^{1,t-1})$ from the advice d that was given at this round.

$$P(a, t\,|h^{1,t-1}, d) = \frac{e^{-\lambda \cdot AC^a(t,h^{1,t-1})} + Z}{e^{-\lambda \cdot AC^F(t,h^{1,t-1})} + \sum_{a\in A} e^{-\lambda \cdot AC^a(t,h^{1,t-1})}} \tag{8}$$

Where Z is set to equal $e^{-\lambda \cdot AC^F(t,h^{1,t-1})}$ when $a = d$, and otherwise zero; λ is a smoothing parameter. Note that all methods have parameters which must be learned from data. These parameters are assessed in section 4.3.1.

4.2 Agent Design for Sender Role

In this section we formally define the problem of finding the optimal strategy for the sender player in a selection process, and present several approximate solutions to the problem given a model of the receiver's decision making process. To this end we present two possible agent designs, one that uses a Markov Decision Process and one that uses a social preference model.

4.2.1 Markov Decision Process

In this approach the sender's decision making process is represented as a continuous MDP. These models provide a decision-theoretic framework for reasoning about uncertainty over time and are used extensively in MAS to model planning coordination activities [23, 22]. To represent the selection process from the sender's point of view as an MDP, we define the set of world states for the MDP as follows.[1] Every time t, state $v^t \in \mathcal{V}$ and history sequence

[1] We use the term "world state" to disambiguate the states of an MDP from those of a selection process.

$h^{1,t} \in H^{1,t}$ define a world state $s^t = (v^t, h^{1,t-1})$. The set of all such world states at time t is:

$$S^t = \{(v^t, h^{1,t-1}, t) \mid v^t \in \mathcal{V}, h^{1,t} \in H^{1,t}\} \tag{9}$$

and the set of possible world states is defined as $S = \cup_{t=1}^{\infty} S^t$. The set of actions for the sender is the set $|A|$ of actions in the selection process. The reward function for the MDP, denoted $r(s^t)$, is defined as $-c_S^{t-1}$ (which is a part of the history at time $t-1$ - see Section 3.)

The discount factor for the MDP is γ. The transition function of the MDP is set to

$$P(s^{t+1} \mid s^t, d^t) = P(a^t \mid h^{1,t-1}, d^t, t) \cdot P(v^{t+1}) \tag{10}$$

where s^{t+1} as above and $P(v^{t+1})$ is the probability that the selection process state v^{t+1} will occur. Finally, the initial state of the MDP is sampled from the world states subset $\{(v, \emptyset, 1) \mid v \in \mathcal{V}\}$ according to $P(v)$, and the optimality criterion is set to be the minimization of the expected accumulated cost.

Proposition 1 *Solving the MDP described above will yield a policy that satisfies equation 2.*

Proof Given a world state $s^t = (v^t, h^{1,t-1})$, we define the $Q(s^t, d)$ and value function $V(s^t)$ for the MDP as follows:

$$Q(s^t, d) = r(s^t) + \gamma \int_{s'} P(s' \mid s^t, d) \cdot V(s') ds' \tag{11}$$

$$V(s^t) = \max_d Q(s^t, d) \tag{12}$$

and the optimal policy $\pi^*(s^t)$ is defined as

$$\pi^*(s^t) = \arg\max_d Q(s^t, d) \tag{13}$$

Recall that $s^t = (v^t, h^{1,t-1})$ and $r(s^t) = -c_S^{t-1} = -c_S(a^{t-1}, v^{t-1})$. Therefore Equation 11 may be replaced by:

$$Q((v^t, h^{1,t-1}), d) = -c_S(a^{t-1}, v^{t-1}) + \gamma \cdot \sum_{a \in A} P(a \mid h^{1,t-1}, d, t) \cdot \int_{v^{t+1}} P(v^{t+1}) \cdot \max_d Q((v^{t+1}, h^{1,t}), d) dv^{t+1} \tag{14}$$

According to Equation 1 we obtain that $EC_S^t(v^t, h^{1,t-1}, d)$ is proportionate to $-Q(s^t, d)$. Therefore, the optimal policy $\pi^*(s^t)$ in Equation 13 satisfies Equation 2.

Therefore, solving the continuous MDP described above yields an optimal policy for the sender given a model of the receiver, $P(a^t \mid h^{1,t-1}, d^t, t)$. However, the world states of the MDP incorporate the continuous state of the selection process and discrete histories of arbitrary length, which makes the MDP structure too complex to be accurately solved. In addition, we cannot use existing approximation algorithms, which assume a finite state space [32], partition of the state space [16], or use kernel-based methods [36], due to the mixture of the continuous component (selection process state) and an arbitrarily large discrete component (action and advice history) of the world state.

Given these constraints, we suggest an agent design that does not solve the MPD explicitly, but uses the models for human receivers described above to reason about the consequence of their actions over time. The agent, called *MCS*, chooses the optimal advice for the current time step while using Monte-Carlo Simulation [33, 27] for selecting future states according to the transition function of Equation 10, and selecting future actions of the sender according to a uniform probability distribution.

4.2.2 Social Preference Approach

According to the social preference theory, people consider others' outcomes as well as their own when making strategic decisions [20]. The agent design we propose here is called SAP, a Social agent for Advice Provision, that generates advice according to the following social model. Our approach explicitly reasons about the trade-offs between the costs to both participants in the selection process based on a social weight. The intuition behind this agent is the following: assume the agent only considers the system's cost. After several interactions with the user, the user is likely to ignore any future advice, resulting in a high cost to the agent. On the other hand, if the agent considered only the user's cost, the user will probably follow the agent's advice, however, the user will be choosing actions which are not as good for the agent. Therefore, SAP tries to find the optimal balance for the agent by applying a weight to each of the participants' costs. For a state v and a weight w, a policy for advice provision is a decision d with minimal social cost.

$$\begin{aligned} d = \pi(v, w) = \arg\min_{d \in A} (1-w) \cdot (c_R(d, v)) + \\ w \cdot (c_S(d, v)) \end{aligned} \tag{15}$$

where w is a constant weight. In practice we scale c_R (and c_S) by dividing it by the average cost of the receiver (or sender respectively), so that $w = 0.5$ will imply an equal weight for both c_R and c_S. We will refer to this wight, w, as the selfishness of the agent.

To compute the most beneficial weight w^*, we need to assume some behavior on the part of the user ($P(a \mid h^{1,t-1}, \pi(v, w), t)$) when he interacts with an agent that provides pieces of advice to him ($\pi(v, w)$ based on Equation 15). See examples of such models for human behaviors in Section 4.1. Then, the weight most beneficial to the agent, w^*, is searched in the space of all weights. The result is the weight with minimal total expected cost for the agent.

For a given world state v and history h^t, we can define the sender's expected cost $EW_S^t(v, h^{1,t-1}, w)$ for weight w and fixed policy $\pi(v, w)$. Note that this is not the optimal expected cost for the sender described in Equation 1 as it does not require to solve the intractable $\min$ expression in Equation 1 to obtain the future advice but instead uses $\pi(v, w)$ as a fixed policy.

$$\begin{aligned} EW_S^t(v, h^{1,t-1}, w) = \sum_{a \in A} P(a \mid h^{1,t-1}, \pi(v, w), t) \cdot \\ \left(c_S(a, v) + \gamma \int_{v'} P(v') EW_S^{t+1}(v', h^{1,t}, w) dv'\right) \end{aligned} \tag{16}$$

Note that in each iteration of the search process, w remains fixed for that iteration in the rightmost term of equation 16. The weight w is chosen to minimize the sender's aggregate costs for the fixed policy $\pi(v, w)$

$$w^* = \arg\min_w EW_S^t(v, h^{1,t-1}, w) \tag{17}$$

4.3 Empirical Methodology

We evaluated the different agent models (SAP and MDP) using an empirical study in a route-selection domain. In the route-selection domain a driver needs to choose one of 4 possible routes to get to work. The system can advise the driver to take one of the routes before the driver makes a choice. The road conditions (i.e., travel time and fuel consumption) constitute the state of the world, and vary due to traffic and maintenance. This information is unknown to the driver when he makes his decision. The driver's goal is to minimize the travel time over all rounds, and the system's goal is to reduce fuel consumption over all rounds. This is obviously one example and it shows an extreme case where user's and agent's goals do not conflict but do not necessarily overlap. Real world scenarios will naturally be more cooperative. For example, a user might prefer to arrive the fastest possible route but he would also like to save fuel. That is, while arriving fast is the most preferred criteria he does not oppose to saving fuel as long as it does not significantly affect his time of arrival. Our results show that even in the less cooperative situation, the agent succeeds in changing the user's choices such that both will benefit. As stated, the purpose of our advice provider agent is not to impose the action that is most beneficial to the agent, but to lead the user to change his choices in the direction of the most beneficial action as long as his other preferences can be preserved.

After the driver chooses a route, both participants incur a cost which depends on the road conditions of the chosen route. At this point the interaction continues to the next round with a probability of 0.96. (This probability was chosen to align with the expected number of commuting days of 25 which is the average commuting days in one month). The conditions of the roads in each round are sampled from a joint distribution that is known to the agent, but not to the driver. We modeled the fuel consumption and travel time using a multivariate log-normal distribution.

We enlisted 123 subjects, 57.6% females and 42.4% males, from the USA (recruited via Mechanical Turk). The subjects' ages ranged from 19 to 69, with a mean age of 37.6 and median of 35. Subjects were paid 12 cents for participating in the study, and additionally received between 5 to 50 cents depending on their performance. The subjects were told that the probability of a new round was 0.96. The actual number of rounds was not revealed to the subjects (nor to the computer agents). The subjects were paid a bonus proportional to the average travel time (the lower the travel time the higher the bonus). All subjects were provided an explanation of the game and its details, as described in the beginning of this section and they had to pass a quiz. The subjects were told that the agent providing advice had a goal different from theirs. In each round, after receiving the advice from the agent, the subjects had to select a road. Then the subjects were told how much time it took them to travel via that road. The history, including previous advice, previous actions and previous travel time was available to the subjects at all times.

4.3.1 Model Selection for the Receiver

To compare the various models of the receiver, we collected 2250 rounds of 90 subjects to train and evaluate the Short-term memory, hyperbolic discounting (Hyper), SoftMax , and Exponential Smoothing (ES) models that were described earlier. In this training phase, the users chose roads, while receiving recommendations from one of the baseline agents: *Sender*, that advised to take the road with the least fuel consumption , *Receiver* that advised to take the road with the lowest travel time or *Silent* that did not provide any advice. For each of these models, we estimated the maximum-likelihood (ML) value of the parameters

Table 2 Fit-to-data of different receiver models (the lower the better)

model	d.f.	N-Log-Like.
SoftMax	1	178.5
ES	2	172.2
hyper	2	$\mathbf{169.4}$
short memory	1	186.9

Table 3 Settings used in the route selection domain.

parameter	road #1	road #2	road #3	road #4
average travel time	72	84	52	64
travel time stdev	14	24	16	4
average fuel consumption	4	4.4	8	6
fuel consumption stdev	1.2	1.2	2	1.6

using sampling, and computed the fit-to-data of the test set using the ML values. All results reported throughout the section were confirmed to be statistically significant using the Mann-Whitney U test with $\alpha = 0.05$. Table 2 presents the fitness of the models employing a tenfold-cross-validation on all the training data (lower values indicate a better fit of the model). As shown in the table, the Hyper model, which modeled the receiver using the hyperbolic discounting theory (Equations 3 and 8) exhibited a higher fit-to-data than all the other models of human receivers.

We hypothesized that the use of the social utility approach would lead to the best performance of the agent sender, measured in terms of fuel consumption. To evaluate this hypothesis, we used different agent designs to generate offers to people which incorporated the decision-making strategies that were described in the previous section. Specifically, we used an agent that incorporated the social utility approach to make offers, termed the Social agent for Advice Provision (*SAP*). Building upon a human model, SAP, using a simulation of the environment, searched for w^* (the optimal weight in Equation 17). Since the *hyper* model had the best fit-to-data, SAP used it as the human model. Iterating on different possible w, SAP simulated 10000 users for each w, where each user was simulated for a full process (until it terminated). We considered w's in $\{0, 0.01, 0.02, ..., 0.99, 1\}$. SAP chose the w with the lowest overall average cost as w^*. Then, in each round, SAP provided advice according to Equation 15, using the optimal weight. The second agent used the MDP model to make offers, by solving Equation 11. We estimated $V(s^t)$ using Markov Chain Monte Carlo sampling [33,27] in a manner similar to that of the MCTS method mentioned in Silver et al. [44].[2] We term this agent *MCS*. We also employed two baseline agents, *Random* that offered roads with uniform probability and *Silent* that did not provide any advice.

We evaluated these agent designs in simulation as well as in experiments involving new human subjects. The simulation studies consisted of sampling 10,000 road instances according to the distribution over the fuel consumption and travel time in Table 3. Fuel and energy consumption were sampled independently.

As an alternative to the hyperbolic discounting model, we also considered an approach using an $\epsilon-$greedy strategy to describe possible behavior of a receiver. This strategy assumes that the decision problem from the point of view of the receiver is a Multi Armed Bandit Problem (MAB) [4,46], in which there are $|A| + 1$ arms (one for each action, and one for

[2] This method is more common in POMDPs, however, since our state space is very large, we use this method as well.

Table 4 Simulation results comparing agent strategies

human model	agent strategy	fuel (liters)	time (minutes)
hyper	Random	6.120	64.40
	Silent	6.297	63.04
	MCS	5.792	65.92
	SAP	**5.520**	64.54
$\epsilon-$greedy	Random	7.046	58.08
	Silent	7.104	57.68
	MCS	6.812	59.26
	SAP	**6.432**	55.84

following the advice of the sender). The MAB approach has been used in MAS in the past to model team coordination [45].We therefore assume that the receiver records the utility obtained from each of the actions (or arms) and is more likely to choose an action (or arm) that performed better in the past. If following the advice yielded a high performance for the user, he will be more likely to follow the advice in future actions. It provides a rational baseline that seeks to minimize travel time for receivers over time.

Table 4 presents results of the simulation. We compared the fuel consumption costs incurred by the different sender agents for each model used to describe human behavior. As shown in Table 4, the cost accumulated by the SAP agent using the hyperbolic discounting model was 5.52 liters (shown in bold), which was significantly lower than the costs incurred by all other agents using the hyper models to describe human behavior. Similarly, the cost accumulated by the SAP agent using the $\epsilon-$greedy model were significantly lower than the costs incurred by all other agents using the $\epsilon-$greedy model.

4.3.2 Evaluation with People and Generalization

Given the demonstrated efficacy of the SAP agent in the simulation described above, we aimed to evaluate the ability of the SAP agent to generalize to new types of settings and new people. We hypothesized that a SAP agent using the hyperbolic discounting model to describe receiver behavior when selecting w^* would be able to improve its performance compared to a SAP agent using the $\epsilon-$greedy model. We randomly divided the subjects into one of several treatment groups. The subjects in the *Silent* group received no advice at all. The subjects in the *SAP-hyper* group received advice from the SAP agent that used a hyperbolic model to describe the receiver's behavior. The subjects in the *SAP-*ϵ group received advice from the SAP agent that used an $\epsilon-$greedy strategy to describe the receiver's behavior when selecting w^*. The subjects in the *Receiver* group were consistently advised to choose the road that was most beneficial to them, (i.e., associated with the lowest travel time). Lastly, the subjects in the *Sender* group were consistently advised to choose the road which was best for the sender (i.e., associated with the lowest fuel consumption).

Figure 1 presents the fuel consumption of each one of the treatment groups (the black error-bars represent 95% confidence bars). As can be seen in the figure, the SAP-hyper agent significantly ($p < 0.05$ using the Mann-Whitney test) outperformed all other agent-designs, accumulating a cost of 5.08 liters. The MCS method (which uses Monte Carlo sampling) came in second, accumulating an average cost of 5.35 liters. Table 5 shows additional information on each one of the treatment groups. The performance for agents and for people is measured in terms of overall fuel consumption and commuting time, respectively. The

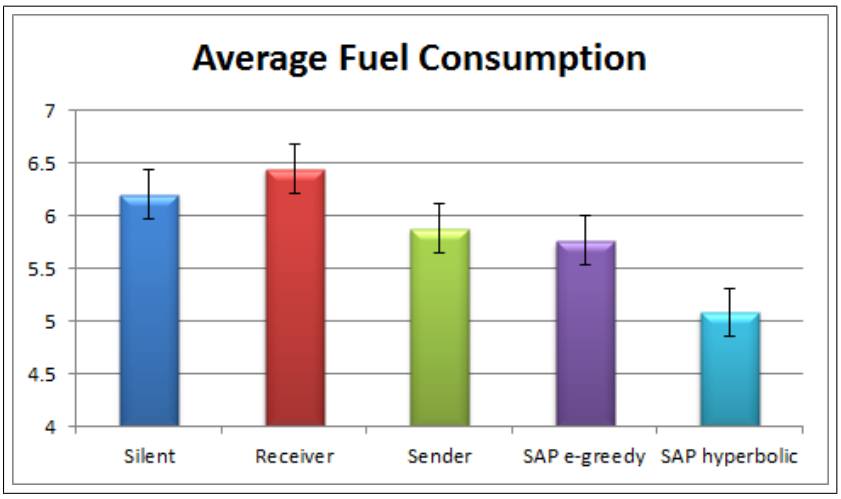

Fig. 1 Average fuel consumption for each of the treatment groups (the lower the better).

Table 5 Performance results of agents interacting with people. The selfishness rate equals w in Equation 15

method	selfishness	fuel	time	acceptance
Silent	–	6.20	64	–
receiver	0	6.44	56.6	63.6%
sender	1	5.88	64.32	31.0%
SAP-ϵ	0.29	5.76	56.6	70.8%
MCS	–	5.35	67.1	52.2%
SAP-hyper	0.58	**5.08**	64.8	52.6%

"selfishness" column in the table measures the degree to which the agent was self-interested (the weight w in Equation 15).

4.4 Discussion

As we have shown, the SAP-hyper model was able to outperform all other alternative agent designs when interacting with people in the route-selection domain.The MCS (the pure decision theoretic model) came in second. In addition to SAP-hyper's higher performance in terms of energy consumption in comparison to MCS, the SAP-hyper method enjoys two additional advantages:

1. Online calculations are minor, and are limited to finding a minimum among several linear combinations (as opposed to MCS which simulates many future branches and thus requires high CPU processing that is calculated online, before it can provide advice).
2. The performance for the users was very similar to the performance of the Silent and sender methods (and much better than the MCS method).

The advice acceptance rates (i.e. the percent of times which a user followed the advice) for the SAP-hyper were lower than those for SAP-ϵ, which we attribute to the higher degree of selfishness of the SAP-hyper agent. Unsurprisingly, the best performance for people (travel time of 56.6 minutes) was achieved when using an agent that only considered people's costs (*receiver*). However, a similar result in terms of travel time was also obtained

by the $\epsilon-$greedy agent. Another surprising result is that the acceptance rate for SAP-ϵ was higher than that of the receiver agent, whose degree of selfishness was 0, and consistently recommended the route that was best for people. We hypothesize that this may have been caused by an unintended "too-good-to-be-true" signaling effect that is perceived by people.

One may be concerned with the relatively low user acceptance rate or by the relatively poor user performance for SAP-hyper. This may raise the concern that SAP might not perform as well when longer interactions are expected. Recall that the agent's goal was only to minimize its own cost. Although the agent did consider the user's cost and thus its satisfaction, it was considered a means to an end in order to minimize the agent's overall cost. If the system expects a longer period of interaction with the user (i.e. greater γ), the user's satisfaction will be more important to the agent, and therefore the social weight will be balanced towards the user's benefit (causing an increase in user acceptance rate and performance). Furthermore, if user satisfaction is important to the agent itself, it can be explicitly added to the agent's utility. However, we chose a more confrontational setting to demonstrate the efficacy of the method.

We conclude this section with two illustrative examples of the reasoning used by the SAP-hyper agent. In the first example, one of the roads incurs a very low cost for the agent (3 liters), but constitutes an extremely high cost for the person (43 minutes). In this example, the SAP-hyper agent recommended the road that was associated with the *highest* cost for the agent (4.19 liters), but a very low cost for the person (18 minutes). The person accepted this advice and chose the recommended route. In the next round, the agent advised the person to take a road that incurred a relatively high cost for the person (31 minutes) and a very low cost for the agent (1.6 liters). This offer was again accepted.

5 Climate Control Domain

In this section, we present a different type of choice selection process which includes a simulation system where a car driver needs to set how much power he would like his Climate Control System (CCS) to consume. We denote this the "power level" of the CCS. Higher values of the power level are associated with increased energy consumption by the system. The sender player represents a system which suggests a power level setting to the receiver (the driver). As in the road selection setting, in each round the sender can suggest to the receiver to perform a certain action before the receiver makes his selection.

This domain differs from the route selection domain in the following ways:

- Ordered actions: The action set is an ordinal scale which represents the energy consumption level of the CCS in the car. The roads in the previous domain that we examined were not sorted in any scale. The actions were a set of non-ordered options.
- Partial acceptance: The receiver may *partially* accept the advice (e.g. set the power level of the CCS to a lower level than initially intended, but not as low as suggested by the sender). This makes the task of modeling the receiver significantly more difficult. In the roads domain, a user could either accept or not accept the advice; in the climate control case a user can partially accept advice and in a sense make a choice is closer to the advice.
- Cost for receiver: The cost for the receiver depends on two attributes: the energy consumption of the CCS, and the user's comfort level which depends on the energy consumption and the state of the world. Therefore, modeling the human behavior becomes a more complex task in this case than in the roads domain.

- Partial observability: The receiver is given an observation (the heat load) that is associated with the state of the world. Therefore he is able to update his belief regarding the state of the world. In the roads domain, the user was assumed not to have any information about the traffic distribution in the different roads.
- Finite state space: In this configuration the state space is constrained which allows us to solve the MDP. In the roads domain, the state space was larger and it was not practical to find the optimal solution to the corresponding MDP.

5.1 Setting Description

In this setting, A is an ordered set of actions $(1, \ldots, |A|)$. Each action represents the setting of the power level of the climate control system. The state of the world $v = (v_1, \ldots, v_{|A|})$ represents the "comfort level" for the receiver (i.e., the driver of the car) when operating the AC system according to each of the possible system settings. Note that the comfort level may have a real (non-integer) value.

In the choice selection process in each round t, the receiver is given a discrete observation $o(v)$ that represents the current heat load in the car (a function of the temperature, humidity and other environmental conditions). Note that because the receiver does not directly observe the state v, he does not know the comfort level. The assumption is that a user who is new to such an interaction does not yet know how he would feel at the end of the drive for any particular setting. The cost function for the receiver, $c_R(a, v)$ is a linear combination of the energy consumption (a) and the comfort level (v_a).

$$c_R(a, v) = \alpha \cdot v_a + \beta \cdot a \tag{18}$$

where $\alpha \leq 0$ and $\beta \geq 0$ are constants in the problem definition. The cost for the sender, $c_S(a, v)$, is determined by the action taken by the receiver (the energy consumption), i.e. $c_S(a, v) = a$. The next round of the choice selection process occurs with a constant probability γ.

Because the receiver is given an observation about the state v, we predict the probability that the receiver will choose action a, which depends on the history, the advice of the sender in the current round, and the state of the world:

$$P(a \mid h^{1,t-1}, d, t, v) \tag{19}$$

Similar to the road selection domain, for a given world state v and history $h^{1,t-1}$, we can define the sender's expected cost $EC_S(v, h^{1,t-1})$ for action (i.e., advice) d as

$$EC_S^t(v, h^{1,t-1}, d) = \sum_{a \in A} P(a \mid h^{1,t-1}, d, t, v)\big(c_S(a, v) + \gamma \sum_{v' \in V} p(v')(\min_{d'} EC_s^{t+1}(v', h^{1,t}, d'))\big) \tag{20}$$

The advice that minimizes the sender's cost is

$$\pi^*(v, h^{1,t-1}, d) = \text{argmin}_d EC_S^t(v, h^{1,t-1}, d) \tag{21}$$

It is important to observe that in our world all variables in the optimization problem for the sender are known to the sender except $P(a \mid h^{1,t-1}, d(v, h^{1,t-1}), v)$, which requires a human model of the receiver. Therefore, the next subsection is dedicated to methods for modeling a human receiver.

5.2 Modeling Human Receivers

An important factor in predicting the receiver's action is the sender's model of the cost incurred by the receiver. This modeled cost is a function of the action taken by the receiver, the history, the advice and the state of the world and is denoted $C(a, v, d, h^{t-1})$. We present several possibilities for such a model. In the simplest case, this modeled cost is assumed to be the receiver's true cost.

$$C(a, v, d, h^{t-1}) = c_R(a, v) \tag{22}$$

We term this candidate True-Cost.

Another candidate for the receiver's cost is a weighted sum over the comfort level a_v and the receiver's action a

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot v_a + w_2 \cdot a \tag{23}$$

We assume that $w_1 \leq 0$ and $w_2 \geq 0$. A similar approach (based on building a hypothetical utility function using a linear combination of the parameters and using a quantal response) for modeling humans was performed successfully in previous work ([35], [7]). This candidate is termed LUQ (Linear combination for hypothetical Utility and Quantal response). Recall that the true cost for the receiver is a linear combination of the comfort level and the action performed by the receiver as well. Although, LUQ assumes that, the modeled cost $C(a, v, d, t, h^{1,t-1})$, is also a linear combination of both attributes of the problem, i.e. the comfort level and the energy consumption, the coefficients (w_1 and w_2) may differ from the coefficients used in $c_R(a, v)$ (α and β)[3].

An alternative to the models shown above is to specifically represent the sender's advice in the cost function of the receiver. We define the hypothetical cost of the receiver from following the advice as $HC^F(t \mid h^t)$ as in the road selection domain. Additionally, we define $HC^N(t, h^t)$, to equal the cost c_R^t when $a^t \neq d^t$ (i.e., the receiver did *not* follow the sender's advice); otherwise it equals some default value (w_3). Note, that while $HC^F(t \mid h^t)$ was already defined in the road selection domain, $HC^N(t, h^t)$ is new to the CCS domain. In the road selection domain, instead of modeling the cost to the user when not following the advice, we model the cost to the user for each of the possible roads (arms). That is, we assume that the user assigns a cost for each of the different roads and a cost for following the advice ($HC^F(t \mid h^t)$). However, in the CCS domain, it would not be rational to assume that the user assigns a cost to each of the power levels, as the user's cost depends also on the users heat load (the user's observation). Therefore, we expect that the user will not assign costs to a certain action (e.g. setting the power level to 3 resulted with a low score), but that the user will assign costs to what happened when he did not follow the advice[4]. We define the following two formulas for the aggregated costs (Equations 24 and 25). These formulas are similar to those defined in the road selection domain (see Equation 3), but account for the slight variation considering how well the user may do when he does not follow the advice (i.e., η in the CCS domain may either be F or N, while in the road selection domain it could

[3] This model does not require an additional parameter for the actual cost for the receiver ($c_R(a, v)$), since $c_R(a, v)$ is already a linear combination of the comfort level and the energy consumption.

[4] In fact, the exact equivalent to the road selection domain, would be assuming that the user set a cost to each of the possible combinations of the heat load and each of the possible power levels. However, such an assumption would result with too many arms, most of which would not be sampled or sampled only once, and thus would not result in a good human model.

have been either F or each of the actions). We define two types of aggregated costs (for human receivers), one which employs hyperbolic discounting:

$$AC^{(\eta)}(h^{1,t-1}) = \sum_{t'<t} \frac{HC^{(\eta)}(t \mid h^{t'})}{\delta \cdot (t-t')} \tag{24}$$

where η is either F or N and δ is the discount factor parameter. For $t = 0$ we use the default parameter w_3.

The second type of aggregated cost employs exponential smoothing:

$$AC^{(\eta)}(h^{1,t-1}) = \sum_{t'<t} HC^{(\eta)}(t \mid h^{t'})\delta^{(t-t')} \tag{25}$$

We can now define the receiver's trust in the advice as a value between 0 (receiver does not trust the advice) and 1 (receiver fully trusts the advice):

$$tr(h^{1,t-1}) = \frac{1}{1 + e^{-\left(AC^N(t,h^{1,t-1}) - AC^F(t,h^{1,t-1})\right)}} \tag{26}$$

As an example assume that the receiver incurred very low costs when following the advice and very high costs when not following it. This will imply that $\left(AC^N(t, h^{1,t-1}) - AC^F(t, h^{1,t-1})\right)$ is a high positive number which in turn implies that $tr(h^{1,t-1})$ is close to 1.

Finally, we can define a candidate model for the receiver's cost that is a weighted average of the comfort level, the energy consumption (the receiver's action) and the trust of the receiver as a function of the distance between the action and the advice. Notice that since in this domain partial acceptance of advice is possible, we can consider the distance of an action from the advice:

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot v_a + w_2 \cdot a + w_4 \cdot tr(h^{1,t-1}) \cdot w_7^{-|d-a|} \tag{27}$$

where $w_1 \leq 0, w_2 \geq 0, w_4 \leq 0, w_7 \geq 0$. Here, the term $tr(h^{1,t-1}) \cdot w_7^{-|d-a|}$ increases proportionally to the receiver's trust in the advice, and the distance between the receiver's action and the advice. In particular, when the trust of the receiver is high, the difference between the advice and the receiver's action has a greater impact on its cost than when the trust of the receiver is low.

The following candidate model for the receiver's cost explicitly models how the receiver learns about the true comfort level over time. We define the receiver's estimate of the comfort level given round t, state v and action a as follows:

$$m_b(a, v, t) = \frac{1}{N_Z} \sum_{a' \in A} e^{w_8 \cdot |a'-a| + w_6 \cdot (t+1) \cdot \mathbf{1}\{a' \neq a\}} \cdot v_{a'} \tag{28}$$

where $w_8 \leq 0, w_6 \leq 0$, $\mathbf{1}\{\cdot\}$ is the indicator function and N_Z is a normalizing factor, such that:

$$N_Z = \sum_{\bar{a} \in A} \sum_{a' \in A} e^{w_8 \cdot |a'-\bar{a}| + w_6 \cdot (t+1) \cdot \mathbf{1}\{a' \neq \bar{a}\}} \cdot v_{a'} \tag{29}$$

We note that (1) large differences between a and a' imply more error, and thus the contribution of $v_{a'}$ to m_b decreases (2), as t increases, the receiver learns more about the true v_a and thus the contribution of $v_{a'}$ to m_b decreases.

The following is the receiver's cost which is identical to Equation 27 where the only difference is that the first parameter is multiplied by the receiver's belief over his comfort level ($m_b(a, v, t)$), rather than using the true comfort level (v_a):

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot m_b(a, v, t) + w_2 \cdot a + w_4 \cdot tr(h^{1,t-1}) \cdot w_7^{-|d-a|} \tag{30}$$

Finally, in all the above methods, we recall the function of the logit quantal response and adopt it to the climate control domain, and thus, the probability that the receiver shall choose an action a in any round t, given the state v and the receiver's aggregated hypothetical cost is:

$$P(a \mid h^{1,t-1}, d(v, h^{1,t-1}), v) = \frac{e^{-\lambda \cdot C(a,v,d,t,h^{1,t-1})}}{\sum_{a' \in A} e^{-\lambda \cdot C(a',v,d,t,h^{1,t-1})}} \tag{31}$$

Although λ is another parameter, it is used only for the True-Cost, and all other methods set it at 1 without losing any degree of freedom.

The last model we consider does not use the modeled cost function (C). This is a baseline model which uses the model which was found best in the road selection domain (hyper) and implies it on the CCS domain *without* accounting for the CCS domain different properties. This method, termed *MAB*, assumes that the receiver treats the problem as a multi armed-bandit problem where the advice is considered as an extra arm (for a total of 11 arms). This method is identical to the one used in the road selection domain and uses hyperbolic discounting, and therefore it ignores the receiver's observation, the fact that the actions are ordered and the differences between the two domains.

5.3 Agent Design for Sender

In the previous subsections we proposed different methods for modeling human behavior, which provide an estimation on $P(a \mid h^{1,t-1}, d(v, h^{1,t-1}), v)$. Based on these models we constructed two agents, SAP and MDP, for solving the optimization problem given in Equation 21. In the SAP agent the *Hyper with learning* human model (which resulted in the best fit-to-data see Section 5.6) is used for simulating the receiver's decision making process in order to search for the weights of the social utility function, which result in the lowest overall expected cost for the sender. In the MDP-based agent we simplified the receiver's model by using the *ES w/o learning* model which uses exponential smoothing rather than Hyperbolic discounting and does not assume any learning of the comfort level that occurs on the receiver's side. These simplifications only slightly decrease the suitability of the model to the collected data (see Table 6) but make the MDP feasible to solve. Due to the nature of *ES w/o learning* model the MDP world states do not require the whole history ($h^{1,t-1}$), but instead, allow the calculation of the sender's model of the receiver's cost, based solely on the current state (v), and the aggregated cost functions (AC^F and AC^N). This allows us to redefine the state space as $s = (v, AC^F, AC^N)$ and solve Equation 13. In order to solve the MDP, the state space must be discretized.

5.4 Experimental Settings

In our experiments we simulate a climate control system of an electrical car that interacts with a human driver. The sender in the climate control game represents the vehicle advisor system and the receiver represents the human driver. We set $A = \{1, 2, ..., 10\}$ as the set of possible energy consumption levels. The state of the world, v, was drawn uniformly from $\mathcal{V} = \{v^1, v^2, v^3, v^4, v^5, v^6\}$. The receiver's observation $o(v)$, was attributed to the heat load where 1 corresponds to a very light heat load, 2 to light, 3 to a moderate heat load, 4 to heavy, 5 very heavy and 6 to an extreme heat load. In our experiments we used the following function to determine the comfort level:

$$v_a^o = 10 \cdot \frac{1}{1 + e^{-(a-o)}} \tag{32}$$

This function was chosen since it encapsulates the following favorable properties: 1. The higher the CCS energy consumption the higher the comfort level. 2. The higher the heat load the lower the comfort level for a fixed CCS energy consumption level. 3. The comfort level is always between 0 and 10.

We set $c_S(a, v) = a$, i.e. the system cost is simply the energy consumption level. $c_R(a, v)$, the user's cost function, was captured as a utility function and was set as twice the comfort level (v_a) minus the energy consumption level a. More Formally:

$$c_R(a, v) = -2 \cdot v_a + a \tag{33}$$

5.5 Experiments

A total of 272 subjects from the USA (recruited via Mechanical Turk), of whom 44.4% were females and 55.6% were males, participated in the experiments in the climate control domain. The subjects' ages ranged from 19 to 67, with a mean age of 32.3 and median of 30. All subjects had to pass a short quiz to assure that they understood the game.

Every round the subjects were told the heat load for the current round and the advice given by the system. They had to select an energy consumption level for the climate control system (a number from 1 to 10). Then, they were told their comfort level and their final score for that round. Every round the subjects were shown their history, containing previous actions, previous observations, previous advice and the utility they gained. Similarly to the road selection domain, the subjects were paid 12 cents for participating in the study, and they received between 5 to 50 cents depending on their performance. The subjects were told that the probability of a new round was 0.96. They actually played 25 rounds, resulting in data obtained from $272 \cdot 25 = 6800$ rounds.

For the MDP agent, we discretized the state space to hold 40 different ranges for each hypothetical cost value, the states also held each of the 6 possible states of the world yielding a total of $40 \cdot 40 \cdot 6 = 9600$ states. Each state had 60 transitions; the number of the action (10) multiplied by the number of states of the world for the next round (6). We used value-iteration to solve the MDP - which took approximately 12 hours to solve on an Intel i5 $2.4Ghz$ CPU.

Along with all the above strategies, we also considered the behavior of a fully rational sender interacting with a fully rational receiver. A fully rational sender would never advise an energy consumption level which is strictly higher than the energy consumption level that is best for the receiver. (Assume by contradiction that the sender advises d where

Table 6 Fit-to-data of different receiver models in the climate control domain (lower is better)

model	d.f.	N-Log-Like.
True-Cost	1	830.6
LUQ	2	757.0
Hyper w/o learning	7	706.2
ES w/o learning	7	713.0
Hyper with learning	9	**677.3**
ES with learning	9	684.2
MAB	4	863.8

$c_S(d,v) > c_S(a',v)$ and $c_R(d,v) < c_R(a',v)$, whereby the sender may improve its advice to a' resulting in a lower cost in the current round along with reducing the receiver's cost which may increase future performance.) Therefore, a fully rational receiver, given the state, will search for its best action but never set its energy consumption level below the sender's advice. However, since the sender is trying to minimize the energy consumption level, it will always advise the lowest energy consumption level available (1). We will refer to a sender that always advises the lowest energy consumption level simply as *sender*.

As base-line we tested two additional strategies: *Silent* that did not provide any advice and *receiver*, that consistently advised the CCS energy consumption level that was most beneficial to the receiver.

We randomly divided subjects into one of five different groups, each of which received advice provided by a different strategy method of those listed in Section 5.3 (Silent, Receiver, Sender, SAP, MDP). The data obtained from the first three groups (Silent, Receiver and Sender) served to train the human models used by SAP and MDP.

5.6 Results

We begin by describing the fit-to-data of the various models we described in Section 5.2 using the data gathered in the Silent, Receiver and Sender groups.

Table 6 presents the fit-to-data of all the models which we tested using a tenfold cross validation on learning the parameters while minimizing the negative log-likelihood. As depicted in the table, *Hyper with learning* resulted in the best fit-to-data and was therefore our preferred method for modeling human behavior. The results presented in Table 6 cannot be directly compared to those in Table 2 since the domains are drastically different. Still, intuitively the latter are much lower due to the fact that in the road selection problem the subjects had to choose between four options while in the climate control domain the subjects had to choose between 10 different climate control energy consumption levels and the training data set was larger in the CCS domain.

Figure 2 presents the average performance for each of the groups, i.e. the average consumption level of the subjects (the lower the better). SAP significantly ($p < 0.001$ using the Mann-Whitney test) outperformed each of the other methods (including the MDP method).

Table 7 presents some additional data on each of the groups, including the number of subjects, the average comfort level, the average user score and the average acceptance rate (the percentage of times that the subject followed the exact advice). Unsurprisingly, the subjects in the *receiver* group yielded the best score, however, the acceptance rate of both the MDP method and SAP were very close to that of the subjects following the advice in the *receiver* group.

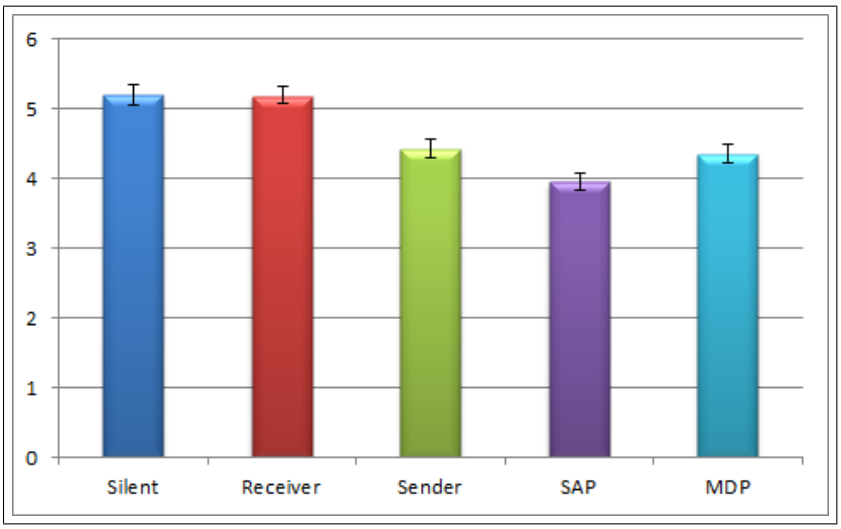

Fig. 2 Average energy consumption level for each of the treatment groups (the lower the better).

Table 7 Performance results of the interactions with people

method	no. of subjects	energy consumption	comfort level	user score	acceptance
Silent	57	5.202	8.744	12.289	–
Receiver	58	5.197	8.933	**12.67**	**36.7%**
Sender	47	4.437	7.843	11.264	19.5%
SAP	55	**3.952**	7.466	11.02	34.5%
MDP	55	4.361	7.996	11.652	33.8%

5.7 Discussion: Partially Informed and Ordered Actions Domains

In this section we introduced the climate control game and described a method for modeling the human decision making process in such a complex domain. We assimilated this model into SAP in order to provide advice to the user. The climate control game was designed in a manner that allowed construction of a complete MDP. Though the MDP method outperformed other baseline methods, SAP outperformed all methods including the MDP. It may seem surprising that SAP, which uses a relatively simple method outperformed the MDP approach. We explain this by the fact that the user model had to be simplified and discretized in order to suit the MDP. Furthermore, a human model may never be exact, therefore, overrelying on a noisy model as the MDP does, may cause the SAP, which only uses the human model as a guideline, to perform better.

As in the route selection domain, we assume that the sender has perfect knowledge of the world. In the route selection domain, we assumed that the receiver knows the expected travel time (and energy consumption) on each of the roads and in the CCS domain we assume that the receiver knows the expected energy consumption and user comfort level for each of the user actions. In real life, the agent may be required to collect a vast amount of data in order to predict these values well. If such data is unavailable in real life, the agent may consider some form of uncertainty. Previous work [6] which dealt with information disclosure (as

described in Section 2) was successfully extended in [7] while adding the assumption that the sender has some noisy observation.

If enough data can be gathered on a specific user, one may consider an improvement to SAP in which the parameters of the human model can be learned over time. Personalizing the advice may also be attributed to different cost functions associated with different users (e.g. some users may care more about energy consumption). However, explicitly changing SAP weights according to the user's acceptance rate may cause a user to feel manipulated.

6 Conclusions

In this paper we considered a two player game, in which an agent repeatedly supplies advice to a human user followed by an action taken by the user which influences both the agent's and the user's costs. We presented the Social agent for Advice Provision (SAP) which models human behavior combining principles known from behavioral science with machine learning techniques. We tested the performance of the SAP agent when interacting with human users in different types of domains. These domains differ in three main aspects. First, the amount of information a user has about the state of the world may be different whereby it may exist at some level or may not exist at all. Second, advice can affect the choice of a user at the global level by having a possible effect on all possible choices or it may have only a local effect on one action only. Third, the domains were different in the complexity of their state space making it possible to implement and solve the problem with an optimal solution or enabled only an approximate solution.

The results from all the experiments that were run in these different domains with different mechanisms for modeling the agent and human behaviors show the following consistent insights:

(1) SAP is successful - it outperforms all other agent implementations tested.

(2) SAP is simple to implement since its strategy for advice provision does not depend on the history of the interaction with its current user (modeled as hyper). Therefore, it is possible to deploy it in many common situations, where there is no knowledge about the number of times that users have used the system in the past.

(3) SAP is practical for real world scenarios since online advice may be provided, which demands very low CPU usage, i.e., SAP can be computed online with a time complexity of $O(|A|)$.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
2. Amazon. Mechanical Turk services. http://www.mturk.com/, 2010.
3. Ofra Amir and Yaakov Kobi Gal. Plan recognition and visualization in exploratory learning environments. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(3):16, 2013.
4. Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *FOCS*, pages 322–331, 1995.
5. A. Azaria, A. Hassidim, S. Kraus, A. Eshkol, O. Weintraub, and I. Netanely. Movie recommender system for profit maximization. In *RecSys*, 2013.
6. A. Azaria, Z. Rabinovich, S. Kraus, and C. V. Goldman. Strategic information disclosure to people with multiple alternatives. In *AAAI*, 2011.
7. A. Azaria, Z. Rabinovich, S. Kraus, and C. V. Goldman. Strategic information disclosure to people with multiple alternatives. *Transactions on Intelligent Systems and Technology (TIST)*, 5(4):64–86, 2014.

8. A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and Y. Gal. Strategic advice provision in repeated human-agent interactions. In *AAAI*, 2012.
9. S. Bonaccio and R. S. Dalal. Advice taking and decision-making: An integrative literature review and implications for the organizational sciences. *Org. Behavior and Human Decision Processes*, Vol. 101(2):127–151, 2006.
10. C. F. Camerer. *Behavioral Game Theory. Experiments in Strategic Interaction*, chapter 2. Princeton Unv. Press, 2003.
11. C. F. Chabris, D. I. Laibson, and J. P. Schuldt. Intertemporal choice. *The new Palgrave dictionary of economics*, 2:1–11, 2006.
12. LS. Chen, FH. Hsu, MC. Chen, and YC. Hsu. Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences*, 178(4):1032–1048, 2008.
13. A. Das, C. Mathieu, and D. Ricketts. Maximizing profit using recommender systems. *ArXiv e-prints*, page 0908.3633, 2009.
14. A. Deaton and C. Paxson. Intertemporal choice and inequality. *The Journal of Political Economy*, 102(3):437–467, 1994.
15. Avshalom Elmalech, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. When suboptimal rules. In *AAAI-15*, 2015.
16. Zhengzhu Feng, Richard Dearden, Nicolas Meuleau, and Richard Washington. Dynamic programming for structured continuous markov decision problems. In *the 20th conference on Uncertainty in artificial intelligence*, pages 154–161. AUAI Press, 2004.
17. Maier Fenster, Inon Zuckerman, and Sarit Kraus. Guiding user choice during discussion by silence, examples and justifications. In *ECAI*, pages 330–335, 2012.
18. Brian J Fogg. Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002:5, 2002.
19. Jon Froehlich, Leah Findlater, and James Landay. The design of eco-feedback technology. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2010.
20. Yaakov Gal, Sarit Kraus, Michele Gelfand, Hilal Khashan, and Elizabeth Salmon. An adaptive agent for negotiating with people in different cultures. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):8, 2011.
21. Noah Gans, George Knox, and Rachel Croson. Simple models of discrete choice and their performance in bandit experiments. *Manufacturing & Service Operations Management*, 9(4):383–408, 2007.
22. Claudia V Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 137–144, 2003.
23. Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. In *NIPS*, volume 1, pages 1523–1530, 2001.
24. Philip A. Haile, Ali Hortasu, and Grigory Kosenok. On the empirical content of quantal response equilibrium. *American Economic Review*, 98(1):180–200, 2008.
25. Chen Hajaj, Noam Hazon, and David Sarne. Ordering effects and belief adjustment in the use of comparison shopping agents. In *AAAI-14*, pages 930–936, 2014.
26. Chen Hajaj, Noam Hazon, David Sarne, and Avshalom Elmalech. Search more, disclose less. In *AAAI*, 2013.
27. W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
28. Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 256–265, 1998.
29. Catholijn M Jonker, Koen V Hindriks, Pascal Wiggers, and Joost Broekens. Negotiating agents. *AI Magazine*, 33(3):79, 2012.
30. Tanyoung Kim, Hwajung Hong, and Brian Magerko. Coralog: use-aware visualization connecting human micro-activities to environmental change. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4303–4308. ACM, 2009.
31. J. E. Lisman and M. A. P. Idiart. Storage of 7 ± 2 Short-Term Memories in Oscillatory Subcycles. *Science*, 267:1512–1515, March 1995.
32. Janusz Marecki, Sven Koenig, and Milind Tambe. A fast analytical algorithm for solving markov decision processes with real-valued resources. In *IJCAI*, pages 2536–2541, 2007.
33. Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.
34. G. A. Miller. The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81–97, March 1956.

35. T. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe. Analyzing the effectiveness of adversary modeling in security games. In *AAAI*, 2013.
36. D. Ormoneit and S. Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2):161–178, 2002.
37. Bhavik Pathak, Robert Garfinkel, Ram D Gopal, Rajkumar Venkatesan, and Fang Yin. Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems*, 27(2):159–188, 2010.
38. Dane Petersen, Jay Steele, and Joe Wilkerson. Wattbot: a residential electricity monitoring and feedback system. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 2847–2852. ACM, 2009.
39. James Pierce, Diane J Schiano, and Eric Paulos. Home, habits, and energy: examining domestic interactions and energy consumption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1985–1994. ACM, 2010.
40. F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
41. Shawn W Rosenberg, Lisa Bohan, Patrick McCafferty, and Kevin Harris. The image and the vote: The effect of candidate presentation on voter preference. *American Journal of Political Science*, pages 108–127, 1986.
42. Michael Rovatsos and Alexandros Belesiotis. Advice taking in multiagent reinforcement learning. In *AAMAS*, page 237. ACM, 2007.
43. Guy Shani, David Heckerman, and Ronen I. Brafman. An MDP-based recommender system. *J. Mach. Learn. Res.*, 6:1265–1295, 2005.
44. David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, pages 2164–2172, 2010.
45. Peter Stone and Sarit Kraus. To teach or not to teach?: decision making under uncertainty in ad hoc teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 117–124. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
46. Joanns Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European Conference on Machine Learning*, pages 437–448. Springer, 2005.
47. Wolfgang Wahlster and Alfred Kobsa. *User models in dialog systems*. 1989.
48. I. Yaniv and E. Kleinberger. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*, Vol. 83, No. 2:260–281, 2000.