

Agents for Automated Human Persuasion

Amos Azaria

Computer Science Department

Bar Ilan University

Ph.D. Thesis

Submitted to the Senate of Bar-Ilan University

Ramat-Gan, Israel

September 2014

This work was carried out under the supervision of:
Prof. Sarit Kraus and Prof. Yonatan Aumann
Department of Computer Science,
Bar-Ilan University

Acknowledgements

This thesis summarizes a wonderful journey of research. I would like to take this opportunity to thank all of those who made this journey possible, successful and fascinating. This work was in part supported by ERC #267523.

Contents

1	Automated Human Persuasion	1
1.1	Introduction	1
1.2	Related Work	5
1.3	Publications	8
I	Persuasion by Advice Provision	9
2	Multi-dimensional. Influential Advice	11
2.1	Introduction	11
2.2	The Volt Climate Control System	12
2.3	CARE	13
2.3.1	CARE Training Data	13
2.3.2	Energy Consumption Model	14
2.3.3	Human Comfort Level Model	14
2.3.4	CARE Method for Advice Provision	15
2.4	Training Data Collection Methods	15
2.4.1	Data Collection for Modeling Energy Consumption	15
2.4.2	Data Collection for Modeling Human Users	16
2.5	Graphical User Interface	17
2.6	Experimental Evaluation	19
2.7	Results	20
2.8	Discussion	22

CONTENTS

2.9	Conclusions	22
2.10	List of Notations	22
3	Recommending a set of actions.	25
3.1	Introduction	25
3.2	PUMA	27
3.2.1	Algorithm for the Hidden Agenda Setting	27
3.2.2	Algorithm for Revenue Maximizing	28
3.3	Experiments	31
3.3.1	Hidden Agenda Setting	33
3.3.2	Revenue Maximizing Settings	34
3.4	Discussion	38
3.5	List of Notations	38
4	Long-Term Influential Advice.	39
4.1	Introduction	39
4.2	The Model	40
4.3	The UMPA Approach	42
4.3.1	Modeling Diversity in People's Reactions	44
4.3.2	Predicting Advice Deviations	45
4.3.3	Estimating the Cost of an Advised Path	47
4.3.4	Searching for Good Advice	48
4.4	Experimental Evaluation	49
4.4.1	Methodology	50
4.4.2	Basic Results	51
4.4.3	UMPA Advice Algorithm Performance	52
4.5	Conclusions	54
4.6	List of Notations	55
5	Providing Advice in Repeated Interactions	57
5.1	Introduction	57
5.2	Choice Selection Processes	59
5.3	Route Selection Domain	62
5.3.1	Human Receivers as Multi-Armed Bandits	63

5.3.2	Agent Design for Senders	66
5.3.3	Empirical Methodology	69
5.3.4	Discussion	74
5.4	Climate Control Domain	75
5.4.1	Setting Description	76
5.4.2	Modeling Human Receivers	78
5.4.3	Agent Design for Sender	81
5.4.4	Experimental Settings	81
5.4.5	Experiments	82
5.4.6	Results	83
5.4.7	Discussion: Partially Informed and Ordered Actions Domains	84
5.5	Conclusions	86
5.6	List of Notations	87

II Persuasion by Information Disclosure and Presentation 89

6 Which Information to Disclose? 91

6.1	Introduction	91
6.2	The Information Disclosure Game with Two-Sided Uncertainty	93
6.3	Solving Information Disclosure Games with Two-Sided Uncertainty	94
6.3.1	Mathematical Program	94
6.3.2	Finding an Optimal Policy	96
6.4	People Modeling for Disclosure Games in Multi-attribute Selection Problems	98
6.4.1	Multi-attribute Road Selection Problem with Two Sided Uncertainty	99
6.4.2	The Sandwich Game	100
6.4.3	Hypothesis	100
6.4.4	Non-monetary Utility Estimation for the Road Selection Problem with Two-Sided Uncertainty	100
6.4.5	Non-monetary Utility Estimation for the Sandwich Game with Two- Sided Uncertainty	102
6.5	Experimental Evaluation	102
6.5.1	Experimental Design	103
6.5.2	Human Subjects	105

CONTENTS

6.5.3	Experimental Results	106
6.5.4	Results of the Multi-attribute Road Selection Game with Two-sided Uncertainty	107
6.5.5	Sandwich Game Results	110
6.5.6	Deciding between LUQA and GTBA	111
6.6	Conclusions	112
6.7	Proofs of Theorems Concerning Message Space	113
6.7.1	Proof of Theorem 1	113
6.7.2	Proof of Theorem 2	115
6.8	List of Notations	115
7	Persuasion Method Matters	119
7.1	Introduction	119
7.2	Human Decision Making Under Uncertainty Hypotheses	122
7.2.1	Expected Utility Hypothesis	122
7.2.2	Prospect Theory	122
7.2.3	Bracketing	124
7.3	Prospect Presentation Problem	125
7.4	An Agent for the Prospect Presentation Problem	126
7.4.1	Solving the Prospect Presentation Problem	126
7.4.2	Decision Policy Modeling in APPP	127
7.5	Evaluation	130
7.5.1	Experimental Setup	130
7.5.2	Results	132
7.5.3	Discussion	135
7.6	Conclusion	135
7.7	List of Notations	136
8	Final Remarks	139
	Bibliography	141

List of Figures

2.1	A screen-shot with additional energy consumption information provided by CAREless (the circle in the bottom left corner).	18
2.2	A screen-shot of the GUI with CARE's advice. In this example, the driver set the temperature to $18^{\circ}C$ (rather than $21^{\circ}C$ as advised by CARE), the fan to 4 (rather than 1 - as indicated by the purple line), the air delivery to face and feet (rather than face-only) and the mode to "comfort" (rather than "eco"). This resulted in an energy consumption level of 63% of the maximal energy consumption level (right green circle), rather than only 25% if the driver would have followed CARE's advice (left purple circle).	18
2.3	The mean energy consumption level of the subjects who received advice from CARE and CAREless, compared to the mean energy consumption levels of the subjects when they did not receive any advice.	21
2.4	The energy consumption level of the climate control system of each subject when receiving advice from CARE compared to the baseline of the same subject when not receiving any advice.	21
3.1	A screen-shot of a subject selecting movies he liked	31
3.2	Recommendation page screen-shot	32
3.3	An example of PUMA's selection process	35
3.4	Average revenue per system (in dollars)	37
4.1	Path selection problem visualized in a small maze	41
4.2	A second example of a path and a cut	43

LIST OF FIGURES

4.3	Average agent's costs	52
4.4	Average users' costs	53
4.5	Users' satisfaction and trust	54
5.1	Average fuel consumption for each of the treatment groups (the lower the better).	73
5.2	Average energy consumption level for each of the treatment groups (the lower the better).	85
6.1	System utility in road game Γ_{ρ}^1 . The center gained a significantly higher utility from the actual users than the utility it would have gained if all of the users were rational ($p < 0.001$)	107
6.2	User utility in road game Γ_{ρ}^1 . The actual drivers gained a significantly lower utility, on average, than they would have gained if they all would have acted rationally ($p < 0.001$).	107
6.3	System utility in road game Γ_{ρ}^2 . The center performed significantly better when using LUQA rather than GTBA ($p < 0.05$).	107
6.4	System utility for LUQA in road game Γ_{ρ}^2 . LUQA performed significantly better when it received full information ($p < 0.05$).	107
6.5	User utility in sandwich games. The difference between a fully rational seller and the actual human sellers is minor and not statistically significant.	110
6.6	System utility in sandwich game Γ_{σ}^2 . The difference between the organizer utility when using LUQA and when using GTBA is minor and not statistically significant.	110
6.7	System utility in the sandwich game. The difference in the organizer's utility between actual users and the utility it would have gained if all of the users were rational is minor and not statistically significant.	112
7.1	A subject facing a set of prospects presented separately.	131
7.2	A subject facing a set of prospects presented in the combined mode.	132
7.3	Average score obtained with each of the methods.	134

List of Tables

2.1	List of Notations	23
3.1	Function forms for the considered functions. α and β are non-negative parameters and $r(m)$ is the movie rank.	28
3.2	Demographic statistics for the hidden agenda setting	33
3.3	Coefficient of determination for functions tested for the hidden agenda setting .	34
3.4	The percentage of subjects who wanted to watch each movie, average promotion gain, overall satisfaction and the percentage of movies marked as good recommendations	35
3.5	Demographic statistics for the revenue maximizing setting	36
3.6	Coefficient of determination for the functions tested for the revenue maximizing settings.	36
3.7	The percentage of subjects who would pay for a movie, the average revenue and the overall satisfaction.	37
3.8	List of Notations	38
4.1	List of Notations	56
5.1	Fit-to-data of different receiver models (the lower the better)	71
5.2	Settings used in the route selection domain.	72
5.3	Simulation results comparing agent strategies	73
5.4	Performance results of agents interacting with people. The selfishness rate equals w in Equation 5.15	74

LIST OF TABLES

5.5	Fit-to-data of different receiver models in the climate control domain (lower is better)	84
5.6	Performance results of the interactions with people	85
5.7	Notation list	88
6.1	Seller types	105
6.2	Observation table in the sandwich game	106
6.3	Mean square error of modeling human decision-making	109
6.4	List of Notations	117
7.1	Average performance of APPP compared to the other agents	134
7.2	List of Notations	137

List of Abbreviations

- AMT: Amazon Mechanical Turk
- APPP: Agent for the Prospect Presentation Problem
- CARE: Climate control Adviser for Reducing Energy consumption
- CCS: Climate Control System
- CPT: Cumulative Prospect Theory
- CPU: Central Processing Unit
- ES: Exponential Smoothing
- EUH: Expected Utility Hypothesis
- GPS: Global Positioning System
- GTBA: Game Theory Based Agent
- GUI: Graphical User Interface
- HA: Hidden Agenda
- Hyper: Hyperbolic discounting
- LUQ: Linear combination for social Utility and Quantal response
- LUQA: LUQ based Agent
- LWU: Linear Weighted-Utility

List of Abbreviations

- MAB: Multi-Armed Bandit
- MCS: Monte Carlo Sampling
- MDP: Markov Decision Process
- PUMA: Profit and Utility Maximizer Algorithm
- QRE: Quantal Response Equilibrium
- RM: Revenue Maximizing
- SAP: Social agent for Advice Provision
- SP: Subgame Perfect
- TV: Television
- UMPA: User Modeling for Path Advice
- USA: United States of America

Abstract

With the rise in advanced technologies, computer systems that take an active role in human's decision-making tasks have become more popular. Frequently, a system and a human user do not share the exact same goal. This thesis focuses on such interactions in which, on the one hand, the systems and the humans do not share the exact same goal, but on the other hand, their incentives do not contrast either (i.e. their goals do not compete with one another). The objective of this thesis is to develop automated agents for human persuasion. Many real life interactions may benefit from such agents. In the domain of health care, for example, a therapist may seek to encourage a patient to exercise or take a certain medication which the patient may try to avoid. An agent may help in encouraging the patient. In the automobile environment, an agent may provide advice to drivers with respect to cruise speed or acceleration ratio in order to reduce the human environmental footprint. In the domain of on-line learning and agent may encourage students to complete a course and avoid dropping out in the middle.

Our approach for composing such automated agents relies heavily on modeling human behavior. Building such a human model may be very challenging, since humans are susceptible to various psychological effects and their behavior may rely on unanticipated or unknown factors. To overcome this challenge we based our human model on literature from social science, psychology and human decision-making studies. We collected data on a specific domain and used machine learning techniques in order to learn parameters which explain best human behavior.

This thesis deals with three different types of persuasion methods:

- Advice provision: The agent may advise the human to take a certain action. In this case, the system may either be exposed to more information than the human, or it may use its computational advantage. The agent's advice may influence the human when

Abstract

considering which action to take and thus the agent should provide advice which will encourage the human to take an action preferable to the agent. Note, that the agent does not necessarily want the human to take the action which it advises, but rather influence the human's choice of action. This is the most common case in real life scenarios and thus a considerable part of this thesis focuses on this case.

- Information disclosure: The agent has information unknown to the human, and can reveal full or partial information in order to encourage the human to take a certain action. The agent will reveal information which will encourage the human to take an action preferable to the agent.
- Presentation method: The agent has specific data to show to the human, but may still choose between different forms of presenting this data. The agent must present the data in a way that will encourage the human to take an action which is preferable to the agent.

The work presented in the thesis was based on experiments relying on results from hundreds of people for every domain considered. This high number of subjects allowed us to build more accurate human models and evaluate the methodology by means of an extensive study.

Automated Human Persuasion

1.1 Introduction

Computer systems are increasingly being deployed in platforms that involve interactions with people as well as with other computer agents. Many of these scenarios require computer agents to generate advice to their human users about what choices to make. Such settings arise in application domains like coaching, rehabilitation and route-navigation.

Although in general these interactive systems are cooperative, users and machines may have different interests. In this thesis, we assume that the automated agent and the human user have different goals, however, we consider non-competitive environments (i.e. the gain of one party is not necessarily at the expense of the other). In particular, we study automated agents interested in persuading their users to perform actions that increase the agent's utility. Consider a route selection domain where an automatic system suggests commuting routes to a human driver. Both participants in this setting share the goal of getting the driver from home to work and back. However, each participant also has its own incentives. The driver wishes to choose the route that minimizes the commuting time, while the system which may represent the government may prefer the driver take a longer route that emits fewer pollutants, or does not pass near schools and playgrounds.

Machines can try to persuade their users to perform certain actions by implementing different methods. In this thesis we focus on three different methods for persuasion. The first part of this thesis focuses on agents providing advice, i.e. agents which may advise their users to perform certain actions. We examine how to automatically generate advice that will encourage users to choose actions preferred by the system (agent). The second part of this thesis considers

1. AUTOMATED HUMAN PERSUASION

agents who provide information about the state of the world (unknown to the user, but relevant to his decision). This information is revealed in a manner which intends to persuade the human to take certain actions beneficial to the agent. In the last chapter we consider an environment in which an agent may consider different methods to present information to a user. We compose an automated agent which chooses a method that is expected to encourage humans to take a certain action.

Throughout this thesis we used the following methodology for composing automated agents: First, we build a formal model which is phrased as an optimization problem for the agent. Then, based on literature from social science, psychology and human decision-making studies, we model human behavior and build a general human model. We then collect relevant data and build an agent which uses this data along with machine learning techniques in order to learn the parameters that best suit the models. Then, based on the human model and using various optimization methods, the agent finds the action which is most likely to cause the human to choose an action which is best for the agent.

Efficient interaction with humans requires understanding and modeling their behavior. For example, while equilibrium strategy is theoretically considered the most rational one, agents using such strategies often perform poorly in practice [1, 2, 3]. Since humans commonly do not use equilibrium strategy themselves, replying with such a strategy can be suboptimal.

However, building this human model may be very challenging. First, it is known that, in many cases, people follow suboptimal decision strategies. This bounded rational behavior [4] is attributed to: sensitivity to the context of the decision-making; lack of knowledge of the user's own preferences; the effects of complexity; the interplay between emotion and cognition and the problem of self-control. Furthermore, people discount the advice they receive from experts[5] and it was shown that if the adviser has a monetary stake in the advice being followed, people will follow its advice even less [6]. Finally, the learned model should be generalized to new environments as well as different people. To face these challenges we integrate machine learning and psychological models for predicting human response to advice.

This thesis is composed of two parts, the first, "Persuasion by Advice Provision", and the second part is "Persuasion by Information Disclosure and Presentation". In the first part of this thesis, we represent the interaction as a two player game, which includes a sender and a receiver. Both players have their own utility (or cost) functions. The receiver has a set of actions, which it will be required to choose from, A . The sender, after observing the state of the world, may advise the receiver to take a certain action $d \in A$; this advice may influence the

receiver's choice. After receiving the sender's advice, the receiver may either take the action advised by the sender, take a different action unrelated to the advice, and in some domains, the receiver may choose to take an action which is not identical to the advice, but is influenced by it. The outcome to both the receiver and the sender is determined only by the receiver's action and the state of the world, i.e., the sender's advice may have only an indirect impact on both parties' outcomes. In each of the chapters in this part we consider different models which enhance this basic model and result in a more complex and realistic interaction.

In Chapter 2 we consider a setting in which the receiver is assumed to be a driver in an automobile who needs to set the Climate Control System (CCS). The driver receives advice from a system seeking to reduce energy consumption. The driver controls a set of parameters such as the CCS's temperature, fan strength and air delivery method. In this setting, after receiving advice from the system, the driver may follow the advice exactly, ignore it completely, or perform some kind of compromise between the advice and his own initial preferences.

In Chapter 3 we consider a setting in which the receiver chooses a set of actions ($a \subset A$) and the sender recommends a set of actions as well. In this setting, the receiver may only select a set of actions from the recommendations provided by the sender. This model is in-fact similar to models used in recommender systems. Specifically, we consider a movie recommender system. We consider two different utility functions for the system. In the first, each movie is associated with some value and the system needs to maximize the value of all movies selected by the user. In the second setting, the system needs to maximize its expected revenue.

In Chapter 4 we consider a setting in which the sender provides a single yet long lasting advice. The receiver, which is a player in a game, must find a path on a grid from a starting point to a destination. There are many paths, but the player is told to find a short path. The system advises the player to take a specific path. The player may follow the path part of the way, perform some short-cuts and return to the path at a later stage. We calculate the probability for each such short-cut and show how to calculate a path which will be most beneficial to the system.

The last chapter in this part, considers repeated interactions between the sender and the receiver, i.e. the basic interaction is continuously repeated with some (high) probability. In this scenario, the sender must consider the long term impact of its advice, as providing bad advice in the current round, may cause the receiver to ignore the advice in future rounds. We consider two different domains. The first, is a route selection domain in which a driver must select, day after day, from a set of routes. In the second domain, a driver sets the power of the air

1. AUTOMATED HUMAN PERSUASION

conditioning system in the car. We introduce a method for providing advice for such settings, which is based on considering a social utility of both the driver and the system. We show that this method outperforms other possible methods such as Monte Carlo sampling and a Markov Decision Process (MDP).

The second part of this thesis entails persuasion by information disclosure and presentation. In chapter 6 we consider a game consisting of a sender (system) and a receiver (human), in which the sender may reveal partial (but truthful) information in order to persuade a receiver into taking a certain action. We consider two different domains; the first is a road selection domain, and the second consists of what we call the sandwich game. In the road selection domain the receiver (driver) must choose a road from a set of roads. These roads have different traffic states and some are associated with a toll. The sender (system) and the driver have different utility functions which depend on the state of the traffic and the toll associated with the road chosen by the driver. The exact traffic state on the roads is known only to the system. The system may provide either partial or full information to the driver. The system's goal is to maximize its (expected) utility. In the second domain, the sandwich domain, the receiver takes the role of a seller who needs to plan in advance how many sandwiches to prepare for a conference. The sender plays the role of the conference organizer and has a different utility function than the seller. Both utility functions depend on the size of the conference (known a priori only to the organizer) and on the number of sandwiches prepared by the seller. The organizer may disclose some information to the seller. In both domains, the sender himself may have some error in his observation. We present two approaches for solving the problem for the sender. The first assumes that the receiver is fully rational—we solve the problem for the sender under this assumption. The second approach builds a human model using principles from social science and uses this model to find the best information for the sender to disclose. We show that the performance of the different approaches varies, depending on properties associated with the different domains.

The last chapter focuses on presenting identical information using different presentation modes. We consider a lottery domain, in which a user faces a set of lotteries (some with negative payoff) and may choose to either accept them all together or reject them all. These lotteries may be presented to the user using different presentation methods. We assume that a system gains utility every time a user accepts the set of lotteries. We present an automated agent that models human decision making in such settings and then calculates the expected utility for

the system from each of the forms of representation. The agent then selects the presentation method which yields the highest expected utility for the system.

Most of the work presented in this thesis was conducted using Amazon’s Mechanical Turk platform, which is a crowd sourcing web service that coordinates the supply and demand of tasks which require human intelligence to complete. Amazon Mechanical Turk enabled us to recruit hundreds of people for every domain we considered¹. It allowed us to build more accurate human models and evaluate the methodology by means of an extensive study. Our experience in running experiments on Mechanical Turk demonstrated that almost all subjects considered our tasks seriously. Our experience confirms other studies [7] about the viability of this medium for empirical research. One exception in this thesis, where we did not rely on Amazon’s mechanical Turk but rather on real drivers appears in the second chapter. In this case drivers were provided advice from an agent regarding the settings of a climate control system in a real car.

1.2 Related Work

Persuasion of humans by computers or technology has raised great interest in the literature. In his book [8], Fogg surveyed many technologies that try to persuade humans, and analyzed the main properties required for such persuasion technologies to be successful. One example of such a persuasion technology (pg. 50) is an exercise bicycle connected to a TV (“Telecycle”). In this system, as one pedals at a higher rate the image on the TV becomes clearer. Consequently the Telecycle encourages humans to exercise at higher rates. Fogg also described different methods for persuasive systems such as a social actor - an example is the Banana-Rama slot machine which has characters that celebrate every time the gambler wins. Fogg stated that in order to be persuasive, a system must be credible, i.e. both in regards to trustworthiness and expertise. Froehlich et al. [9] surveyed many persuasive technologies with the goal of reducing environmental impact.

Previous work on advice provision and information disclosure spans the computational and social sciences disciplines. Game theory researchers have studied persuasion games [10, 11], in which a “sender” player needs to decide how much information to disclose to a “receiver” player to influence the receiver’s strategy in a way that will benefit the sender [12, 13, 14].

¹ All of the study procedures were authorized by the ethics review board of Bar Ilan and Ben Gurion Universities.

1. AUTOMATED HUMAN PERSUASION

Renault et al. [15] considered repeated interactions that follow a Markov chain observed solely by the sender. After observing the state, the sender sends a message to the receiver revealing partial information, full information or no information on the state. Additionally, the receiver cannot observe its utility until the end of the multi-period interactions. They studied the trigger equilibrium that is built on some core strategy and allows deviations. In equilibrium, the sender assumes some behavior on the receiver's action and provides advice accordingly. The sender records the receiver's actions and if the receiver deviates from the behavior assumed by the sender, the sender stops providing information to the receiver. The receiver on the other hand, listens to the advice as long as it is proved to be sufficiently accurate, however, once the sender deviates and sends inaccurate advice, the receiver ignores any pursuant advice and uses the best response under his given knowledge.

Models for predicting users' ratings have been proposed that are used by recommendation systems to advise their users (See Ricci et al. [16] for a review). It has been shown that recommender systems, in general, are beneficial for the providing business [17], since they help the user make a purchase. Most works in this realm do not explicitly try to maximize the system's revenue, but only consider the utility of the user (which, as stated, indirectly increases the system's revenue). Chen et al. [18] developed a recommender system that tries to maximize product profitability. Chen et al. assumed the usage of a collaborative filtering recommender system which, as part of its construction, provides a theoretically-based probability that a user will purchase each item. Their system multiply this probability by the revenue from each item and recommend the items that yield the highest expected revenue.

Pathak et al. [19] studied the cross effects between sales, pricing and recommendations on Amazon books. They show that recommendation systems increase sales and cause price changes. The recommendation systems that they considered are price independent, and the effect on prices is only indirect—items which are recommended more are bought more, which affects their price (the pricing procedure used in their data takes popularity into account).

Das et al. [20] provide a mathematical approach for maximizing business revenue using recommender systems. However, they assumed that as long as the recommendations are similar enough to the customer's own ratings, the customer is likely to follow the recommendations. Therefore, Das et al. did not model the actual drop in user acceptance rate as the item becomes less relevant or as the item price increases, as in our work. Similarly, Hosanagar et al. [21] used a mathematical approach to study the conflict which a business confronts when using recommender systems. On one hand the business would like to recommend items with higher

revenue (margins), but on the other hand it would like to recommend items which the users are more likely to buy. Hosanagar et al. show that in order to increase its total revenue, the business must balance between these two factors. Shani et al. [22] used a discrete-state MDP model to maximize the system's utility function, taking into account the future interactions with their users. In their work, the system may, for example, decide to recommend a game console, since, if purchased, the user is likely to purchase many games for it in the future.

Route or path selection, which is included in several chapters in this thesis, has become one of the most prominent applications of computer assisted guidance (see a survey in [23]). In fact, route guidance systems using GPS have become pervasive over the years, thanks to the significant research effort in addressing both the cognitive limitations and the range of individual preferences of human users (e.g. [24, 25]). Many of the challenges in the development of route guidance systems stem from the high variance among individuals regarding their evaluation and acceptance of route advice. This variance makes it important to tailor route advice and guidance to a specific user. To this end, a wide range of machine learning techniques have been used to capture and utilize user routing preferences (e.g. [25]). Instead of tailoring routes to users, we model user attitudes towards route advice such that the choices made by the users, after being given advice, will be beneficial to the agent. In addition, we assume that the system and the user have different goals.

There has been some work on driver acceptance of unreliable route guidance information [26]. Antos and Pfeffer [27] designed a cooperative agent that uses graphical models to generate arguments between human decision-makers and computer agents in incomplete information settings. They used a qualitative approach that does not model the extent to which people deviate from computer-generated advice. Other works have demonstrated a human tendency to accept advice given by an adversary in games [6]. Some theoretical analysis suggests this behavior to be rational [28]. To some extent, these results were used in the framework of large population traffic manipulation (either by explicitly changing the network topology or by providing traffic information, e.g. [29, 30]).

It is well known that the way information is presented also may have an impact on the human decision-making process. Rosenberg et al. [31] studied the effect that photographs of political candidates have on voters' perception and show indeed that these images significantly affect their votes. Seuken et al. [32] studied how the UI design choices for markets, such as the number of choices and the use of dynamic prices, affect users' abilities to make good economic decisions. Fenster et al. [33] designed an agent that influences human decision-making in a

1. AUTOMATED HUMAN PERSUASION

conversational setting. In their work they 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 tried to convince the human about a location by providing examples for her to emulate, or by providing justifications for a certain choice.

1.3 Publications

Some of the results that appear in this thesis were published in:

1. A. Azaria, S. Kraus, C. V. Goldman, and O. Tsimhoni, Advice Provision for Energy Saving in Automobile Climate Control Systems, In Proceedings of IAAI, 2014 [34] and AI Magazine [35]. These papers are the basis for chapter 2.
2. A. Azaria, A. Hassidim, S. Kraus, A. Eshkol, O. Weintraub, and I. Netanel, Movie Recommender System for Profit Maximization, In Proceedings of RecSys, 2013 [36]. This paper is the basis for chapter 3.
3. A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and O. Tsimhoni, Giving Advice to People in Path Selection Problems, In Proceedings of AAMAS, 2012 [37]. This paper is the basis for chapter 4.
4. A. Azaria, Z. Rabinovich, S. Kraus, C. V. Goldman, and Y. Gal, Strategic Advice Provision in Repeated Human-Agent Interactions, In Proceedings of AAI, 2012 [38] and Journal of Autonomous Agents and Multiagent Systems [39]. These papers are the basis for chapter 5.
5. A. Azaria, Z. Rabinovich, S. Kraus, and C. Goldman, Strategic Information Disclosure to People with Multiple Alternatives, In Proceedings of AAI, 2011 [40] and Transactions on Intelligent Systems and Technology (TIST) ACM Journal, 2014 [41]. These papers are the basis for chapter 6.
6. A. Azaria, A. Richardson, and S. Kraus, An Agent for the Prospect Presentation Problem, In Proceedings of AAMAS, 2014, [42]. This paper is the basis for chapter 7.

Part I

Persuasion by Advice Provision

Multi-dimensional. Influential Advice

2.1 Introduction

We begin by presenting the simplest scenario, in which an automated agent may provide advice. In this chapter, we consider a one shot advice provision. The advice is multi-dimensional, i.e., includes several parameters, and is influential, i.e., the user may decide not to follow the exact advice but will still take it into account when deciding which action to perform.

Since we had the opportunity to work with General Motors, we use the domain of advice provided in automobile systems. We consider a Chevrolet GM Volt car in a summer environment, whereby the driver would like to turn on the climate control system to cool down the very warm car in order to drive comfortably. An agent advises the driver how to set the car's climate control system. In this scenario the computer agent and human user do not share the exact same goal. While the agent may care mainly about the car's energy consumption, the driver, on the other hand, is usually more interested in his own comfort level while less interested in the car's energy consumption. Thus, the agent faces the challenge of providing advice that will reduce energy consumption while taking into consideration the driver's comfort level, i.e., advice that will persuade the driver to set the system settings such that he reduces the energy consumption of the system.

To provide efficient advice, our agent builds two models. The first, is a model of the preferences of the driver, estimating his comfort level in a given climate control setting. The second is an estimate of the energy consumption of a given setting. Both the drivers' preferences and the car's energy consumption are very noisy and difficult to estimate. Both models were built using data collected by running experiments in the Chevrolet Volt. The data for building

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE

the drivers' model was collected from only 15 participants. Based on the constructed models we formalized the optimization problem of the agent, which wishes to minimize the energy consumption while maintaining a reasonable level of estimated comfort. We also designed a Graphical User Interface (GUI) that allows the agent to provide the advice in a convenient and attractive way for the driver. In order to evaluate our agent, we ran experiments with 49 human users who were required to set the climate control parameters of the Chevrolet Volt when it was very hot outside. We tested three different types of advice provision methods. The first *no advice* method did not provide any advice and presented the subjects with an interface similar to the original Volt climate control system. The second *energy info* method provided the subjects with information regarding their current energy consumption level (based on the energy consumption model we built). The third *agent* method provided the subjects with advice on how to set the climate control system, along with the energy consumption information. We show that, on average, the subjects consumed less energy when interacting with the *energy info* or *agent* method vs. the *no advice* method. However, statistically significant differences were found only when comparing the *agent* method with the *no advice* method. When using our agent, the subjects saved approximately 17% of the energy consumption of the climate control system.

2.2 The Volt Climate Control System

The study in this chapter was based on the Volt's climate control system. In this system the driver can control the settings s as described in this tuple (T, F, D, M) where: **Temperature** (T) is associated with a temperature in Celsius and can receive values between 16 and 35 degrees; **Fan strength** (F) is associated with the fan blower and can receive values between 1 and 6; **Air delivery** (D) may either be set to face (in which D is set to 0) or face and feet (in which D is set to 1); and **Mode** (M) may either be set to "eco" (when M is set to 0) or to "comfort" (when M is set to 1). According to the Volt's user manual, the 'eco' mode tries to reduce energy consumption, while the "comfort" mode aims at maximizing the user's comfort level. Given a setting s we use subscript s_T to refer to the temperature in that setting, s_F to refer to the fan strength, s_D for the air delivery and s_M for the mode of the setting.

2.3 CARE

In this section we present our Climate control Adviser for Reducing Energy consumption (CARE). CARE requires the composition of two models, one for modeling the climate control's energy consumption as a function of its settings and the other for modeling human comfort level as a function of the climate control's settings. CARE uses these models in order to provide a driver with advice regarding the settings of the climate control system, while taking into account both the expected energy consumption and the expected comfort level. The comfort level is captured by a number between 1 and 10 where:

- 1: "I'm very uncomfortable; I would not be willing to drive under these conditions.";
- 3: "I'm uncomfortable, but I might be willing to compromise.";
- 5: "Reasonable, I would be willing to drive under these conditions.";
- 7: "I'm comfortable; I would like to drive under these conditions."; and
- 10: "I'm most comfortable, I would be happy to drive under these conditions."

2.3.1 CARE Training Data

Constructing CARE requires two sets of training data: ψ_e and ψ_c . ψ_e is used to train the parameters for the energy consumption model. It is composed of a tuple with the following format for every instance i : $\psi_e^i = (e, T, F, D, M, E, I)$ where e is the energy consumption level, given the other parameters; T, F, D and M are the variables set on the climate control system; E is the external temperature as displayed in the dashboard; and I is the internal temperature as measured with a manual thermometer located between the 2 front seats.

ψ_c is used to train the parameters for the comfort model. It is composed of a tuple with the following format for every instance i : $\psi_c^i = (c, T, F, D, C, E, I)$ where c is the comfort level reported by the subject, given the other parameters; C is the initial comfort level, i.e. the comfort level reported when the driver enters the car; and all other parameters are as described in ψ_e ¹.

¹Notice that the mode, M , does not appear in the comfort level model; this attribute will be explained later.

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE

2.3.2 Energy Consumption Model

We modelled the energy consumption of the climate control system based on the following equation:

$$e(T, F, D, M, E, I) = (w_1 \cdot (-T) + w_2 \cdot F + w_3 \cdot D + w_4 \cdot E + w_5 \cdot I) \cdot ((1 + w_6) \cdot M) \quad (2.1)$$

where w_1, w_2, \dots, w_6 are parameters learned by the model. This form of function assumes that all variables except the climate mode have a linear impact on the final energy consumption. The climate mode is assumed to have a multiplicative impact on the total energy consumption, since in the “comfort” climate mode, all of the climate control components seem to work harder and thus consume more energy. This form of function was compared to other forms and yielded the best fit to the data collected¹. All parameters are assumed to be positive, except w_3 which models the impact of air delivery on energy consumption. w_3 was allowed to obtain negative values and in fact it did end up with a negative value. We used the training data, ψ_e , and searched for the parameters w_1, w_2, \dots, w_6 which maximize the likelihood of the training data (maximum likelihood estimation).

2.3.3 Human Comfort Level Model

CARE’s model for the human comfort level is based on the following equation:

$$c(T, F, D, C, E, I) = v_0 - v_1 \cdot T + v_2 \cdot F - v_3 \cdot F^2 - v_4 \cdot D + v_5 \cdot C - v_6 \cdot E - v_7 \cdot I \quad (2.2)$$

where v_0, v_1, \dots, v_7 are parameters learned by the model. F^2 tries to capture the effect of the noise created by the fan, which is super-linear in the fan’s level. The human comfort level model assumes that the human comfort level is a linear combination of all of the parameters that the human faces (assuming that F^2 models the noise effect). This assumption is common in the literature [40, 43]. According to the car’s user manual, the ‘eco’ mode is supposed to save energy, therefore, CARE never recommended to set the mode to “comfort”, and we only gathered data on subjects’ comfort level when using the ‘eco’ mode. For that reason, the human model does not take the mode into account, and only tries to predict the comfort level for

¹Some of the other functions that were tested included one or more of the following modifications to the above function: the use of M as an additive variable; F as having a multiplicative impact or T as having an impact depending on its offset from I or E .

when the mode is set to “eco”. We used the training data, ψ_c , and searched for the parameters v_0, v_1, \dots, v_7 which maximize the likelihood of the training data. Note that the initial comfort level (C) may change from person to person. This will cause the expected comfort level to vary among people, and thus also the advice provided by CARE may vary among different people. This causes the advice to be personalized, i.e. different drivers may receive different advice.

2.3.4 CARE Method for Advice Provision

Given both the energy consumption model and the human comfort level model, CARE provides the driver with advice regarding the settings of the climate control system. Given the external temperature (E), the internal temperature (I) and the initial comfort level (C), CARE provides the driver with advice, $a(E, I, C) \in S$, that yields an expected comfort level of at least 7 while minimizing the expected energy consumption of the climate control system. CARE only considers advice in which the mode is set to “eco” (i.e. M is set to 0). Comfort level 7 was chosen as the minimal target comfort level since a comfort level of 7 means that the driver is comfortable. More formally, CARE provides advice such that:

$$\begin{aligned} a(E, I, C) = \arg \min_{s \in S} e(s_T, s_F, s_D, M, E, I) \text{ s.t.} \\ s_M = 0; c(s_T, s_F, s_D, C, E, I) \geq 7 \end{aligned} \quad (2.3)$$

where $e(\cdot)$ is obtained from Equation 2.1, and $c(\cdot)$ is obtained from Equation 2.2. Since the search space was small ($|S|$ was much smaller than 1000), we performed an exhaustive search to find the optimal advice. However, in a climate control system with additional variables, CARE may consider a more efficient method of search.

2.4 Training Data Collection Methods

We used the following methods to gather the necessary data in order to train CARE’s two models.

2.4.1 Data Collection for Modeling Energy Consumption

We collected data on energy consumption directly from the car in order to train the energy consumption model (ψ_e) while the climate control system was on. We conducted a total of 120 measurements. Each measurement was a 10-minute duration. We let the car warm up (and the compressor cool down) for 10 minutes between consecutive measurements. We conducted

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE

these measurements for various temperatures, starting at $T = 16$ up to $T = 26$, and sampled different values for all of the different variables. Many of our measurements (36) focused on the range between $T = 20$ and $T = 25$, when $M = 0$ (“eco” mode), which is the natural range for candidates for the advice, and the function must be the most accurate at those variables.

Evidently the fan strength, F , had a greater impact on the energy consumption level than the temperature, T . That is, increasing the fan by one unit consumed more energy than reducing the temperature by one degree Celsius. The raw data we collected strengthened this result, as we observed that when the fan was set to a higher level, not only did the blower consume more energy, but so did the compressor. Both the external and internal temperatures (E and I) had a milder effect on the energy consumption level. Interestingly, the air delivery, D , had a negative impact on the energy consumption level, i.e. when the air delivery was set to face and feet, the climate control system consumed less energy than when set only to face. The final function obtained was:

$$e(T, F, D, M, E, I) = (-0.0095T + 0.016F - 0.003D + 0.005E + 0.005I) \cdot (1.17M). \quad (2.4)$$

2.4.2 Data Collection for Modeling Human Users

Data collection for the human model (ψ_c) requires human subjects, and thus is difficult to gather. We therefore had to assure that as many instances as possible are in the range that is most likely to be used by CARE. Merely randomly selecting different settings may not have yielded adequate information for training the human model.

We recruited 15 subjects for training the Human Model, of which 4 subjects were females and 11 were males. The subjects’ ages ranged from 21 to 73, with a mean of 30 and a median of 27. All subjects live in Israel. The subjects were first asked to fill out a questionnaire on demographic information. Then the comfort level scale was explained to them.

The subjects entered the car and sat in the driver’s seat with their hands on the steering wheel and set the vents to point in their direction. While the climate control system was still off, the subjects were asked to rate their comfort level. The subjects were told how to operate the climate control and set it so that they would feel most comfortable. These settings were left on for 4 minutes. The subjects were asked for their comfort level and were required to explain why they had chosen that level. The subjects then exited the car and the car was left to warm up for 4 minutes. The subjects then returned to the car and the experiment operator set the next

setting for them and waited 4 minutes. The subjects had to report and explain their comfort level and had to wait 4 minutes outside the car. These stages were repeated for a total of 8 different settings for every subject (resulting in 120 instances for all of the 15 subjects).

The subjects' comfort levels seem to have been influenced mostly by the temperature that was set on the climate control system, T . The fan, F , also had an impact on the comfort level, though not as strong as the temperature. Recall that the opposite happened when modeling the energy consumption level (this result motivated CARE to advise settings with the fan set to low values). Most subjects reported a reduced comfort level when the fan was too strong (some reported that the noise was what bothered them). The other parameters seemed to have a milder impact on the subject's comfort level. The final formula for the human model was: $c(T, F, D, C, E, I) = 16.6 - 1.3T + 0.98F - 0.064F^2 - 1.22D + 0.32C - 0.17E - 0.48I$.

2.5 Graphical User Interface

We implemented a panel based on the original climate control panel in the VOLT car, with additional add-ons. We used three different methods of advice, each with a different Graphical User Interface (GUI):

1. The first GUI is identical to the original climate control panel in the VOLT car. This option was used for the control group and is associated with drivers who do not receive any advice.
2. The second GUI has an additional information circle, which supplies the driver with an estimate of the current energy consumption level. This information appears as the percent of the current energy consumption from the maximum energy consumption obtained in the training data (the lower the better). This GUI is referred to as CAREless. Note that CAREless does not provide any active advice either. An example can be seen in Figure 2.1 (where the current consumption is 40% of the maximum.).
3. The third GUI is equipped with the full functionality of CARE. The driver is presented with both the advice provided by CARE and an estimate of the current energy consumption (similar to the information provided by CAREless). Figure 2.2 shows a screen-shot of a case in which the driver set the climate control differently from CARE's advice. The current user's selection is shown in green and the advice appears in purple.

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE



Figure 2.1: A screen-shot with additional energy consumption information provided by CAREless (the circle in the bottom left corner).



Figure 2.2: A screen-shot of the GUI with CARE's advice. In this example, the driver set the temperature to 18°C (rather than 21°C as advised by CARE), the fan to 4 (rather than 1 - as indicated by the purple line), the air delivery to face and feet (rather than face-only) and the mode to "comfort" (rather than "eco"). This resulted in an energy consumption level of 63% of the maximal energy consumption level (right green circle), rather than only 25% if the driver would have followed CARE's advice (left purple circle).

2.6 Experimental Evaluation

In order to evaluate the performance of CARE and CAREless we recruited 49 subjects for the evaluation phase, of which 33 were males and 16 were females¹. The subjects' ages ranged from 21 to 73, with a mean of 35 and median of 31. All subjects live in Israel. The subjects were paid 100 NIS each (27\$, which is equivalent to the price of fancy lunch in Israel). Each of the subjects was randomly assigned an advice provider, which was either CARE or CAREless; 24 subjects were assigned to receive advice from CARE, while 25 subjects were assigned to receive advice from CAREless. It is well known that people vary in their preferred temperature, fan, etc. settings and the outside temperature changes from time to time. Had we randomly assigned some subjects to a control group, those subjects may have had an average consumption that may have differed from the average consumption of the subjects receiving advice merely because of these differences. Therefore, in order to control this variance, we chose an experimental design that examined the effect of advice as a within-subject variable rather than a between-subject variable. We had each subject run the experiment twice, once with no advice (the control group) and once with advice given either from CARE or from CAREless. We counterbalanced the order among the type of experiments, i.e. approximately half of the subjects first ran the experiment with no advice, while the other half first ran the experiment with advice. Within each of these groups, approximately half of the subjects received advice from CARE while half received advice from CAREless.

Every subject adhered to the following procedure. First the subject was asked to fill out forms and demographic data, was then led to the car and was asked by the operator for his initial comfort level. Then, on a dedicated laptop (not on the car display) the subject was shown the GUI which, according to the experiment type, either presented advice from CARE, from CAREless or no advice. The subject then told the operator how to set the climate control system. The operator set the climate control system and updated the GUI accordingly, and showed it to the subject. The subject could then ask to modify the climate control system again. The subject remained in the car for a total of 10 minutes. The subject could ask the operator to modify the climate control system in these 10 minutes as many times as he wanted. The subjects then had to wait outside the car for 10 minutes between the experiments. The car doors and trunk were left open and the climate control system was turned off for those 10 minutes, in order to allow the temperature in the car to equalize to the outside temperature.

¹All experiments with human subjects were approved by the corresponding IRB.

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE

After the 10 minutes the subject came back to the car and ran the second experiment (which also lasted 10 minutes). The subject was then asked some final questions.

2.7 Results

The results were analyzed using repeated measures of ANOVA with total energy consumption as a dependent variable, advice (yes/no) as a within-subject variable, type of advice (CARE/CAREless), gender of the subject and order of presentation (baseline, first or second) as between-subject variables. Thus, the statistical model had one within-subject factor and three between-subject factors. The statistical analysis revealed no significant findings, except a trend suggesting that the effect of the advice depended on the type (either CARE or CAREless). We therefore ran separate analyses for each of the two advice types. When subjects were given advice by the CARE algorithm, their total energy consumption significantly decreased from 0.24 KWH to 0.20 KWH, showing an improvement of 17% ($F(1, 21) = 7.6, p < 0.05$)¹. This improvement amounted to a mean energy savings described in the 95% confidence interval: $[-24\%, -5\%]$. Neither the effect of the presentation order nor its interaction with the effect of advice was found to be significant. A similar analysis for the CAREless advice did not show any improvement in total energy consumption ($F(1, 23) = 0.12$). Figure 2.3 presents the mean energy consumption level of the climate control system, which was obtained by the subjects when receiving advice from CARE vs. CAREless, compared to the mean energy consumption level of the same subjects when they did not receive any advice.

Figure 2.4 shows the energy consumption level of the climate control system of each subject when receiving advice from CARE compared to the baseline of that same subject when not receiving any advice. As illustrated in the figure, for 19 of the 24 subjects an improvement was shown over their baseline when receiving advice from CARE (their associated points appear under the 45° diagonal). The figure also indicates that for three subjects, CARE reduced energy consumption by approximately 50% (from approximately 0.25 KWH to approximately 0.12 KWH).

¹We made corrections for multiple comparisons, and after the Bonferroni correction, the type-I error remained < 0.05 .

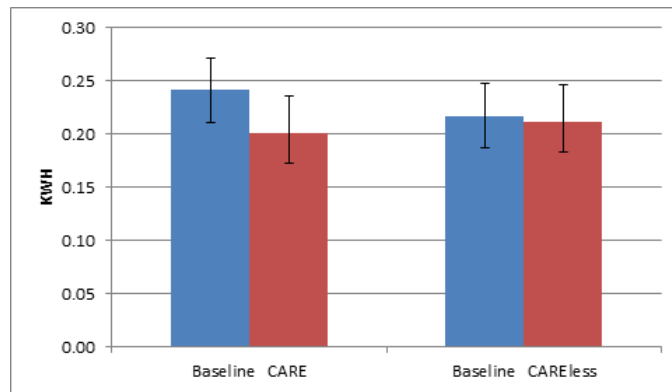


Figure 2.3: The mean energy consumption level of the subjects who received advice from CARE and CAREless, compared to the mean energy consumption levels of the subjects when they did not receive any advice.

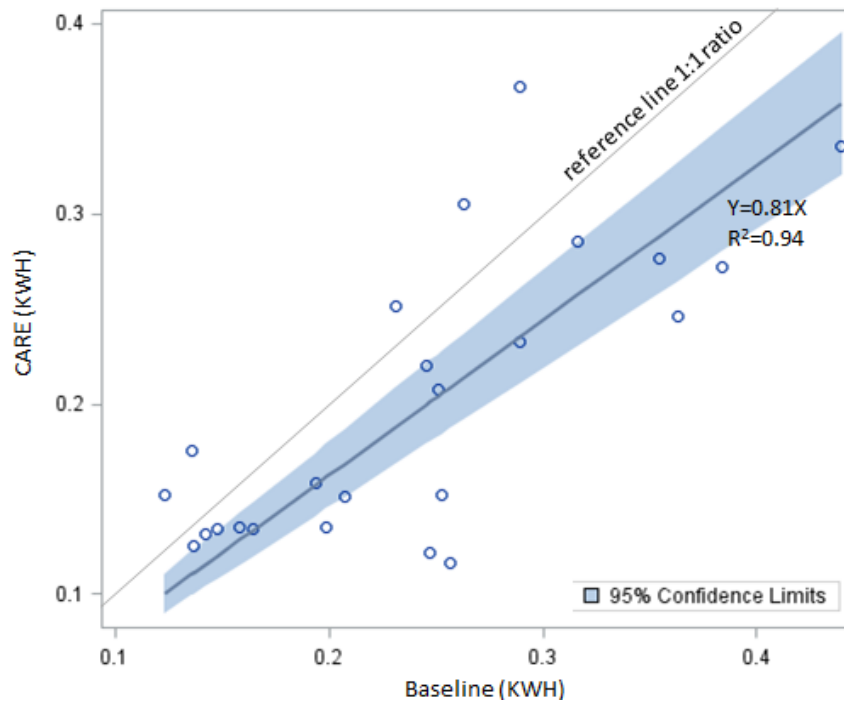


Figure 2.4: The energy consumption level of the climate control system of each subject when receiving advice from CARE compared to the baseline of the same subject when not receiving any advice.

2.8 Discussion

CARE significantly outperformed the control group. Perhaps this occurred not only because some of the subjects actually accepted the advice and used it, but seemingly even those who did not accept the advice were influenced by it. Some subjects also used the advice as a baseline and edited it. For example, one of the subjects, when receiving no advice, set the climate control system to a temperature of $23^{\circ}C$ and the fan to 4. However, when the same subject received the advice to set the temperature to $24^{\circ}C$ and the fan to 1, she set the temperature to $24^{\circ}C$ as suggested, but the fan to 2. Later, when she became a little too warm, she set the fan to 3, and the temperature at $24^{\circ}C$. Clearly, CARE caused her to reduce her energy consumption. In total, of the 24 subjects who received CARE's advice, only 4 followed the exact advice. Nonetheless, CARE caused a decrease in the energy consumption of subjects who did not follow its exact advice.

In order to ensure that the advice provided to the user was easy to understand, we asked the subjects the following question: "Was the information on the screen clear?" and we asked them to specify a number between 1 and 10. The average answer was 9.15, indicating that the GUI was very understandable. Another interesting observation is that females tend to consume less energy than males, 0.201 vs 0.242 (when looking only at the no-advice condition). This raises the possibility that demographic data may be used in addition to the information provided explicitly by the driver when entering the car.

2.9 Conclusions

In this chapter, we presented a method to persuade a driver to reduce the energy consumption of the climate control system of his electrical car. We showed via experiments that the proposed methodology leads to a significant reduction of energy consumption. The methodology requires the collection of data on the energy consumption of the climate control system and on the drivers' behavior. Nevertheless it is effective even with a small number of examples (15 drivers in our experiment). We designed a GUI to present the advice that facilitates understanding the advice. The reported work is the first step in the process of the deployment of a persuasive agent in a car.

2.10 List of Notations

2.10 List of Notations

notation	meaning
c	comfort level.
C	initial comfort level.
D	air delivery (“face” or “face and feet”).
e	energy consumption level.
F	fan strength (between 1 and 6).
E	external temperature.
I	internal temperature.
M	mode (“eco” or “comfort”).
s	climate control system setting.
T	temperature (between 16 and 35).
v	parameters learned by the comfort level model.
w	parameters learned by the energy consumption model.
s_D	air delivery in a specific setting.
s_F	fan strength in a specific setting.
s_M	mode in a specific setting.
s_T	temperature in a specific setting.
ψ_e	energy training data.
ψ_c	comfort training data.

Table 2.1: List of Notations

2. MULTI-DIMENSIONAL. INFLUENTIAL ADVICE

Recommending a set of actions.

3.1 Introduction

A prime example massively used to provide advice to users is recommender systems. Thus, in this chapter we consider a recommender system that recommends a set of actions rather than a single action to the user. Recommender systems usually take place in environments where users face a very large number of possible actions (for example, renting a movie or buying a book). The recommender system, in most cases, does not recommend a single item, but rather recommends a set of items to a user, who may in turn decide to buy/rent any subset of these items (or possibly other items). The main goal in designing recommender systems is usually to predict the user's most preferable items and to supply her with the best list of recommendations. This trend is prevalent whether we consider a social network recommending friends [44], consumer goods [45] or movies [46].

In this chapter, we provide evidence that a business may gain significantly by providing users with recommendations that may not be best from the user's point of view but rather serve the business' needs. We provide an algorithm which uses a general recommender system as a black-box, but modifies the recommendations to increase the utility of the business. We perform extensive experiments with it in various cases. In particular, we consider two settings:

1. The *Hidden Agenda* setting: In this setting, the business has items that it wants to promote, in a way which is opaque to the user. For example, a movie supplier who provides movies on a monthly fee basis but has different costs for different movies, or a social

3. RECOMMENDING A SET OF ACTIONS.

network that wants to connect users who are less engaged to more engaged ones. Netflix, for instance, set a filter to avoid recommending new releases which incur high costs for them [47].

2. The *Revenue Maximizing* setting: In this case the goal of the recommender system is to maximize the expected revenue, e.g. by recommending expensive items. In this setting, there is an inherent conflict between the user and the business.

To study these settings, we conducted experiments on Amazon Mechanical Turk (AMT) in which subjects were asked to choose a set of favorite movies (from a given set), and then were given recommendations for another set of movies. For each recommendation, the subjects were asked if they would or would not watch the movie¹. Finally, in order to test possible reduction in satisfaction which may have been caused by tuning the recommendations to the business' needs, we asked each subject how good she feels about the recommendations she received.

This form of experimentation makes two assumptions. First, we simulate long term effects by asking users about their satisfaction from the list. Second, we assume that asking users if they are willing to pay for a movie is the same as actually taking their money and screening the movie. Both assumptions are common in the literature (for a comparison between hypothetical and real scenarios see [48]). We hope to integrate the algorithm in a real world system to circumvent the assumptions.

Manipulating the recommender system in order to increase revenue (or to satisfy some other hidden agenda) raises some ethical concerns. If users believe that a particular algorithm is being used (e.g. collaborative filtering), then they could be irritated if they find out that recommendations are being edited in some way. However, most businesses do not provide the specification of their recommender system (treating it as a “secret sauce”), which eliminates this concern. Furthermore, several companies (including Netflix, Walmart and Amazon) admitted human intervention in their recommender system [19], so it may well be that companies are already tweaking their recommender systems for their own good. In this sense, an important lesson to take away from this work is “users beware!”. We show that businesses garner a large gain by manipulating the system, and many companies could be tempted by this increase in revenue. In this chapter we propose a method which allows businesses to harness their existing recommender system in order to increase their revenue.

¹In the Revenue Maximization setting each recommended movie also came with a price tag. We describe the experiments later in the experiments section.

To summarize, this chapter considers a recommender system that recommends a set of movies (actions) to a user. We provide algorithms for utility maximization of the movie supplier service, in two different settings, one with prices and the other without. These algorithms are provided along with an extensive experiment demonstrating their performance.

3.2 PUMA

In this section we present the Profit and Utility Maximizer Algorithm (PUMA). PUMA uses a black-boxed recommender system which supplies a ranked list of movies. This recommender system is assumed to be personalized to the users, even though this is not a requirement for PUMA.

3.2.1 Algorithm for the Hidden Agenda Setting

In the hidden agenda setting, the movie system supplier wants to promote certain movies. Movies are not assigned a price. We assume that each movie is assigned a promotion value, $v(m)$, which is in $V = \{0.1, 0.2, \dots, 1\}$. The promotion value is hidden from the user. The movie supplier wants to maximize the sum of movie promotions which are watched by the users, i.e. if a user watches a movie, m , the movie supplier gains $v(m)$; otherwise it gains nothing. The movie supplier can recommend n movies to each user.

The first phase in PUMA's construction is to collect data on the impact of the movie rank, $r(m)$, in the original recommender system on the likelihood, $p(m)$, of the users to watch the movie. To this end we provide recommendations, using the original recommender system. However we provide only a fraction of the recommendations and not all of them. We provide all movies which are ranked $\{1, k+1, 2k+1, 3k+1, \dots, (n-1) \cdot k+1\}$ for the given subject in the original recommender system. We then cluster the data according to the movie rank and, using least squared regression, we find a function that best explains the acceptance rate as a function of the movie rank. We consider the following possible functions: linear, exponent, log and power (see Table 3.1 for function forms). We do not consider functions which allow maximum points (global or local) that are not at the edges, since we assume that the acceptance rate of the users should be the highest for the top rank and then gradually decrease. Since these functions intend to predict the probability of the acceptance rate, they must return a value between 0 and 1. Consequently a negative value returned must be set to 0 (this may only happen with the linear and log functions; though, in practice, we did not encounter this need).

3. RECOMMENDING A SET OF ACTIONS.

Table 3.1: Function forms for the considered functions. α and β are non-negative parameters and $r(m)$ is the movie rank.

function	function form
linear (decay)	$\alpha - \beta \cdot r(m)$
exponent (exponential decay)	$\alpha \cdot e^{-\beta \cdot r(m)}$
log (logarithmic decay)	$\alpha - \beta \cdot \ln(r(m))$
power (decay)	$\alpha \cdot r(m)^{-\beta}$

Among the functions that we tested, the linear function turned out to provide the best fit to the data in the hidden agenda setting (it resulted in the greatest coefficient of determination (R^2)). Therefore, the probability that a user will want to watch a movie as a function of its rank (for the specific user) takes the form of:

$$p(m|r(m)) = \alpha - \beta \cdot r(m) \quad (3.1)$$

where α and β are constants.

Given a new user, PUMA calculates for each movie its expected promotion value, based on the output of the original recommender system. This value is given by:

$$p(m|r(m)) \cdot v(m) \quad (3.2)$$

Then, to maximize its expected promotion value, PUMA sorts all movies according to their expected promotion value. PUMA selects the top most n movies as its recommendation to the user.

3.2.2 Algorithm for Revenue Maximizing

In this setting, every movie is assigned a fixed price (different movies may have different prices). Each movie is also assumed to incur a cost for the vendor. PUMA seeks to maximize the revenue obtained by the vendor, which is the sum of all movies purchased by the users minus the sum of all costs incurred by the vendor.

PUMA's variant for the Revenue Maximizing settings entails a much more complex problem than PUMA's variant for the hidden agenda since it is unknown to PUMA how the movie

price may influence the likelihood of the users to buy it. Therefore, PUMA must consider both the movie rank *and* the movie price when building a user model.

Building a model by learning a function of both the movie rank and the movie price together requires too many data points. Furthermore, in such a learning phase the movie supplier intentionally provides sub-optimal recommendations, which may result in a great loss. Instead, we assume that the two variables are independent, i.e. if the movie rank drops, the likelihood of the user to buy the movie similarly decreases for any price group.

In order to learn the impact of the price on the likelihood of the users to pay to watch a movie, we use the recommender system as is, which provides recommendations from 1 to n . We cluster the data into pricing sets where each price (fee f) is associated with the fraction of users who are willing to pay that price to watch a movie (m). Using least squares regression we find a function that best explains the data as a function of the price. We tested the same functions described above (see Table 3.1 - replacing the movie rank with the movie fee), and the log function resulted in a nearly perfect fit to the data. Therefore, the probability that a user will be willing to pay in order to watch a movie as a function of its price takes the form of:

$$p(m|f(m)) = \alpha_1 - \beta_1 \cdot \ln(f(m)) \quad (3.3)$$

where α and β are constants.

In order to learn the impact of the movie rank (r) in the recommender system on the likelihood of the users to pay to watch a movie, we removed all prices from the movies and asked the subjects if they were willing to pay to watch the movie (without mentioning its price). As in the hidden agenda settings, we provided recommendations in leaps of k' (i.e. recommendations were in the group $\{1, k' + 1, \dots, (n - 1) \cdot k' + 1\}$). We clustered the data according to the movie rank and once again using least squared regression we found a function that best explains the data as a function of the movie rank. Among the functions that we tested (see Table 3.1), the log function turned out to provide the best fit to the data for the movie rank as well (resulting in the greatest coefficient of determination (R^2)). Using the log function (which is a convex function) implies that the drop in the user acceptance rate between movies in the top rankings is larger than the drop in the user acceptance rate within the bottom rankings. The difference in the function which best fits the data between the hidden agenda setting and the revenue maximizing setting is reasonable, since, when people must pay for movies they are more keen to pay for movies that better suit their exact taste. Consequently the acceptance rate drops more

3. RECOMMENDING A SET OF ACTIONS.

drastically. Thus, the probability that a user will be willing to pay in order to watch a movie as a function of its rank takes the form of:

$$p(m|r(m)) = \alpha_2 - \beta_2 \cdot \ln(r(m)) \quad (3.4)$$

A human model for predicting the human willingness to pay to watch a movie, $p(m|r(m), f(m))$, requires combining Equations 3.3 and 3.4; however this task is non-trivial. Taking $p(m|r(m)f(m))$ as $p(m|r(m)) \cdot p(m|f(m))$ does not make sense: for example if both signals say that the probability of watching is 0.5 then the output should be 0.5 and not 0.25. Using this intuition, we assume that Equation 3.3 is exact for the average rank for which it was trained, which is $\frac{n}{2} + 1$. Therefore, by adding a correction term, $\gamma(m)$, to Equation 3.4, we require that Equation 3.4 provides the same viewing probability as Equation 3.3 on $\frac{n}{2} + 1$:

$$\alpha_2 + \gamma(f(m)) - \beta_2 \cdot \ln(\frac{n}{2} + 1) = \alpha_1 - \beta_1 \cdot \ln(f(m)) \quad (3.5)$$

Isolating $\gamma(m)$ we get:

$$\gamma(f(m)) = (\alpha_1 - \alpha_2) + \beta_2 \cdot \ln(\frac{n}{2} + 1) - \beta_1 \cdot \ln(f(m)) \quad (3.6)$$

Therefore, our human model for predicting the fraction of users who will pay to watch a movie, m , given the movie price, $f(m)$, and the movie rank, $r(m)$ (obtained from the recommender system) is:

$$p(m|r(m), f(m)) = \alpha_2 + ((\alpha_1 - \alpha_2) + \beta_2 \cdot \ln(\frac{n}{2} + 1) - \beta_1 \cdot \ln(f(m))) - \beta_2 \cdot \ln(r(m)) \quad (3.7)$$

and after simple mathematical manipulations:

$$p(m|r(m), f(m)) = \alpha_1 - \beta_2 \cdot \ln(\frac{r(m)}{\frac{n}{2} + 1}) - \beta_1 \cdot \ln(f(m))$$

Once a human model is obtained, PUMA calculates the expected revenue from each movie simply by multiplying the movie revenue with the probability that the user will be willing to pay to watch it (obtained from the model) and returns the movies with the highest expected revenues. The revenue is simply the movie price ($f(m)$) minus the movie cost incurred by the vendor ($c(m)$). I.e. given a human model, PUMA recommends the top n movies which maximize:

$$(f(m) - c(m)) \cdot p(m|r(m), f(m)) \quad (3.8)$$



Figure 3.1: A screen-shot of a subject selecting movies he liked

3.3 Experiments

All of our experiments were performed using Amazon’s Mechanical Turk service (AMT). A total of 245 subjects from the USA participated in all experiments of which 50.6% were females and 49.4% were males, with an average age of 31.5. The subjects were paid 25 cents for participating in the study and a bonus of an additional 25 cents after completing it. We ensured that every subject would participate only once (even when considering different experiments). The movie corpus included 16,327 movies. The original movie recommender system received a list of 12 preferred movies for each user and returned a ranked list of movies that had a semantically similar description to the input movies, had a similar genre and also considered the year the movies were released and the popularity of the movies (a personalized non-collaborative filtering-based recommender system). We set $n = 10$, i.e., 10 movies were recommended to each subject.

After collecting demographic data, the subjects were asked to choose 12 movies which they enjoyed most from a list of 120 popular movies (see Figure 3.1 for a screen-shot). Then, depending on the experiment, the subjects were divided into different groups and received different recommendations.

3. RECOMMENDING A SET OF ACTIONS.

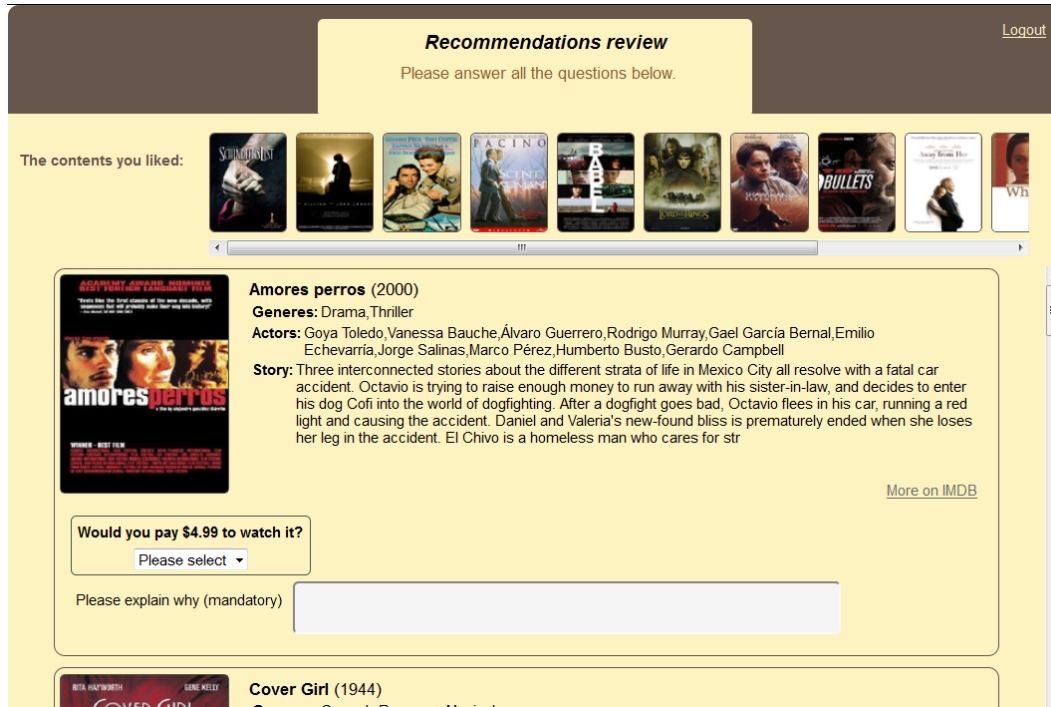


Figure 3.2: Recommendation page screen-shot

The list of recommendations included a description of each of the movies (see Figure 3.2 for an example). The subjects were shown the price of each movie, when relevant, and then according to their group were asked if they would like to pay in order to watch it, or simply if they would like to watch the movie. In order to ensure meaningful responses, the subjects were also required to explain their choice ("Please explain why (mandatory)"). After receiving the list of recommendations and specifying for each movie if they would like to pay to watch it, the subjects were shown another page including the exact same movies. This time they were asked whether they had seen each of the movies ("Did you ever watch *movie name*?"), whether they thought that a given movie was a good recommendation ("Is this a good recommendation?") and rated the full list ("How would you rate the full list of recommendations?") on a scale from 1 to 5. These questions were intentionally asked on a different page in order to avoid framing [49] and to ensure that the users provided their true preferences¹.

¹We conducted additional experiments where the subjects were first asked whether they watched each movie and then according to their answer, were asked whether they would pay to watch it (again). We obtained similar results, however, we do not present them since they may have been contaminated by the framing effect.

Table 3.2: Demographic statistics for the hidden agenda setting

Group	Number of subjects	Fraction of females	Average age
Rec-HA	31	58.1%	32.3
PUMA-HA	30	50.0%	32.6%
Learn-HA	30	53.3%	29.4%

3.3.1 Hidden Agenda Setting

In the hidden agenda setting we assume that the subjects are subscribers and therefore they were simply asked if they would like to watch each movie ("Would you watch it?"). The hidden agenda setting experiment was composed of three different groups. Subjects in the *Rec-HA* group were recommended the top 10 movies returned by the original recommender system. Subjects in the *PUMA-HA* group were recommended the movies chosen by PUMA. Subjects in the *Learn-HA* group were used for data collection in order to learn PUMA's human model. Table 3.2 presents some demographic statistics on the subjects in the three groups.

For the data collection in the movie rank phase (Learn-HA) we had to select a value for k (which determines the movie ranks for which we collected data; see Section 3.2.1). The lower the k is, the more accurate the human model is, resulting in better (lower) rankings. On the other hand, the higher the k the more rankings the human model may cover. In the extreme case where the ranking has a minor effect on the human acceptance rate, the vendor may want to recommend only movies with a promotion value of 1. Even in this extreme case, the highest movie rank, on average, should not exceed $|V| \cdot n$, which is 100. Therefore, we set $k = 10$, which allowed us to collect data on movies ranked: $\{1, 11, 21, 31, 41, 51, 61, 71, 81, 91\}$.

Unfortunately, the data collected in the Learn-HA group was very noisy, as the movies in the 11th rank resulted in a much higher acceptance rate than those in the 1st rank. Furthermore, the movies in the 71st rank resulted in a much lower acceptance rate than those in the 81st rank. Therefore the coefficient of determination (R^2) was only 0.31. Still, the tendency of the data was clear (the correlation between the movie rank and the acceptance rate was negative, 0.56, which implies that the original recommender system performed well), and additional data would have probably yielded a better coefficient of determination. Nevertheless, the fit-to-data

3. RECOMMENDING A SET OF ACTIONS.

Table 3.3: Coefficient of determination for functions tested for the hidden agenda setting

function	R^2
linear	0.31
exponent	0.29
log	0.21
power	0.21

that was reached was definitely adequate, as depicted in the performance results. Table 3.3 lists the coefficient of determination for all functions tested.

The specific human model obtained, which was used by PUMA (in the hidden agenda settings) is simply:

$$p(m|r(m)) = 0.6965 - 0.0017 \cdot r(m) \quad (3.9)$$

PUMA significantly ($p < 0.001$ using the student t-test) outperformed the original recommender system by increasing its promotion value by 57% with an average of 0.684 per movie for PUMA-HA versus an average of only 0.436 per movie for the Rec-HA group. No statistically significant differences were observed between the two groups concerning the average satisfaction for each of the movies or the user's satisfaction from the full list. However, it is likely that the use of more data would result in a statistically significant drop in the user's satisfaction. Our best estimate is a 3% drop in the fraction of good recommendations (from 71% rated as good recommendations in the Rec-HA group vs. 69% in the PUMA-HA group), and a 7% loss in the satisfaction from the entire list. See Table 3.4 for additional details.

3.3.2 Revenue Maximizing Settings

For the revenue maximizing settings, all movies were randomly assigned a price which was in $F = \{\$0.99, \$2.99, \$4.99, \$6.99, \$8.99\}$ ¹. We assumed that the vendor's cost does not depend on the number of movies sold and therefore we set $c(m) = 0$ for all movies. The subjects were asked if they would pay the price to watch the movie ("would you pay \$*movie price* to watch

¹We used random pricing since we did not find any correlation between Amazon's movie price and features such as popularity of the movie in IMDB, year of release, parental rating and country of production.

Table 3.4: The percentage of subjects who wanted to watch each movie, average promotion gain, overall satisfaction and the percentage of movies marked as good recommendations

Group	Want to watch	Average promotion	Overall satisfaction
Rec-HA	76.8%	0.436	4.13
PUMA-HA	69.8%	0.684	3.83
Learn-HA	62.0%	-	3.77

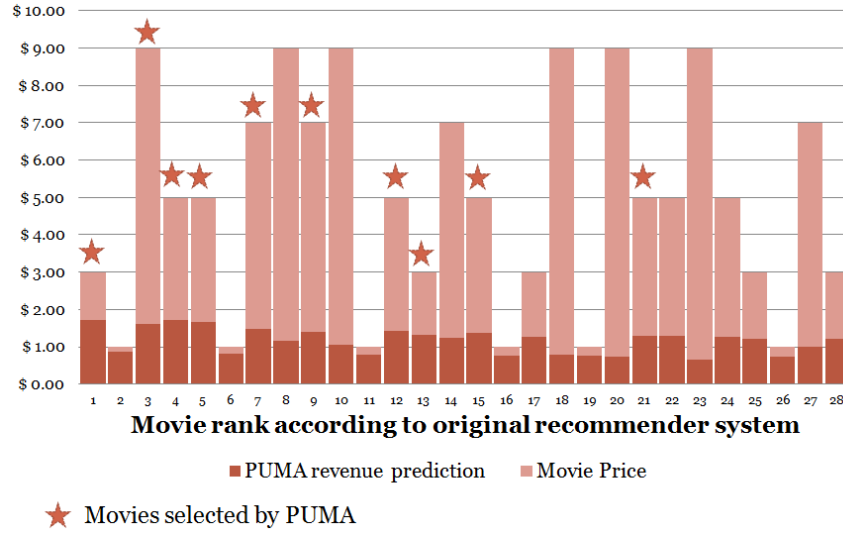


Figure 3.3: An example of PUMA’s selection process

it?”). As in the hidden agenda setting, subjects were divided into three groups. Subjects in the *Rec-RM* group received the top 10 movies returned by the original recommender system. Subjects in the *PUMA-RM* group received the movies chosen by PUMA. Subjects in the *Learn-RM* group were used in order to obtain data about the decay of interest in movies as a function of the movie rank (as explained in Section 3.2.2). The subjects in this group were asked if they were willing to pay for a movie, but were not told its price (“Would you pay to watch it?”). Table 3.5 presents some demographic statistics on the subjects in the three groups.

In the movie rank learning phase, we set $k' = 5$, i.e., recommendations were in the group $\{1, 6, 11, 16, 21, 26, 31, 36, 41, 46\}$. Once again, this is due to the fact that $k' \cdot n = |F| \cdot n$ (even

3. RECOMMENDING A SET OF ACTIONS.

Table 3.5: Demographic statistics for the revenue maximizing setting

Group	Number of subjects	Percentage of females	Average age
Rec-RM	31	41.9%	32.1
PUMA-RM	28	67.8%	29.7
Learn-RM	30	40.0%	33.9

Table 3.6: Coefficient of determination for the functions tested for the revenue maximizing settings.

function	R^2
linear	0.43
exponent	0.39
log	0.60
power	0.54

if the movie ranking has a minor impact on the probability that the user will watch the movie, and therefore PUMA would maintain a certain price; nonetheless, on average, it is not likely that PUMA would provide movies which will exceed rank $|F| \cdot n$, and therefore no data is needed for those high rankings). The coefficient of determination value using the log function on the learning data was 0.60, as presented in Table 3.6.

The specific human model obtained, which was used by PUMA (in the revenue maximizing settings), is:

$$p(m|r(m), f(m)) = 0.82 - 0.05 \cdot \ln\left(\frac{r(m)}{6}\right) - 0.31 \cdot \ln(f(m))$$

As illustrated in Figure 3.4, PUMA significantly ($p < 0.05$ using student t-test) outperformed Rec-RM, yielding an average revenue of \$1.71, opposed to only \$1.33 obtained by Rec-RM. No significant difference was revealed when testing the overall satisfaction level from the list: 4.13 vs. 4.04 in favor of the Rec-RM group. However, more data would probably result in a statistically significant difference. Our best estimate for this loss would be about 2.2%.

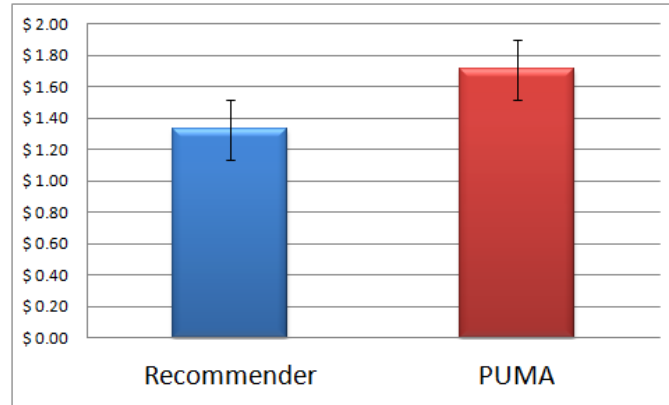


Figure 3.4: Average revenue per system (in dollars)

Table 3.7: The percentage of subjects who would pay for a movie, the average revenue and the overall satisfaction.

Group	want to buy	average revenue	overall satisfaction
Rec-RM	39.1%	\$1.33	4.13
PUMA-RM	37.1%	\$1.71	4.04
Learn-RM	56%	–	4.03

While the average movie price was also similar in both groups, with an average movie price of \$5.18 for Rec-RM and an average movie price of \$5.27 for PUMA-RM, the standard deviation was quite different: 2.84 for the Rec-RM group, and only 1.95 for PUMA-RM, in which 64.6% of the movies were either priced at \$2.99 or \$4.99.

Figure 3.3 demonstrates the selection process performed by PUMA for a specific user. After calculating the expected revenue using the human model, PUMA selected movies #1, #3, #4, #5, #7, #9, #12, #13, #15 and #21, which yielded the highest expected profit. In this example, when comparing PUMA's recommendation's expected revenue to the expected revenue from the first 10 movies (which would have been selected by the original recommender system), the expected revenue increased from \$1.34 to \$1.52 (13%).

3. RECOMMENDING A SET OF ACTIONS.

3.4 Discussion

There might be concern about PUMA's performance in the long run, when it is required to interact with the same person many times. Although this has not been explicitly tested, we believe that PUMA's advantage over the recommendation system will not degrade; this is due to the fact that the overall satisfaction from PUMA's recommendations and the average movie fee for PUMA's recommendations (in the revenue maximization setting) are both very close to that of the original recommender system. An interesting property of PUMA is that it allows online learning, as it may collect additional statistics on-the-fly and use it to refine its human model. In the revenue maximization setting, there is a clear conflict between the business and the user: recommending movies the advertiser prefers (expensive ones) is bound to reduce the probability that the suggestions will be accepted. In the hidden agenda setting, all movies are a-priori the same for the user, and hence the only loss in showing a recommendation that the business would like to promote is that it is lower on the user's list. As a result, there is an even greater gain in changing the list of recommendations, and a larger gap can be seen between PUMA and the recommendation engine in the hidden agenda setting.

3.5 List of Notations

notation	meaning
$f(m)$	price (fee) of a movie for the user.
$r(m)$	movie rank.
k	natural number parameter determining which movies to present.
m	a movie.
n	number of movies presented to each user.
$p(m)$	prediction on the likelihood of a user to watch movie m .
$v(m)$	promotion value.
V	promotion value range.
α	non-negative parameter.
β	non-negative parameter.

Table 3.8: List of Notations

Long-Term Influential Advice.

4.1 Introduction

In this chapter we consider a setting in which a system (sender) provides long-lasting advice to a human (receiver). We consider a path selection problem, in which a navigation application system provides a path to a user (a human driver). The human views the system’s advice and may decide occasionally to follow it or divert from it and possibly return to the advice later. This differs from the work presented in previous chapters in which we considered advice which may only have a one-time impact on the user.

To enable formal discussion of the path selection problem, we employ a grid model. The human’s task in this setting is to find a path from an origin to a destination on a large colored grid (see Figure 4.1 for an example). We assume that the human’s sole objective is to choose the shortest path. The human receives advice from an agent whose objectives also include the number of color changes in the path. Choosing a path on the grid corresponds, for example, to selecting a route for commuting between home and work. The colors on the grid represent constraints, such as environmental and social considerations. Switching between colors on the path represents the violation of one of these constraints. The person’s preferences consider the length of the route only, while the agent’s preferences take into account both the length of the route as well as the number of constraint violations.

We developed the User Modeling for Path Advice (UMPA) approach to generate advice, which comprises a training stage and three additional steps required to learn from the data and to generate the agent’s advice. We first ran experiments with human subjects to collect data on how users react when provided with advice. The system proposes three types of advice in

4. LONG-TERM INFLUENTIAL ADVICE.

different testing scenarios: advice that is optimal to the user, advice that is optimal to the system and advice that considers both the user's and the system's preferences. We found three types of user behaviors: those that follow the system's advice, no matter how bad it is subjectively perceived to be; those that ignore the advice and follow their chosen path; and those that modify the advised path. The latter phenomenon is very interesting since it illustrates that simply the fact that advice is provided may affect the user's choices. The users' modifications may completely change the advice or their own choice, but this change occurs only as a result of having seen such a system proposal. In particular, we noticed that users of the third type, in some cases along the path, took shortcuts, sometimes they took the long way around and sometimes along the path they modified the advised path, but this modification was exactly the same length as the original path. We term all these modifications to the original path as *cuts*. That is, cuts are deviations from a suggested path and are alternative segments for connecting two local points from the original path. A cut may improve the path from the user's point of view by shortening it, but may decrease the benefit to the agent.

Once we collected this data, the UMPA approach proceeded to 1) learn the percentage of types of users who will follow, ignore or modify the given advice, 2) learn the probability each cut will be chosen for a given advised path and 3) compute the advice with the lowest expected cost for the agent given the users' predicted types and behaviors.

We evaluated the UMPA approach in an extensive empirical study comprising close to 700 human subjects solving the path selection problem in four different mazes. The results showed that our UMPA agent outperformed alternative approaches for suggesting paths, based on either the user's or the system's preferences. In addition, the people were satisfied with the advice provided by the UMPA agent.

4.2 The Model

We employ the following maze model. We assumed that a user has to solve the shortest path problem within a rectangular maze either by constructing a path or by considering a path suggestion. More formally, we define a *maze* M as a grid of size $n \times m$ with one vertex marked as the source S and another vertex as the target T (see Section 4.6 for a notation list for this chapter). Each vertex v is associated with a label $c(v)$ that we will refer to as the *color* of v . We use the color white to denote an *obstacle*. $x(v)$ and $y(v)$ denote the horizontal and the vertical grid coordinates of the vertex v , respectively. We assume that the user can move along the grid

4. LONG-TERM INFLUENTIAL ADVICE.

at the maze during the limited amount of time given to the user. Therefore, the user may find it beneficial to take some advice provided to him regarding which route to choose.

The *best-advised path problem* is modeled as the agent’s task to compute a full path that, once presented to a user, will yield the agent the lowest expected cost.

Figure 4.1 visualizes the formal setting in a small maze. In the figure, obstacles are represented by the color white, while the start and the target nodes are black. In turn, the dotted nodes represent the advised path, while the crossed nodes represent a valid (partial) path selected by the user.

4.3 The UMPA Approach

We assume the availability of training data for the prediction stages (see experiments in Section 4.4). UMPA is given a training set, Ψ , of tuples (M', π, μ, α) collected from experiments where people were provided with advice and where: M' is a maze; π is an advised path through the maze; α is a binary variable indicating whether the user considers π to be a good solution or not (α equals 1 or 0, respectively); and μ is the solution selected by a human user, who was presented with M' and π . In addition, we assume that Ψ includes examples (M', μ) collected from games where the agent was silent. Given a maze M (not in the maze set from the training examples), we employ a three-stage process to solve the best-advised path problem: (i) We cluster users into one of three types, depending on the extent to which their path selection behavior adheres to suggested paths that may be more beneficial to the agent than to themselves. Then we predict the likelihood that a user will belong to one of these three clusters; (ii) We predict the likelihood that people will deviate from a suggested path; and (iii) We generate the advised path using a decision theoretic approach which utilizes the prediction from the first two stages in order to compute the expected cost of the agent from a given path. In the next subsections we provide details of our implementation of each one of these steps.

Predicting human response to an advised path is difficult due to the diversity in people’s behavior. We propose to integrate psychological models into the machine learning process. In particular, we defined a *Seemliness-value* attribute that measures the path’s direction towards the target node’s horizontal and vertical coordinates. This attribute tries to measure how good the path may seem in the eyes of a human user. This attribute is based on the following principles known from behavioral science:

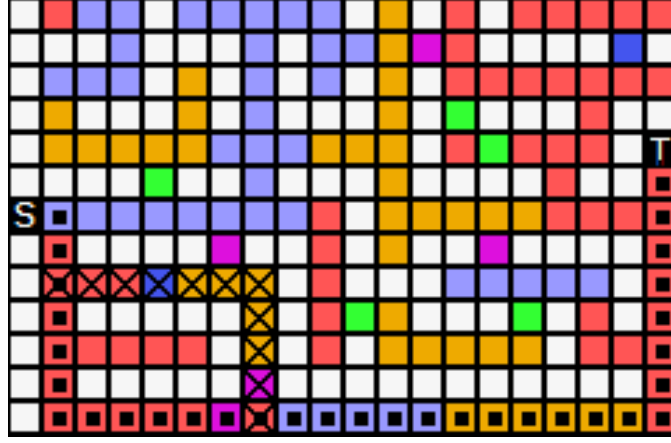


Figure 4.2: A second example of a path and a cut

- Loss aversion (Prospect theory): people dislike losing more than they like winning. Tversky and Kahneman [50] found that losses are weighted roughly twice as much as gains. Therefore, while each step in the path toward the target contributes a single unit to the Seemliness-value, each step away from the target reduces two units from the value.
- Future discount [51]: people care more about the present than the future and therefore discount losses or gains in the future. The farther the loss or the gain is in the future, the more it is discounted. Future discounting is commonly assumed to be exponential, with some discount factor [52]. Therefore, while each step in the path toward the target at the beginning of the path adds one unit (and a step away from the target in the beginning of the path reduces two units), the contribution of any consecutive steps' is multiplied by a discount factor (which is exponential in the number of steps from the beginning of the path).

The total path Seemliness-value is calculated as a discounted sum of steps contribution along the path and is denoted $s(\phi)$. For an intuitive example, the dotted path shown in Figure 4.1 has a relatively high Seemliness-value since its earlier steps are in the target direction and steps in the opposite direction appear only later; however, in Figure 4.2 the dotted path has a relatively low Seemliness-value since the steps at the beginning of the path are in the opposite direction of the target.

4. LONG-TERM INFLUENTIAL ADVICE.

4.3.1 Modeling Diversity in People's Reactions

Based on what was observed in the behavioral data collection experiments (as explained in Section 4.4), UMPA clusters users into three types: *Advice followers*, *Advice ignorers* and *Advice modifiers*. Given a new maze, when considering a path to be given as advice, UMPA would like to estimate the probability of a user belonging to one of these clusters.

For this task, it first labels the examples of Ψ with one of the three types. The labels are determined as follows. *Advice followers* are users who follow the advised path blindly without modifying it, even when believing that it is not of good quality. That is, the user of an example $(M', \pi, \mu, \alpha) \in \Psi$ is labeled as an *Advice follower* if $\mu = \pi$ and $\alpha = 0$. Users that took the system's advice as provided and also believed that the advised path really did have good quality were included in the *Advice modifiers* type set (these users may have chosen the advice because it was of good quality and not because they were told to choose it).

However, most users would at least attempt to improve the advised path, or simply ignore it entirely. In order to characterize these users, we will introduce the concept of a *cut* and a *modified solution*.

Given two vertexes π^i and $\pi^{i'}$, of an advised path π , any path τ between these two vertexes (that does not otherwise intersect with π) is termed a *cut*. Although there may be an exponential number of cuts, certain human cognitive tendencies (see e.g. [24, 53]) allow us to bound the maximal cut length. All users who deviated from the advised path solely by taking cuts are termed *Advice modifiers*.

More formally, given a valid path π , we define a cut τ of length l to be a valid path such that $\exists i, \tau^1 = \pi^i$ and $\exists i' > i, \tau^l = \pi^{i'}$ and $\forall 1 < i'' < l, \nexists j, \pi^{i''} = \pi^j$. We will refer to the sequence of π^i, \dots, π^l as the original segment of cut τ and denote it by $o(\tau)$. Figure 4.1 and Figure 4.2 show examples of cuts marked by crossed nodes. We only consider cuts whose lengths are smaller than some threshold and also not much longer than their original segment. Formally, $l(\tau) \leq \min\{L_1, L_2 \cdot l(o(\tau))\}$, for some $L_1 \in \mathbb{N}$ and $L_2 \in \mathbb{R}^+$.

Finally, we define the *Advice ignorers* as all users who are neither *Advice modifiers* nor *Advice followers*. The relevant examples of Ψ were labeled accordingly. It is important to understand that being an advice follower does not depend on the specific maze and advice. However, deciding whether to ignore advice or use it as a baseline and modify it, depends on the specific maze and advice.

Next we compute the likelihood of users being associated with the different types as required in the first step of the UMPA approach. Based on the literature on route selection (see e.g. [54]), we presume that the proportion of *Advice modifiers* for the given advice π is strongly characterized by the overall Seemliness-value of π , denoted $s(\pi)$. In order to use the Seemliness-value of a path as an indicator for the proportion of Advice modifiers in that path, we first normalize the Seemliness-value by subtracting the average of all Seemliness-values of all paths that appear in the data-set and divide by their standard deviation. Once we have a standardized (scaleless) value, we assume that it predicts a standardized proportion of Advice modifiers in that path, therefore, this value must be unstandardized using the appropriate units found in the data-set. Formally, given Ψ , UMPA generates a set of tuples $\pi', s(\pi'), prop(\pi')$ where $prop(\pi')$ is the proportion of users in Ψ that received the advice π' and are labeled as *Advice modifiers*. Denote the average (standard deviation) of the $s(\pi')$ s by $AvgSV$ ($StdSV$) and the average (standard deviation) of $prop(\pi')$ s by $AvgBU$ ($StdBU$). Finally, we estimate the proportion of *Advice modifiers* to be: $p_b(\pi) = \frac{s(\pi) - AvgSV}{StdSV} \cdot StdBU + AvgBU$.

The *Advice followers* follow the advised path even if they did not evaluate it as a good path. This allows us to assume that the proportion of *Advice followers* is constant across all advised paths. We extract this proportion from Ψ , and denote it by p_f . The remaining proportion of users, $1 - p_f - p_b(\pi)$, is assumed to be the *Advice ignorers*. This latter set of users deviates from the advised path so much that we may assume that they would have selected the same path even if there were no advice present.

4.3.2 Predicting Advice Deviations

Given the possible advice π , UMPA estimates the probability of a user to take a specific cut τ at a given vertex π^i . We denote this probability as $p(M, \pi, \pi^i, \tau)$ and use $p(\tau)$ when the other parameters are clear from the context. UMPA assumes that the function $p(\tau)$ is a linear combination of three cut features: *cut benefit*, *cut orientation* and *cut seemliness* (see e.g. [54]).

The **Cut Benefit** measures the relative reduction in steps between the cut and the original path segment. Formally, $\frac{l(o(\tau)) - l(\tau)}{l(\tau)}$. For example, the cut shown in Figure 4.1 (marked with crossed nodes) has a positive benefit value since the length of the original path segment (between the first and last nodes of the cut) is greater than the length of the cut. The cut shown in Figure 4.2 has a benefit of 0 since the cut has the same length as the original path segment.

4. LONG-TERM INFLUENTIAL ADVICE.

The **Cut Orientation** captures the tendency of human users to continue with a straight line motion. Its value depends on whether the cut or the original segment conformed to this tendency. The reference motion is the edge between the cut divergence node π^i and its predecessor in the advised path π^{i-1} . If the cut deviates from the advice by remaining in the same direction as the edge (π^{i-1}, π^i) , we say that the cut has a positive, $+1$, orientation. If the original path segment (π^i, π^{i+1}) is similarly directed as (π^{i-1}, π^i) , we say that the cut has a negative, -1 , orientation. Otherwise, the cut's orientation is 0 (neutral). For example, in Figure 4.1 the value of the orientation of the cut marked by crossed nodes is 1, since the cut continues straight while the advised path turns left. The cut shown in Figure 4.2, however, has an orientation of -1 since the original path continues straight and the cut turns left.

The **Cut Seemliness** measures how seemly the cut is in the user's eyes. This value is calculated by subtracting the Seemliness-value of the original segment from the Seemliness-value of the cut. The seemliness of the cut shown in Figure 4.2 is positive since the first steps of the cut are in the same direction of the target, while the first steps in the original segment are in the opposite direction of the target.

Given that there is a very large number of cuts, it is almost impossible to collect enough examples in Ψ to learn the weights of $p(\tau)$'s features directly. Therefore, this estimation process was divided into two steps. First, UMPA estimates the probability, $r(M, \pi, \pi^i, \tau)$, that a cut τ will be taken by a user at vertex π^i , assuming that τ is the only possible cut at π^i . It was assumed that r is a linear combination of the three cut features described above, similar to $p(\tau)$. To compute the weights of $r(\tau)$'s features, UMPA created a training set of the form $(M', \pi, \pi^i, \tau, prop(\pi^i))$, where τ is a cut of π that starts at π^i and is the cut that was taken at π^i by the highest number of users according to Ψ . $prop(\pi^i)$ is the proportion of users that visited π^i and deviated by taking any cut. Using these examples, the weights were estimated using linear regression.

Next, $r(\tau)$ is used to compute $p(\tau)$ after normalization. For any π^i , it was assumed (based on the way that $r(\tau)$ was learned) that the probability of the deviation at π^i across all cuts is equal to the highest $r(\tau)$ value of a cut starting at π^i . This probability is then distributed across all possible cuts, starting at π^i , proportional to their $r(\tau)$ value.

4.3.3 Estimating the Cost of an Advised Path

Given - a maze M and the possible advice π , UMPA estimates the expected cost that an agent may incur when presenting users with π . We denote this estimation by $ECost(\pi)$. This estimation is based on Ψ (the set of examples labeled with user types).

Notice that the contribution of the *Advice followers* is relatively easy to calculate. These are users that, independent of the maze or the particulates of the advised path π , always comply fully with π . Therefore, their contribution to $ECost(\pi)$ will always be $Cost_a(\pi)$ multiplied by the ratio of *Advice followers*.

The contribution of the *Advice ignorers* is calculated based on the data of users who received no advice. Let $\Omega_\emptyset = \{\tau | (M, \phi) \in \Psi\}$, i.e. the set of paths in Ψ selected by users who did not receive any advice. We assume that the contribution of *Advice ignorers* to $ECost$ is the average agent cost on the paths in Ω_\emptyset . We use $ECost_i$ to denote this value.

Calculating the contribution of the *Advice modifiers* to the agent's expected cost is more complex and is described hereunder. Having the estimated probability for each cut $p(\tau)$, an estimation of the agent's cost associated with *Advice modifiers* from advice π starting at π^i is denoted as $b(\pi, \pi^i)$. It can be calculated using the following recursive formulas:

$$b(\pi, \pi^{l(\pi)}) = 1 \quad (4.1)$$

$$b(\pi, \pi^i) = \sum_{\tau, \tau^1 = \pi^i} p(\tau) \cdot ((Cost_a(\tau) - 1) + b(\pi, \tau^{l(\tau)})) + \quad (4.2)$$

$$+ (1 - \sum_{\tau, \tau^1 = \pi^i} p(\tau)) \cdot (b(\pi, \pi^{i+1}) + Cost_a(\pi^i \pi^{i+1}) - 1) \quad (4.3)$$

Note that the expression $Cost_a(\pi^i \pi^{i+1}) - 1$ is the agent's cost of traveling from π^i to π^{i+1} , which can either be 1 if no color switching occurs, or $W + 1$ if color switching occurs. Now, using b , UMPA can estimate the contribution of the *Advice modifiers* to the agent's expected cost of an entire path π setting $ECost_b(\pi) = b(\pi, S)$.

Simply calculating the expected cost from the using Equations 4.1 may not be efficient, since the recursive call is performed more than once. We therefore suggest the following algorithm which provides an efficient method for computing $ECost_b$:

Input: A maze, with an advised path π .

Output: $ECost_b(\pi)$ – estimated cost contributed by Advice modifiers

- 1: $ECost_b \leftarrow Cost_a(\pi)$.
- 2: $vec \in \mathbf{R}^{l(\pi)} \leftarrow \vec{0}$. $vec(0) = 1$.

4. LONG-TERM INFLUENTIAL ADVICE.

```

3: for each  $i < l(\pi)$  do
4:   for each cut  $\tau$  s.t.  $\tau^1 = \pi^i$  do
5:     {Predict the fraction of Advice modifiers who take the cut}
      $a(\tau) \leftarrow (1 + \sum_{j < i} vec[j]) \cdot p(\tau)$ 
6:      $ECost_b \leftarrow ECost_b + (Cost_a(\tau) - Cost_a(o(\tau))) \cdot a(\tau)$ .
7:     {Update mass at cut entry point.}
      $vec[i] \leftarrow vec[i] - a(\tau)$ 
8:     {Update the cut exit point}
      $vec[j|\pi^j = \tau^{l(\tau)}] \leftarrow vec[j] + a(\tau)$ 
9: return  $ECost_b$ .

```

Intuitively, the algorithm’s basic assumption is that the set of users forms a continuous unit mass. The algorithm then traces the flow of this unit of mass along different cuts that diverge (or converge) at vertexes along the advised path.

In more detail, the algorithm begins by stating that, even if all users are *Advice followers*, their contribution will be at least $Cost_a(\pi)$ and initializes the utility estimate of this value. Then the vector vec , which lists for each vertex along the path π the proportion of people who have reached it, is initialized by placing proportion 1 (all people) at the start node and zero (no people) at all other nodes along the path. The algorithm then systematically propagates the mass of people along the path and the path’s cuts. At every node along the path, the mass of people split – some continue along the path to the next node along π , while others take one of the available cuts. Specifically, they split proportionately to the probability of users to adopt a particular path segment. Those who choose a cut τ are advanced and added to the mass of people who reach the end point of that cut.

This algorithm can be implemented with a complexity of $O(\#cuts + l(\pi))$.

Given the users’ proportions as estimated in Section 4.3.1 and the utility contributions estimated above, we can compose the final *heuristic* estimate of the advised path cost $ECost(\pi)$, which is the expected agent’s cost across all human generated path solutions in response to π :

$$\begin{aligned}
 ECost(\pi) = & p_f \cdot Cost_a(\pi) + (1 - p_f - p_b(\pi)) \cdot ECost_i + \\
 & p_b(\pi) \cdot ECost_b(\pi)
 \end{aligned}$$

4.3.4 Searching for Good Advice

Searching for advice is done by transforming the maze(grid) into a tree such that the start node, S , is associated with the root of the tree. Each node in the tree is associated with a vertex in the

maze. A node n_v in the tree that is associated with the vertex v will have an offspring which is associated with v' if no ancestor of n_v is associated with v' and v' is connected to v in the grid. Note that a vertex in the grid might be associated with many nodes in the tree. When given a node n_v in the tree that is associated with the vertex v , there is a unique path in the tree from the root node of the tree to n_v that is associated with a path on the grid from S to v . We denote such a path as $\theta(S, v)$.

A^* [55], which is a best-first search algorithm in graphs, uses the sum of a cost function and a heuristic function in order to determine which node to view next. We use the A^* search algorithm on the tree, to find a path π from the root node S to any target T . The cost function for a given node n_v is $ECost(\theta(S, v))$ and the agent uses the minimal agent cost of traveling between v and T as the heuristic function of n_v in the tree. We use Dijkstra’s algorithm, which is an efficient algorithm for calculating the shortest path from a given node to all other nodes in a graph, starting at T , in order to calculate the minimal agent cost to travel from each vertex to T .

The search only considers paths with cuts where the agent does not gain by the user taking them. That is, the agent only provides advice for which it will benefit from the user following the advice. UMPA does so since providing advice which the system will gain from the user deviating from the advice may be perceived as deception. Formally, UMPA only considers paths such that, for any suffix $\sigma = \pi^i \dots \pi^{l(\pi)}, i \geq 1$, $ECost(\sigma) \geq Cost_a(\sigma)$ holds. If A^* stops at a path that does not satisfy the condition above, it will be rejected, and A^* will be forced to continue the search.

4.4 Experimental Evaluation

We developed an online system that allows people to solve path selection problems in a maze. It can be accessed via <http://azariaa.com/selfmazeplayer.swf>. The maze design was chosen to remove all effects of familiarity with the navigation network from the experiments. Furthermore, every human subject was presented with a single instance of the problem in order to exclude effects of learning or trust. We ran two kinds of experiments. First, the experiments were aimed at collecting data on users’ behaviors when facing advice that either benefited the users or the system utilities regarding route selection. Second, after the UMPA approach was applied using the collected data, we ran experiments to validate our hypothesis regarding users’ behavior change as a result of providing them with advice adapted to the user’s

4. LONG-TERM INFLUENTIAL ADVICE.

behavior which was learned in the first experiments. Furthermore, the main goal was to test the hypothesis that UMPA would outperform all of the other advice generator methods that we considered.

Participation in our study consisted of 681 subjects from the USA: 383 females and 298 males. The subjects' ages ranged from 18 to 72, with a mean of 37.

4.4.1 Methodology

Experimental Setup

Each experiment consisted of a colored-maze panel similar to the one depicted in Figure 4.1. A single panel was shown to each participant. The user's task was to select the shortest path through the maze that connected the source and target nodes. When subjects were presented with advice from the system, they *were informed* that this advice was calculated to reduce the number of color switches in addition to minimizing the path length. We implicitly asked the subjects a question regarding the system's intention to make sure that they understood this crucial point. We used four distinct mazes, all of size 80×40 . These mazes were complex enough so that users would find it difficult to compute the shortest path in the limited time allotted for the task. We set the weight W for color switching to 15.

We ran data collection for four training sessions to learn users' behaviors in three mazes. Then we ran our UMPA algorithm on the fourth maze to compute the advice, using information about this maze and the parameters learned from the other three mazes (we did this for each one of the four mazes). That is, UMPA's results were averaged over four different mazes and training and testing data were strictly separated.

Finally, we presented the subjects with post-task questions that were designed to assess the general attitude towards computer advice and the subjective evaluation of the advised path quality.

Basic Algorithms

We compared the performance of our UMPA algorithm to the following three cases:

- *No advice (silent)* – no advice is presented on the maze panel,
- *Shortest path* – the advice presented corresponds to the shortest path from source to target,

- *Greedy* – the advice that the user gets is the path computed to minimize the agent’s cost of traversing it, $Cost_a$.

The *Shortest* solution is the one that minimizes the cost of the user and, therefore, we expect that its acceptance by the users will be high. Moreover, the number of *advice ignorers* will be small and the probability of deviation will be low as well. However, since the agent’s cost for this path is usually high, we expect that presenting *Shortest* will yield the agent a relatively high average cost. When providing *Greedy* advice, we run the risk that most of the users will ignore it, while the ones that will accept it will yield the highest benefits for the agent. We first compared the agent’s average cost when providing any one of these three types of advice. (This comparison was performed using ANOVA). Then we chose the one that was best for the agent and compared the UMPA solution to this *baseline algorithm*. Then we considered UMPA estimation methods, its performance vs. the baseline algorithm and whether it decreased the user’s benefit and satisfaction or if it was mutually beneficial for both the agent and the user.

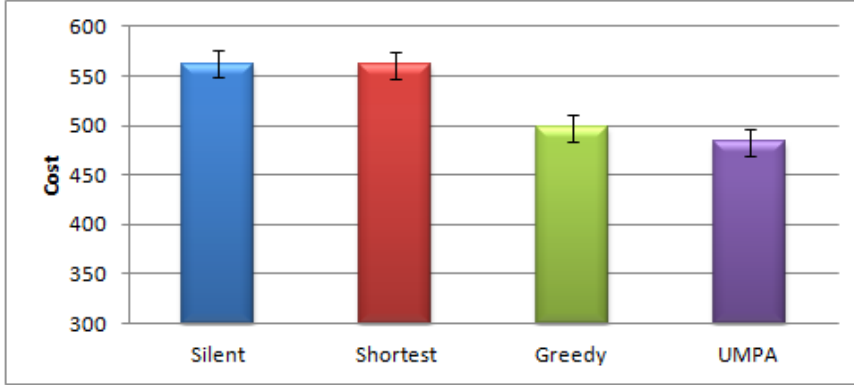
4.4.2 Basic Results

We calculated the effects that Silent, Shortest and Greedy types of advice have on the average agent cost across paths selected by users in our experiments. The corresponding three bar charts on the left of Figure 4.3 summarize the results (the lower the better). The average costs over four mazes of types Silent, Shortest and Greedy were 559.73, 559.55 and 501.68, respectively. That is, the paths chosen by users after receiving *Greedy* advice resulted in a significantly ($p < 0.001$) lower cost for the agent than the cost attained when the other two types of advice were given (Shortest and Silent).

We also studied the statistics of the effect of the advice on the user’s cost (see the three left-most bar charts in Figure 4.4). As expected, the cost of the paths chosen by users was significantly lower (130.85) when *Shortest* advice was provided, than when the other two types of advice were given (Greedy (144.6) and Silent (142.75)). Moreover, we wanted to check whether giving advice that results in the lowest costs for the agent can also decrease the costs for the users, when compared to the case where no advice is provided. The results were mixed and no significant difference was found between Greedy and Silent. That is, while *Greedy* advice significantly decreased the agent’s cost, it did not significantly increase the user’s costs. We concluded that the UMPA advice generation algorithm should be compared to the case where *Greedy* advice is provided.

4. LONG-TERM INFLUENTIAL ADVICE.

Figure 4.3: Average agent's costs



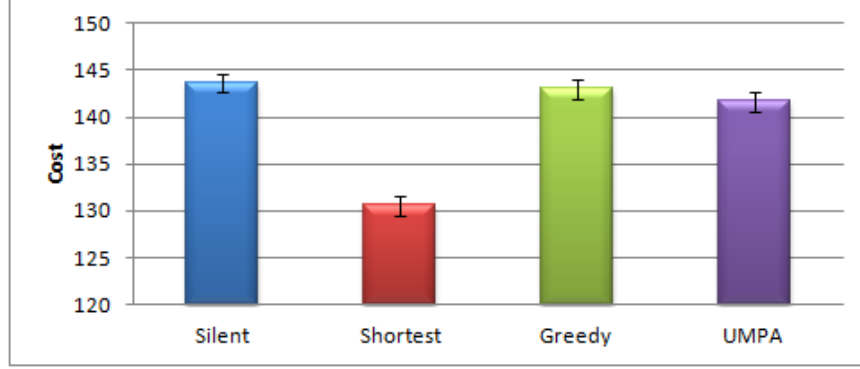
4.4.3 UMPA Advice Algorithm Performance

We set the UMPA parameters as follows: the length of a cut L_1 was bound to 40; a cut's potential increase in length L_2 to 20% of the corresponding original segment and the discount factor δ in the cut-seemliness feature calculation was set to 0.95.

The first step in the evaluation of our UMPA algorithm was to verify the effectiveness in computing $p(M, \pi, \pi^i, \tau)$ (i.e., the *predicted* number of users that will take cut τ when facing divergence node i , when advice π was provided in maze M). We found a high correlation (0.77) between this prediction and the actual fraction of users who took it when reaching the cut's divergence node. A high correlation (0.7) was also found between the actual fraction of users that took advice π or manipulated it, the *Advice modifiers* and our *predicted* number of such users, $p_b(\pi)$. Finally, we obtained a high correlation (0.76) between the estimated value of advice π , $ECost_a(\pi)$ and the empirical average value of the actual paths selected in response to advice π . This is significant since the correlation between the agent's cost of π itself and the empirical average of the selected path was only 0.06.

We then compared the average cost attained by the agents when users chose paths after receiving either the UMPA-based advice or *Greedy* advice. Consider the two corresponding bar charts on the right side of Figure 4.3 (the lower the better). UMPA's average of costs over the four mazes was 484.95 compared with the *Greedy* advice that was 501.68. That is, on average, the UMPA approach outperformed *Greedy* advice, resulting in significantly lower costs ($p < 0.05$) for the agent.

We also compared the average cost incurred by the paths chosen by users to the users themselves when receiving the advice provided by the UMPA algorithm and *Greedy* advice (see

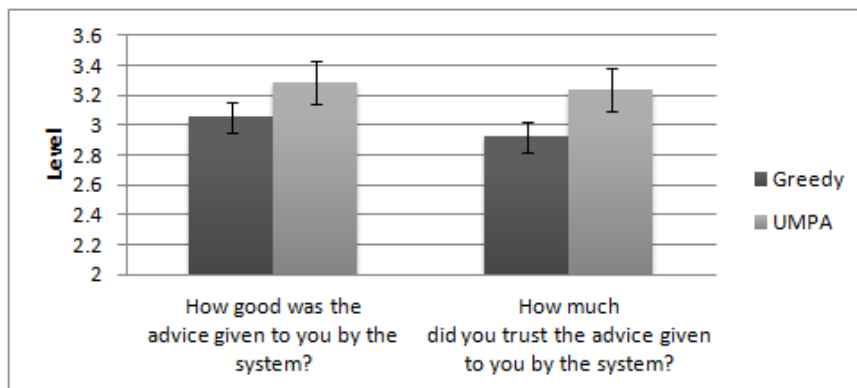
Figure 4.4: Average users' costs

the two right-most bar charts in Figure 4.4). The average results attained by the users that were given UMPA advice (142.33) were significantly better (lower cost) than those attained by users who were presented *Greedy* advice (142.33) ($p < 0.05$). In summary, when comparing the results obtained by running two advice generation techniques (one that provides UMPA advice and the other provides *Greedy* advice), we can conclude that UMPA-based advice outperforms *Greedy* advice. That is, both the average cost incurred by the agent and the average cost incurred by the human users decreased significantly when the users were provided with UMPA advice. So UMPA manipulative advice is indeed *mutually* beneficial when compared with *Greedy* advice.

Finally, we considered the subjective view of the users on the paths that were advised. Users were presented with the following questions after they finished the route selection task: (i) "How good was the advice given to you by the system?" and (ii) "How much did you trust the advice given to you by the system?" The possible answers were on a scale of 1-5, where 5 indicated the highest satisfaction and 1 the lowest satisfaction. The results are presented in Figure 4.5. Regarding the first question, UMPA advice was considered to be significantly better than *Greedy* advice, with $p < 0.05$. The average rating for UMPA was 3.29 and the average rating for *Greedy* was only 3.05. Similarly, with respect to trust, the average rating of UMPA was 3.23 whereas the average rating of *Greedy* was only 2.92, i.e., users trusted UMPA advice more than *Greedy* advice ($p < 0.05$).

4. LONG-TERM INFLUENTIAL ADVICE.

Figure 4.5: Users' satisfaction and trust



4.5 Conclusions

In this chapter we presented a computational model for advice generation in human-computer settings where the advice provided by the system, although being a one shot advice, may have long term impact on human behavior. To assess the potential effectiveness of our approach, we performed an extensive set of path selection experiments in mazes. Results showed that the agent was able to outperform alternative methods that either solely considered the agent's or the person's benefit, or did not provide any advice.

The approach that was described in this chapter can be technically summarized as follows: first, users' responses to basic advice patterns were sampled. Then a model of the response was created using machine learning and relevant psychological models. Finally, inverse kinematics of the model was solved in order to find the most profitable advice. This technical structure can be repeated in any domain or task where a self-interested agent can provide advice to a human user and the basic response data can be obtained. Specifically, whenever the task can be converted into a path-in-graph formulation (e.g. supply-chain plans), our solution can become an out-of-the box, yet tunable, method for providing advice.

Given these encouraging results, we deem that the proposed technology can be applied to other applications where the agent's goal is to provide people with advice that will lead them to take beneficial actions over a period of time. Recent applications, such as coaching humans in weight-loss programs, programs to help quit smoking or online service providers such as automated travel agents are promising domains.

4.6 List of Notations

Notation	Meaning
$c(v)$	vertex color (label).
$Cost_u(\pi)$	cost of a path π for the user.
$Cost_a(\pi)$	cost of a path π for the agent.
$ECost(\pi)$	<i>heuristic</i> estimate of the advised path, π , cost for the agent.
$ECost_a(\pi)$	<i>heuristic</i> estimate of the advised path, π , cost for the agent from <i>advice followers</i> .
$ECost_i(\pi)$	<i>heuristic</i> estimate of the advised path, π , cost for agent from <i>advice ignorers</i> .
$ECost_b(\pi)$	<i>heuristic</i> estimate of the advised path, π , cost for the agent from <i>advice modifiers</i> .
$l(\pi)$	length of path π .
L_1, L_2	scalar parameters.
m	maze height.
M	a maze.
n	maze width.
$p(M, \pi, \pi^i, \tau)$	an estimation of the probability of a user to take a specific cut τ at a given vertex π^i , given the possible advice π .
$p_b(\pi)$	proportion of <i>advice modifiers</i> given an advised path π .
p_f	proportion of <i>advice followers</i> .
$s(\pi)$	Seemliness-value of path π .
S	source / starting point.
T	target.
v	vertex.
$x(v)$	horizontal coordinate of v .
$y(v)$	vertical coordinate of v .
π	path.

4. LONG-TERM INFLUENTIAL ADVICE.

τ	cut.
Ψ	training set.

Table 4.1: List of Notations

Providing Advice in Repeated Interactions

5.1 Introduction

In this chapter we focus 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 with incomplete information called *choice selection processes* comprising a human and a computer player. As we assume in previous chapters, both of the participants in the 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 a person not knowing the traffic conditions on all of the roads. At each round, the computer suggests one of several choices to the person, and the person then makes 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 which accumulates 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. We designed several models of human behavior in choice selection processes that incorporate quantal response, exponential smoothing, and hyperbolic discounting theories taken from behavioral economics. Their predictive power was measured on a sampled set of hundreds of instances of human play. We estimated the parameters of these models using maximum likelihood techniques. The

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

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 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 maximize the agent's utility over time.

The agents' 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 interactions between the computer and the person. The first domain was analogous to a route selection task in which users needed to choose one of several possible commuting routes 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). 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 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. This domain was more challenging, as the computer needed to reason about the option that a person may only partially follow the computer's advice, by selecting comfort levels that are close to, but not equal to the computer's suggestion.

In both 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 different baseline strategies. The first of which provided 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 the agents to the *SAP* approach, which considered the costs for both the agent and the person when making suggestions. The agents were evaluated using hundreds of people that interacted with the system we developed on Amazon's Mechanical Turk [56]. The results show that *SAP* was consistently able to outperform all other agent strategies and produce advice in polynomial running time.

This work is the 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. It provides a model of selection processes for two domains and shows the efficacy of the model in empirical experiments.

5.2 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 suggest to the receiver to take one of the actions in a predefined set A . After observing the suggestion of the sender 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 $c_R(v, a)$ and $c_S(v, a)$, respectively. Both players, receiver and sender, can observe the outcome at the end of each round. However the sender has full information about the distribution over \mathcal{V} and the costs of both participants. 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 .

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 $c_R(v, a^t)$ and c_S^t denotes $c_S(v, a^t)$. 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 $h^{1,t}$). See Section 5.6 for a complete list of notations.

We are interested in developing an agent that can provide useful advice to users while both interact repeatedly. The novelty of this agent design is that the agent not only tries to maximize its utility function when interacting with its users but also considers a model of the user behavior in order to compute the advice strategically given the possible reaction of the user to any advice suggested. In the following sections, we present the algorithm developed for providing advice that outperformed a series of agent behavior candidates including the optimal solution found for an MDP-based agent. The winning strategy comprised a social utility based agent that chooses such advice that minimizes the weighted cost of the agent and the user.

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

The critical point in this strategy is to compute the weight of this social utility function that balances between the user's preferences and the probability of following the advice and the system's (agent's) preferences. In order to compute this weight a model of human behavior needed to be built. Using this model the agent simulates the human receiver and searches for a single weight that performs best and minimizes the agent's overall cost. This weight is then used in practice to provide advice. We proposed, formalized and implemented various models of human behavior taking into account behavioral economics models studied and also different characteristics of possible domains.

In particular, we refer to two types of domains. In the first one, the Multi-armed Bandit, users can choose one action from a set of discrete options. The advice is one of the options. The users do not have information about the state of the world and therefore each interaction can be considered a new interaction. Therefore, a user cannot learn over time. One example of such a domain, is a route-selection domain, where a driver needs to choose which route to navigate each day from his home to his workplace. Assuming the driver has no information about the traffic on each road, an agent can provide advice of one road per interaction. The driver can either accept the advice by choosing to navigate to the proposed road or choose another road. For the Multi-armed Bandit domain, we tested four different models of human behavior:

- SoftMax
- ES (based on Exponential Smoothing)
- Hyper (based on hyperbolic-discounting)
- Short memory.

SoftMax is a known method to solve multi-armed bandit problems [57]. It simulates a user that takes averages of each possible arm and reacts to the averages. Since the user in such a domain does not have information about the states of the game, he cannot actually learn from past experience. Therefore we tested a short memory model where we averaged the user's results from the interactions over the last seven interactions. The exponential smoothing method is a well-known method for modeling the way humans remember past incidents [58]. Hyperbolic discounting is a known method described in the behavioral economics literature ([59, 60]) which refers to the way humans anticipate future events (see a more detailed description below). All human models in both domains rely also on logit quantal response [61] and thus assume that although the user is more likely to choose actions which result in a lower modeled

cost, they might also choose actions which have a higher modeled cost (though with lower probabilities).

In the second domain, Partially Informed Users and Ordered Actions, users have some observation of the state prior to making a decision regarding their actions. In the former domain, a user can either follow the advice or not by choosing a different action. In this domain, a user can follow an advice or choose some other action that can be close to or far away from the advice. An agent can take advantage of this feature when computing the advice. An example of such a domain, is a climate control system deployed in a car. A driver observes the heat load currently in the cabin and needs to choose an action to set the power level of the climate control system. In the simulation system we implemented in the experiments (see Section 5.4), the user chooses a value that can be close to or far away from the advice given. The user would like to increase his comfort level and decrease his energy consumption level. In this domain not only must the agent model how humans remember previous actions but also how humans are influenced by the advice and how they interpret their observation. Due to the differences between the types of domains, we tested different models of human behavior:

- True-Cost (models the user using the true cost).
- LUQ (models the user using a linear combination of the user's expected comfort level and the energy consumption level).
- Hyper w/o learning (based on hyperbolic discounting).
- ES w/o learning (based on exponential smoothing).
- Hyper with learning (based on hyperbolic discounting and also models the receiver's learning).
- ES with learning (based on exponential smoothing and also models the receiver's learning).
- MAB (assumes that the receiver treats the problem as a multi armed bandit problem).

The MAB model disregards all the differences between the two domains and treats the problem as a Multi-Armed Bandit problem (using hyperbolic discounting - the method which performed best in the first domain). We will show that the MAB model does not perform well in this domain. Hyper w/o learning, ES w/o learning, Hyper with learning and ES with learning, all

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

differ by the way the user is modeled to remember past events and whether or not he learns what his expected comfort level is from past actions. The True-Cost and LUQ models simplify the user model and disregard the sender’s advice and are therefore less accurate.

As we show in the following sections, we empirically tested these models and found those that best fit the data collected. We also implemented two models for the advice-provider agent in both domains: the social utility based agent (SAP) and the MDP-Based agent (or Markov-Chain Monte-Carlo). We also implemented 3 additional baseline agents: Silent (which provides no advice), Receiver (which provides the best advice for the receiver) and Sender (which provides the best advice for the sender in the current round and ignores the receiver’s utility function).

5.3 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 $v = (v_1, \dots, v_{|A|})$ is 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 subjective costs when reasoning about which action to take.

We define the *subjective cost* the receiver incurs for taking action a at time t , denoted $SC^a(t, h^t)$ 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 subjective cost equals some default value K . 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 subjective cost of the receiver for using route 66 on day 1 equals 45 minutes, while the subjective 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 $P(a, t \mid h^{1,t-1}, d)$. 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:

$$EC_S^t(v, h^{1,t-1}, d) = \sum_{a \in A} P(a \mid h^{1,t-1}, d, t) \cdot (c_S(a, v) + \gamma \int_{v'} P(v') \operatorname{argmin}_{d'} EC_S^{t+1}(v, h^{1,t}, d') dv') \quad (5.1)$$

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:

$$\pi^*(v, h^{1,t-1}, t) = \operatorname{argmin}_d EC_S^t(v, h^{1,t-1}, d) \quad (5.2)$$

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

The next section describes different models for human behaviors studied to implement $P(a \mid h^{1,t-1}, d, t)$ when computing the advice.

5.3.1 Human Receivers as Multi-Armed Bandits

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. We present four candidate models for describing human receiver behavior that integrate theories from behavioral economics.

- SoftMax
- ES (based on Exponential Smoothing)
- Hyper (based on hyperbolic-discounting)
- Short memory.

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) [62], 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

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

the advice yielded a high performance for the user, he will be more likely to follow the advice in future actions.

Before presenting the model, we need to make the following extensions to our existing formalization. First, we generalize the definition of subjective cost of the receiver for following the advice of the sender. We define the subjective cost incurred by the receiver for taking advice d at time t , denoted $SC^F(t, h^t)$, 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 SC^a , in which a may be any action). $SC^{(\eta)}$ will denote a general subjective cost function in which (η) may either be an action a (implying that $SC^{(\eta)}$ will compute SC^a for any action a) or F (implying that $SC^{(\eta)}$ will compute SC^F).

Next, we generalize the notion of the receiver's subjective cost to include behavior over multiple rounds. Let $AC^a(t, h^{1,t-1})$ denote the aggregate subjective cost incurred by the receiver at rounds 1 through $t-1$ for taking action a , and $AC^F(t, h^{1,t-1})$ the aggregate subjective cost incurred by the receiver at rounds 1 through $t-1$ for following the advice.

We can now describe several models which differ in how they aggregate the receiver's subjective 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* [59, 60], 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 5.3 models the accumulative cost in the *hyper* model:

$$AC^{(\eta)}(t, h^{1,t-1}) = \sum_{t' < t} \frac{SC^{(\eta)}(t, h^{t'})}{\delta \cdot (t - t')} \quad (5.3)$$

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

In the *Exponential Smoothing* model [58], the discount factor δ is constant over time, meaning the perceived difference between the commute times will stay the same over time. The subjective 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

$$AC^{(\eta)}(t, h^{1,t-1}) = \delta \cdot SC^{(\eta)}(t, h^{t-1}) + (1 - \delta) \cdot AC^{(\eta)}(t-1, h^{1,t-2}) \quad (5.4)$$

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

$$AC^{(\eta)}(t, h^{1,t-1}) = AC^{(\eta)}(t-1, h^{1,t-2}) \quad (5.5)$$

If $t = 1$ then $AC^{(\eta)}(t, h^{1,t-1})$ equals a default value L 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 [63, 64]). The aggregated subjective cost for the receiver is defined as follows:

$$AC^{(\eta)}(t, h^{1,t-1}) = \sum_{t-7 \leq t' < t} SC^{(\eta)}(t', h^{1,t'-1}) \cdot \frac{1}{7} \quad (5.6)$$

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 [57] in which the aggregate subjective cost of the receiver for any action is simply the average true cost (as opposed to the subjective cost) of taking this action in past rounds, with no discount factor:

$$AC^{(\eta)}(t, h^{1,t-1}) = \frac{\sum_{1 \leq t' < t} C_R^{t'} \cdot \mathbf{1}\{(\eta) = a_{t'} \vee ((\eta) = F \wedge d_{t'} = a_{t'})\}}{\sum_{1 \leq t' < t} \mathbf{1}\{(\eta) = a_{t'} \vee ((\eta) = F \wedge d_{t'} = a_{t'})\}} \quad (5.7)$$

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.

The probability of the action a 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 subjective cost we adopted the quantal response theory from behavioral economics [61] for choice of actions. This theory assigns a probability of choosing an action a that is inversely proportional to the aggregate subjective 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 subjective

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

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.

$$P(a, t | h^{1,t-1}, d) = \frac{e^{-\lambda \cdot AC^a(t, h^{1,t-1})} + Z}{e^{-\lambda \cdot AC^F(t, h^{1,t-1})} + \sum_{a \in A} e^{-\lambda \cdot AC^a(t, h^{1,t-1})}} \quad (5.8)$$

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 5.3.3.

5.3.2 Agent Design for Senders

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.

Markov Decision Process

In this approach the sender's decision making process is represented as a continuous MDP. 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 $h^{1,t} \in H^{1,t}$ define a world state $s^t = (v^t, h^{1,t-1})$. The set of all such world states at time t is:

$$S^t = \{(v^t, h^{1,t-1}, t) \mid v^t \in \mathcal{V}, h^{1,t-1} \in H^{1,t-1}\} \quad (5.9)$$

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 5.2.)

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

$$P(s^{t+1} \mid s^t, d^t) = P(a^t \mid h^{1,t-1}, d^t, t) \cdot P(v^{t+1}) \quad (5.10)$$

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

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 5.2.*

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:

$$Q(s^t, d) = r(s^t) + \gamma \int_{s'} P(s' \mid s^t, d) \cdot V(s') ds' \quad (5.11)$$

$$V(s^t) = \max_d Q(s^t, d) \quad (5.12)$$

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

$$\pi^*(s^t) = \arg \max_d Q(s^t, d) \quad (5.13)$$

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 5.11 may be replaced by:

$$Q((v^t, h^{1,t-1}), d) = -c_S(a^{t-1}, v^{t-1}) + \gamma \cdot \sum_{a \in A} P(a \mid h^{1,t-1}, d, t) \cdot \int_{v^{t+1}} P(v^{t+1}) \cdot \arg \max_d Q((v^{t+1}, h^{1,t}), d) dv^{t+1} \quad (5.14)$$

According to Equation 5.1 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 5.13 satisfies Equation 5.2. \square

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 [65], partition of the state space [66], or use kernel-based methods [67], 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

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

their actions over time. The agent, called *MCS*, chooses the optimal advice for the current time step while using Monte-Carlo Simulation [68, 69] for selecting future states according to the transition function of Equation 5.10, and selecting future actions of the sender according to a uniform probability distribution.

Social Preference Approach

According to the social preference theory, people consider others' outcomes as well as their own when making strategic decisions [70]. 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. For a state v and a weight w , a policy for advice provision is a decision d with minimal social cost.

$$d = \pi(v, w) = \arg \min_{d \in A} (1-w) \cdot (c_R(d, v)) + w \cdot (c_S(d, v)) \quad (5.15)$$

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 .

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 5.15). See examples of such models for human behaviors in Section 5.3.1. 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. Note that in each iteration of the search process, w remains fixed for that iteration in the rightmost term of equation 5.16.

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 5.1 as it does not require to solve the intractable $\arg \min$ expression in Equation 5.1 to obtain the future advice but instead uses $\pi(v, w)$ as a fixed policy.

$$EW_S^t(v, h^{1,t-1}, w) = \sum_{a \in A} P(a \mid h^{1,t-1}, \pi(v, w), t) \cdot (c_S(a, v) + \gamma \int_{v'} p(v') EW_S^{t+1}(v', h^{1,t}, w) dv') \quad (5.16)$$

The weight w is chosen to minimize the sender's aggregate costs for the fixed policy $\pi(v, w)$

$$w^* = \arg \min_w EW_S^t(v, h^{1,t-1}, w) \quad (5.17)$$

5.3.3 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.

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

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.

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 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 5.1 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 5.3 and 5.8) exhibited a higher fit-to-data than all the other models of human receivers.¹

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,

¹For all the models, we set the default value K to equal the mean travel time of the road associated with the highest commuting time, representing an upper bound for the receiver's cost.

Table 5.1: 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	169.4
short memory	1	186.9

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 5.17). 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). *SAP* chose the w with the lowest overall average cost as w^* . Then, in each round, *SAP* provided advice according to Equation 5.15, using the optimal weight. The second agent used the MDP model to make offers, by solving Equation 5.11. We estimated $V(s^t)$ using Markov Chain Monte Carlo sampling [68, 69] in a manner similar to that of the MCTS method mentioned in Silver et al. [71].¹ 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 5.2. As an alternative to the hyperbolic discounting model, we also considered an approach using an ϵ -greedy strategy to describe possible behavior of a receiver. This strategy is commonly used to solve Multi Armed Bandit problems [57], which describes the choice selection problem from the point of view of the receiver. This strategy provides a rational baseline that seeks to minimize travel

¹This method is more common in POMDPs, however, since our state space is very large, we use this method as well.

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

Table 5.2: 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

time for receivers over time. Table 5.3 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 5.3, 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 ϵ -greedy model were significantly lower than the costs incurred by all other agents using the ϵ -greedy model.

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 ϵ -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 ϵ -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 5.1 presents the fuel consumption of each one of the treatment groups. As can be seen in the figure, the SAP-hyper agent significantly ($p < 0.05$ using the Mann-Whitney test)

Table 5.3: Simulation results comparing agent strategies

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

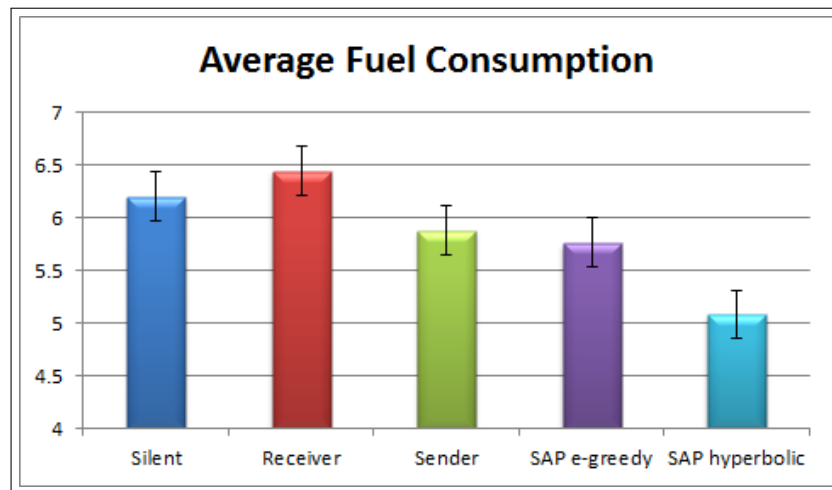


Figure 5.1: Average fuel consumption for each of the treatment groups (the lower the better).

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

Table 5.4: Performance results of agents interacting with people. The selfishness rate equals w in Equation 5.15

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%

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 5.4 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 “selfishness” column in the table measures the degree to which the agent was self-interested (the weight w in Equation 5.15).

5.3.4 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 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 ϵ -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.

5.4 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

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

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 AC 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.

5.4.1 Setting Description

In this setting, A is an ordered set of actions $(1, \dots, |A|)$. Each action represents the setting of the power level of the climate control system. The state of the world $v = (v_1, \dots, v_{|A|})$ 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.

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).

$$c_R(a, v) = \alpha \cdot v_a + \beta \cdot a \quad (5.18)$$

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:

$$P(a \mid h^{1,t-1}, d, t, v) \quad (5.19)$$

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

$$EC_S^t(v, h^{1,t-1}, d) = \sum_{a \in A} P(a \mid h^{1,t-1}, d, t, v) (c_s(a, v) + \gamma \sum_{v' \in V} p(v') (\min_{d'} EC_s^{t+1}(v', h^{1,t}, d'))) \quad (5.20)$$

The advice that minimizes the sender's cost is

$$\pi^*(v, h^{1,t-1}, d) = \operatorname{argmin}_d EC_S^t(v, h^{1,t-1}, d) \quad (5.21)$$

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. PROVIDING ADVICE IN REPEATED INTERACTIONS

5.4.2 Modeling Human Receivers

We provide 7 ways to model a human receiver:

- True-Cost (models the user using the true cost).
- LUQ (using a linear combination of the comfort level and energy consumption).
- Hyper w/o learning (based on hyperbolic discounting).
- ES w/o learning (based on exponential smoothing).
- Hyper with learning (based on hyperbolic discounting and also models the receiver's learning).
- ES with learning (based on exponential smoothing and also models the receiver's learning).
- MAB (assumes that the receiver treats the problem as a multi armed bandit problem).

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.

$$C(a, v, d, h^{t-1}) = c_R(a, v) \quad (5.22)$$

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

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot v_a + w_2 \cdot a \quad (5.23)$$

We assume that $w_1 \leq 0$ and $w_2 \geq 0$. A similar approach (based on building a subjective utility function using a linear combination of the parameters and using a quantal response) for modeling humans was performed successfully in previous work ([43], [40]). This candidate is termed LUQ (Linear combination for subjective 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)¹.

An alternative to the models shown above is to specifically represent the sender's advice in the cost function of the receiver. First, we provide the following definitions. We define the subjective cost of the receiver from following the advice as $SC^F(t | h^t)$ as in the road selection domain. Additionally, we define $SC^N(t, h^t)$, 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 (K). We define two types of aggregated costs (for human receivers), one which employs hyperbolic discounting:

$$AC^{(\eta)}(h^{1,t-1}) = \sum_{t' < t} \frac{SC^{(\eta)}(t | h^{t'})}{\delta \cdot (t - t')} \quad (5.24)$$

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

The second type of aggregated cost employs exponential smoothing:

$$AC^{(\eta)}(h^{1,t-1}) = \sum_{t' < t} SC^{(\eta)}(t | h^{t'}) \delta^{(t-t')} \quad (5.25)$$

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):

$$tr(h^{1,t-1}) = \frac{1}{1 + e^{-(AC^N(t, h^{1,t-1}) - AC^F(t, h^{1,t-1}))}} \quad (5.26)$$

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 $(AC^N(t, h^{1,t-1}) - AC^F(t, h^{1,t-1}))$ 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

¹This 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.

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

partial acceptance of advice is possible, we can consider the distance of an action from the advice:

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot v_a + w_2 \cdot a + w_4 \cdot tr(h^{1,t-1}) \cdot w_7^{-|d-a|} \quad (5.27)$$

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:

$$m_b(a, v, t) = \frac{1}{N_Z} \sum_{a' \in A} e^{w_8 \cdot |a' - a| + w_6 \cdot (t+1) \cdot \mathbf{1}\{a' \neq a\}} \cdot v_{a'} \quad (5.28)$$

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:

$$N_Z = \sum_{\bar{a} \in A} \sum_{a' \in A} e^{w_8 \cdot |a' - \bar{a}| + w_6 \cdot (t+1) \cdot \mathbf{1}\{a' \neq \bar{a}\}} \cdot v_{a'} \quad (5.29)$$

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 5.27 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):

$$C(a, v, d, t, h^{1,t-1}) = w_1 \cdot m_b(a, v, t) + w_2 \cdot a + w_4 \cdot tr(h^{1,t-1}) \cdot w_7^{-|d-a|} \quad (5.30)$$

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 subjective cost is:

$$P(a \mid h^{1,t-1}, d(v, h^{1,t-1}), v) = \frac{e^{-\lambda \cdot C(a, v, d, t, h^{1,t-1})}}{\sum_{a' \in A} e^{-\lambda \cdot C(a', v, d, t, h^{1,t-1})}} \quad (5.31)$$

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.

5.4.3 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 5.21. In the SAP agent the *Hyper with learning* human model (which resulted in the best fit-to-data see Section 5.4.6) 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 5.5) 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 5.13. In order to solve the MDP, the state space must be discretized.

5.4.4 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

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

system and the receiver represents the human driver. We set $A = \{1, 2, \dots, 10\}$ 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:

$$v_a^o = 10 \cdot \frac{1}{1 + e^{-(a-o)}} \quad (5.32)$$

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:

$$c_R(a, v) = -2 \cdot v_a + a \quad (5.33)$$

5.4.5 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 subjective 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 $c_s(d, v) > c_s(a', v)$ 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 5.4.3 (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.

5.4.6 Results

We begin by describing the fit-to-data of the various models we described in Section 5.4.2 using the data gathered in the Silent, Receiver and Sender groups.

Table 5.5 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 5.5 cannot be directly compared to those in Table 5.1 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

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

Table 5.5: 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

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 5.2 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 5.6 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.

5.4.7 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

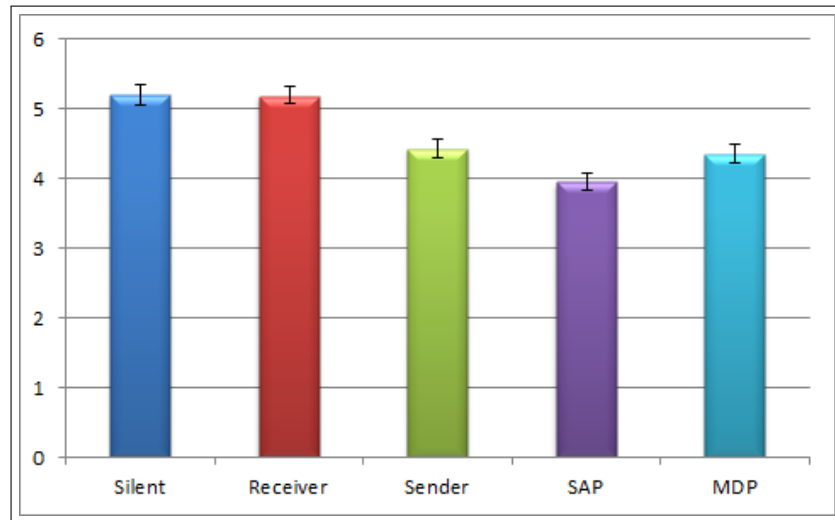


Figure 5.2: Average energy consumption level for each of the treatment groups (the lower the better).

Table 5.6: 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. PROVIDING ADVICE IN REPEATED INTERACTIONS

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, over-relying on a noisy model as the MDP does, may cause the SAP, which only uses the human model as a guideline, to perform better.

5.5 Conclusions

In this chapter 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 $O(|A|)$.

5.6 List of Notations

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) .
K	default value (used for computing subjective cost).
$m_b(a, v, t)$	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 .
$SC^a(t)$	subjective cost for receiver for taking action a at time t .
$SC^F(t)$	subjective cost for receiver for following the advice at time t .
$SC^N(t)$	subjective 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 or parameter.
$AC^a(t, h^{1,t-1})$	aggregated subjective cost for receiver for action a from round 1 to round $t - 1$ (route selection domain).
$AC^F(t, h^{1,t-1})$	aggregated subjective cost for receiver for following the advice from round 1 to round $t - 1$.

5. PROVIDING ADVICE IN REPEATED INTERACTIONS

$AC^N(t, h^{1,t-1})$	aggregated subjective 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 subjective 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 5.7: Notation list

Part II

Persuasion by Information Disclosure and Presentation

Which Information to Disclose?

6.1 Introduction

Computer systems have a major role in providing information to humans. This information may either be via the web (search engine, news, etc.), navigation systems or decision support systems. This information is not always ingenuous; at times, this information may be intended to influence a user into performing certain actions rather than others. In this chapter we focus on scenarios in which an automated agent interacting with humans possesses greater information than them. The automated agent needs to reveal information to humans, thereby leading them to perform actions that are preferable to the agent.

Game theory, in particular persuasion games, are the most popular disciplines that study strategic reasoning as required by the mixed intelligent systems on which we are concentrating. In such games (e.g. [11, 13, 72]) two rational entities interact: a Sender and a Receiver. The Sender provides information and is assumed to be more knowledgeable and the Receiver performs an action based on the information received. In contrast to previous chapters in which the Sender could only send advice, in this chapter the Sender sends actual information.

Some examples may be Google Maps [73] or Waze [74] applications that know possible settings that influence traffic congestion in the relevant countries and their times (e.g., morning rush hours) and have a distribution over the time it takes to drive on most of the roads. Similarly, automated travel agents have extensive prior information on flights and the distribution over their delays.

In this chapter, we extend these game-theoretical models as follows: While the agent holds private information (i.e., unknown to the user), it is also uncertain about the exact current

6. WHICH INFORMATION TO DISCLOSE?

state of the world. For example, the system may have an estimation of the congestion of traffic on different roads which may be unknown to the user. Still, the system may have only an estimation and not the exact value of traffic density at a particular time. We consider the setting of a one-shot interaction where the agent presents the user with information and the user chooses an action based on this information. The agent can present partial information about the state, however any information revealed by the agent must be true (unlike other works which consider manipulating the information presented to the user such as [75]). Similar to previous chapters, the utility functions of the agent and the user are different, but both depend on the state of the world and the action performed by the user. We model this setting as an optimization problem for the Sender and present an algorithm to solve it.

As in previous chapters, we intend to use our methodology for the agent. Namely, we will model human behavior in information disclosure environments and use it to find the optimal behavior for the agent. In order to model human behavior in this environment, we suggest the Linear weighted-Utility Quantal response (LUQ) human model, which relies on the following two assumptions: Linear Weighted-Utility, i.e. people's subjective utility is a linear combination of attributes, and Logit quantal response whereby the probability that people will chose a certain action is proportional to the action's subjective utility.

We ran extensive evaluations involving the participation of over 700 human subjects in two different domains. One domain considers a road selection problem (described in Section 6.4.1) and the second considers a supply-demand interaction detailed in Section 6.4.2. We discovered that, in the road selection problem domain, people deviated from rational behavior and therefore an agent based on the LUQ method significantly outperformed a game theory-based agent. However, in the supply-demand domain, people behaved nearly rationally and thus the LUQ based agent and the game theory-based agent's performance did not differ significantly.

To summarize, our key contributions in this chapter are:

- An extension of the persuasion game model for human-agent interaction with asymmetric information and two-sided uncertainty;
- A formal solution algorithm for the model, parameterized by the Receiver (human) behavior model;
- The LUQ method for building a human behavioral model pertinent to the Sender-Receiver type interaction;

- A methodology determining when one can assume rational behavior and thus use the game theory approach and when one should use the LUQ method.

6.2 The Information Disclosure Game with Two-Sided Uncertainty

In this section we formally describe the protocol of the interaction between the human user and the advising agent. To this end we use the terminology and general format of (Bayesian) persuasion games [72] (hence, naming the human user a Receiver, and the agent a Sender) and a guided route selection example as intuition.

The game describes an asymmetric interaction between two players: a Sender and a Receiver. The Receiver has a privately observed type associated with it ($\theta \in \Theta$) that is sampled from a commonly known distribution ($\theta \sim p_\Theta$). The Sender can send messages to the Receiver and the Receiver can perform actions from a set A . The utilities of the interaction between the players depend on the state of the world $v \in V$ that is sampled independently from the commonly known distribution $v \sim p_V$. The Sender can obtain an observation of the state of the world $\omega \in \Omega$ that is sampled from the commonly known distribution $\omega \sim p_\Omega(\cdot|v)$. The utilities of the interaction between the players are given by two functions $u_s : V \times A \rightarrow \mathbb{R}$ for the Sender, and $u_r : V \times \Theta \times A \rightarrow \mathbb{R}$ for the Receiver.

In our example, θ can correspond to the tolerance or patience exhibited by a driver and influence his utility (see below). The messages sent by the Sender naturally correspond to the traffic management center sending route information. The action chosen by the Receiver corresponds to the driver choosing a specific route. The state of the world corresponds to different traffic conditions across the road network with an appropriate statistic. The traffic management center can monitor the traffic conditions with some degree of uncertainty. The utility functions in our example scenario describe how content the user would be (u_r) if he took a specific route ($a \in A$) given his patience ($\theta \in \Theta$) and current traffic conditions ($v \in V$), and respectively (u_s) how profitable it would be for the traffic management center if the driver adopted a particular route ($a \in A$) given the current traffic conditions ($v \in V$).

The game unfolds as follows:

- The Sender selects a finite set of messages, M , and a disclosure rule $\pi : \Omega \rightarrow \Delta(M)$, where $\Delta(\cdot)$ denotes the space of all distributions over a set. In other words, the disclosure rule specifies the probability $\pi(m|\omega)$ of sending a message m given any possible Sender's observation ω . Note that v is unknown (even through observation) to the Sender

6. WHICH INFORMATION TO DISCLOSE?

at the time of computing this disclosure rule. We will refer to the disclosure rule as the Sender's policy.

- The Sender computes the *effective disclosure rule* $\pi_\Omega(m|v) = \sum_{\omega \in \Omega} \pi(m|\omega)p_\Omega(\omega|v)$.
- The Sender declares and commits to (π_Ω, M) .¹
- The Receiver's private types θ and the state of the world v are independently sampled from p_Θ and p_V , respectively.
- The Sender is supplied with the observation $\omega \sim p_\Omega(\cdot|v)$.
- The Sender samples a message $m \sim \pi(\cdot|\omega)$ and sends it to the Receiver.
- Given the message m , the Receiver performs a Bayesian update to calculate $p_V^m \propto \pi_\Omega(m|\cdot)^T \circ p_V$, where "o" denotes the entry-wise product [76].
- Based on p_V^m and θ the Receiver selects an action $a \in A$.
- Players obtain their respective utilities $u_s(v, a)$ and $u_r(v, \theta, a)$.

6.3 Solving Information Disclosure Games with Two-Sided Uncertainty

To solve the information disclosure game we represent it as a mathematical program (which can be non-linear). Solving such a problem consists of maximizing the expected utility of the Sender by using a particular protocol that chooses what messages to send given its observation of the state of the world. At the same time, the action selection policy of the Receiver contributes the bounding conditions of this mathematical program. In this Section, we analyze such games formally and provide a solution, assuming that the Receiver is fully rational.

6.3.1 Mathematical Program

Since the Sender must commit in advance to its randomized policy, we use a subgame perfect (SP) Bayesian Nash equilibrium where the only choice made by the Sender is the selection

¹In our route selection scenario the above stages correspond to the traffic management center describing and advertising its services.

6.3 Solving Information Disclosure Games with Two-Sided Uncertainty

of the disclosure rule (we analyze the game as if a third party sends the message to the Receiver based on the disclosure rule given to him by the Sender). In the SP equilibrium the Receiver's strategy is the best response to the Sender's policy, simplifying the equilibrium calculations [77].

We limit the possible states of the world V , the Receiver types Θ , the set of observations Ω and the Receiver actions A to finite sets (which we refer to as the finite sets assumption). Let p_V^b denote the beliefs of the Receiver about the state of the world. The Receiver will choose an optimal action:

$$a^* = \arg \max_{a \in A} \mathbf{E}_{v \sim p_V^b} [u_r(v, \theta, a)] \quad (6.1)$$

The set of feasible responses can be limited even further if the disclosure rule π is given. By strategically constructing the rule π , the Sender can influence the actions chosen by the Receiver. Since the Sender has only partial knowledge of the private value θ of the Receiver, the Sender can only compute a prediction of a^* . Denote $p_A : \Delta(V) \rightarrow \Delta(A)$, the Receiver response function and $p_A^m = p_A(\cdot | p_V^m)$. Having precomputed the response function p_A of the Receiver, the Sender can calculate the expected utility of a specific disclosure rule π .

$$\begin{aligned} U_s[\pi] &= \mathbf{E}[u_s] = \sum_{v \in V} \sum_{a \in A} u_s(v, a) p(v, a) \\ &= \sum_{v \in V} \sum_{a \in A} \sum_{m \in M} \sum_{\omega \in \Omega} u_s(v, a) p_V(v) p_A(a | p_V^m) p_\Omega(\omega | v) \pi(m | \omega) \end{aligned}$$

Since we assume that V , Ω and M are finite, we can formulate the disclosure rule construction as an optimization problem over the space of stochastic policies $\pi(m | \omega)$ and the message space M :

$$\pi^* = \arg \max_{M, \pi: V \rightarrow \Delta(M)} U_s[\pi] \quad (6.2)$$

The following theorem shows that if an optimal solution exists, then the set of messages selected by the Sender can be limited to the size of $|\Omega|$.

Theorem 1. *Given an information disclosure game, $\langle V, p_V, \Theta, p_\theta, \Omega, p_\Omega, A, u_r, u_s \rangle$, with the finite sets assumption (i.e. V , Ω and A are finite), if there is an optimal solution (π, M) where $|M| < \infty$, then there exists an optimal solution $(\tilde{\pi}, \tilde{M})$, where $|\tilde{M}| \leq |\Omega|$.*

6. WHICH INFORMATION TO DISCLOSE?

Theorem 1 shows that an optimal solution with a finite message space can be transformed so that the set of messages does not exceed $|\Omega|$. However, it is possible to question whether an optimal solution with a finite message set in fact exists. The following theorem deals with that question, demonstrating that a countable set of messages of an optimal solution can always be replaced by a finite set.

Theorem 2. *Given an information disclosure game, $\langle V, p_V, \Theta, p_\theta, \Omega, p_\Omega, A, u_r, u_s \rangle$, with the finite sets assumption (i.e. V , Ω and A are finite), if the optimal expected utility for Sender U_s is attainable by some protocol (π, M) , then there is an optimal solution with a finite message space.*

We give the complete proofs of Theorems 1 and 2 in Section 6.7 rather than here due to their technicality. Their intuition, however, is easily outlined. For Theorem 1, we show that the effects induced by the extra messages can be achieved by distributing the information that they transfer to other messages without effecting the Sender's utility. The re-distribution process relies on the linear properties of the disclosure rule as a matrix. In turn, for Theorem 2, we show that the utility gains obtained from almost all, but a finite number, of messages is negligible and so is the information which they provide to the Receiver. In fact, they can be aggregated into a single message (thus reducing the total number of used messages to finite) without impacting the Sender's utility.

6.3.2 Finding an Optimal Policy

Unfortunately, directly finding an optimal policy by solving the disclosure rule maximization problem presented in Equation 6.2 is intractable, since it includes a strong non-linear component. More specifically, it assumes availability of the Receiver's best response (defined by Equation 6.1) in a (closed) functional form. However, it is possible to circumvent this hindrance. Instead of assuming a functional best response form, we expand Equation 6.2 by a set of constraints that compare the Receiver's utility from its chosen action to that of all other actions available to him/her. In other words, we transform an explicit (functional) non-linear representation of the Receiver's response into an implicit (constraints-based) linear form.

We begin by generating messages for each possible response from the Receiver. Note that the response will depend on the Receiver's type. Formally, we define a set of functions: $F = \{f : \Theta \rightarrow A\}$. f specifies an action for each Receiver's type. For each function f we create a set of messages. From Theorem 1 we know that for an optimal policy there is need

6.3 Solving Information Disclosure Games with Two-Sided Uncertainty

for $|\Omega|$ messages at most. Therefore, there is no need for more than $|\Omega|$ messages to lead to a specific behavior that is described by a function f . Thus, we create a set M of messages such that, for every $f \in F$, we generate $|\Omega|$ messages denoted by m_f^j , $1 \leq j \leq |\Omega|$.

Using this set of messages with a size of $|\Omega||F|$, we would like to consider possible policies and choose the one that maximizes the Sender's expected utility. However, we need to focus only on policies π where, given a message m_f^j , a Receiver of type θ will really choose an action $f(\theta)$. We achieve this formally by designing a set of inequalities that express this condition as follows.

First, given a message $m \in M$, a Receiver of type $\theta \in \Theta$ and a policy π_Ω , the Receiver will choose an action $a \in A$ only if he believes that his expected utility from this action will be higher than his expected utility from any other action. Note that after receiving a message m , the Receiver's belief that the state of the world is $v \in V$ is proportionate to $p_V(v)\pi_\Omega(m|v)$. Thus, the set of constraints is

$$\forall a' \in A \quad \sum_{v \in V} u_r(v, \theta, a) p_V(v) \pi_\Omega(m|v) \geq \sum_{v \in V} u_r(v, \theta, a') p_V(v) \pi_\Omega(m|v) \quad (6.3)$$

Focusing on a specific message m_f^j , we want to satisfy these constraints for any type $\theta \in \Theta$ and require that the chosen action will be $f(\theta)$. Putting these together after some mathematical manipulations, we obtain the following constraints for $\forall \theta \in \Theta$ and $\forall a' \in A$:

$$\sum_{v \in V} (u_r(v, f(\theta)) - u_r(v, \theta, a')) p_V(v) \pi_\Omega(m|v) \geq 0 \quad (6.4)$$

Note that there may be many functions for which we will not be able to find an effective policy π_Ω that will satisfy the required constraints. However, given such a π_Ω and a function f we can calculate the probability $\pi_A(a|m_f^j)$ that an action $a \in A$ will be chosen when the Receiver receives the message m_f^j , regardless of his type. Formally, given a set $\Theta' \subseteq \Theta$, let $\pi_\Theta(\Theta') = \sum_{\theta \in \Theta'} p_\Theta(\theta_i)$. Then, $\pi_A(a|m_f^j) = \pi_\Theta(f^{-1}(a))$.

Combining all this, we obtain the following optimization problem:

$$\begin{aligned} \tilde{\pi}^* = \arg \max_{\pi} \quad & \sum_{m_f^j \in M} \sum_{a \in A} u_s(v, a) p_V(v) \pi_\Theta(f^{-1}(a)) \pi_\Omega(m_f^j|v) \\ \text{s.t.} \quad & \\ \pi_\Omega = \pi p_\Omega \end{aligned}$$

6. WHICH INFORMATION TO DISCLOSE?

$$\begin{aligned}
& \forall m_f^j \in M, \forall \theta \in \Theta \forall a' \in A \\
& \sum_{v \in V} (u_r(v, \Theta, f(\Theta)) - u_r(v, \theta, a')) p_V(v) \pi_\Omega(m_f^j | v) \geq 0 \\
& \forall \omega \in \Omega \sum_{m_f^j \in M} \pi(m_f^j | \omega) = 1 \\
& \forall m_f^j \in M \pi(m_f^j | \omega) \geq 0
\end{aligned}$$

The complexity of solving the optimization problem within the above algorithm is polynomial in $|A|$, $|V|$ and $|\Omega|$, but exponential in $|\Theta|$ since $|F| \propto |A|^{|\Theta|}$.

We refer to an agent, on behalf of the Sender, which solves the above optimization problem as the Game Theory Based Agent (GTBA).

6.4 People Modeling for Disclosure Games in Multi-attribute Selection Problems

Trying to influence people's action selection presents novel problems for the design of persuasion agents. People often do not adhere to the optimal, monolithic strategies that can be derived analytically. Their decision-making process is affected by a multitude of social and psychological factors [4]. For this reason, in addition to the theoretical analysis, we propose to model people participating in information disclosure games and integrate that model into the formal one. We assume that the agent interacts with each person only once, thus we propose a general opponent modeling approach, i.e., when facing a specific person, the persuasion agent will use models learned from data collected from other people.

The opponent modeling is based on two assumptions on human decision-making:

- **Linear Weighted-Utility:** People's decision-making deviates from the rational choice theory; they use a subjective utility function which is a linear combination of a set of attributes. This utility function may divert from the expected monetary utility function.
- **Logit quantal response (stochastic decision-making):** People do not choose actions that maximize their subjective utility, but rather choose actions proportional to this utility. A formal model of such decision-making was shown in [78, 79] to be of the form:

$$a_r^*(a | \theta, p_V^b) \propto \exp \left(\mathbf{E}_{v \sim p_V^b} [u_r(v, \theta, a)] \right)$$

We name this method for human modeling: Linear weighted-Utility Quantal response (LUQ). (This method was also proved to be successful in modeling human behavior in security games [80].) The study of the general opponent approach and its comparison with the formal

model was done in the context of two games. The Multi-attribute Road Selection Problem with two-sided uncertainty about road traffic and the Sandwich Game with two-sided uncertainty regarding the number of attendees of a particular event. Next we will describe the two games and explain their differences.

6.4.1 Multi-attribute Road Selection Problem with Two Sided Uncertainty

The multi-attribute road selection problem with two-sided uncertainty about the state of the world is an extension of the game that was studied in [40]. It is defined as an information disclosure game Γ_ρ with two players: a driver and a center. The center, playing the role of the Sender, can provide the driver, playing the role of the Receiver, with traffic information about road conditions. In particular, the driver needs to arrive at a meeting place in θ minutes. There is a set H of n highways and roads leading to his meeting location. Each road $h \in H$ is associated with a toll cost $c(h)$. There are several levels of traffic loads L on the roads and a set of highway network states V . A highway network state is a vector $\vec{v} \in V$ specifying the load of each road, i.e., $\vec{v} = \langle l_1, \dots, l_n \rangle$, $l_i \in L$. The traffic load yields a different time duration for the trip denoted $d(\vec{v}_h, h)$ (where \vec{v}_h denotes the traffic load on road h in state \vec{v}). If the driver arrives at the meeting on time he gains g dollars, however he is penalized e dollars for each minute he is late. The chosen road is denoted a . Thus, the driver's monetary utility is given by:

$$u_r(\vec{v}, a, \theta) = g - \max\{d(a, \vec{v}) - \theta, 0\} \cdot e \quad (6.5)$$

The driver does not know the exact state of the highway network, but merely has a prior distribution belief p_V over V . The center also does not know what the exact state of the highway network will be when the driver drives along the chosen road (e.g., even though the traffic flows on a given road, an accident can occur causing the road to be blocked). However, given its observations, the center has a better estimation of the state of the roads. The center has only prior beliefs, p_Θ , regarding the possible meeting times, Θ . Once given the observations on the state, the center sends a message m to the driver which may reveal data about the traffic load of the various roads. The center's utility depends on the actual traffic load and the driver's chosen road $u_s(\vec{v}_a, a)$. It increases with the toll road $c(a)$ and decreases with a 's load as specified in \vec{v} (two examples of such utility functions are given below). The center must decide on a disclosure rule and provide it to the driver in advance (before the center is given some information on the road loads). For the center, the road selection problem is therefore: given a game $\Gamma = \langle H, L, V, \Omega, M, c, d, p_V, p_\Omega, u_s, u_r \rangle$, choose a disclosure rule which will maximize $E[u_s]$.

6. WHICH INFORMATION TO DISCLOSE?

6.4.2 The Sandwich Game

The Sandwich Game is defined as an information disclosure game Γ_σ with two players: a seller and an organizer. The organizer, playing the role of the Sender, can provide the seller, playing the role of the Receiver, with information regarding the anticipated conference attendees. The organizer himself receives noisy information regarding the exact number of attendees (can be interpreted as the number of people who registered to a conference during pre-conference registration). The seller must decide in advance how many sandwiches to prepare for the conference (a). The sandwiches are sold for a fixed price \bar{c} , and it is assumed that each conference attendee will buy a single sandwich. Each seller is associated with a private type θ which indicates the cost of preparing each possible number of sandwiches. Thus the seller's monetary utility given the number of attendees (v), the number of sandwiches prepared (a) and θ is given by $u_r(v, \theta, a) = \min\{a, v\} \cdot \bar{c} - \theta(a)$. Depending on the actual conference size, the organizer is assumed to have some preferences as to the number of sandwiches that should be prepared by the seller ($u_s(v, a)$).

6.4.3 Hypothesis

In the original Road Selection problem presented in [40], which considered only one-sided uncertainty, the agent using the general opponent modeling approach achieved a significantly higher utility than the GTBA agent. The major cause for this effect is that people preferred not to choose jammed roads in the game even when they could arrive on time to their meetings and consequently they did not attempt to maximize their monetary values. Thus, we hypothesized that a similar agent (relying on the LUQ method for human modeling) for the two-sided uncertainty Road Selection problem would also outperform the GTBA agent in the extended game. In addition we designed the Sandwich Game, a new game in which the goal of the players is to maximize their monetary values. We expected that, in such situations, people would be more motivated to maximize their expected monetary values and GTBA may perform similar to an agent which relies on the LUQ method for human modeling.

6.4.4 Non-monetary Utility Estimation for the Road Selection Problem with Two-Sided Uncertainty

Given a game $\Gamma_\rho = \langle H, L, V, \Theta, M, c, d, p_V, p_\Theta, u_s, u_r \rangle$, based on the LUQ method for human modeling, we assume that the driver chooses the road based on a non-monetary subjective

6.4 People Modeling for Disclosure Games in Multi-attribute Selection Problems

utility function, denoted \bar{u}^{Γ_ρ} (here and in the functions defined below, we omit Γ_ρ when it is clear from the context). We further assume that \bar{u} is a linear combination of three parameters given the chosen road: travel time, road load and road toll. We associate different weights (α_s) with each of these parameters: α_d for the trip duration time, α_c for the toll cost, and for all $l_i \in L$ we have α_{l_i} . That is, given a game Γ_ρ , assuming that the driver knew the highway network load \vec{v} and chose road a :

$$\bar{u}_\rho(\vec{v}, a) = \alpha_d \cdot d(\vec{v}, a) + \alpha_c \cdot c(a) + \alpha_{\vec{v}_a} \quad (6.6)$$

Note that the utility associated with a given road depends only on the given road and its load and not on the load of other roads according to the state.

We assume that the user uses logit quantal response and therefore, given Γ_ρ , we assume that the driver will choose road h with a probability of

$$p(a = h | \Gamma_\rho, \vec{v}) = \frac{e^{\lambda \bar{u}_\rho(\vec{v}, h)}}{\sum_{h' \in H} e^{\lambda \bar{u}_\rho(\vec{v}, h')}} \quad (6.7)$$

where λ is a parameter. However, since $\bar{u}_\rho(\vec{v}, h)$ has an extra degree of freedom, we set $\lambda = 1$.

When choosing an action, the driver does not know \vec{v} but only m . Thus, the probability the driver will choose a road h is:

$$p(a = h | \Gamma_\rho, m) = \frac{e^{E[\bar{u}_\rho(\cdot, h | m)]}}{\sum_{h' \in H} e^{E[\bar{u}_\rho(\cdot, h' | m)]}}$$

Consider a set of games \mathcal{G}_ρ such that they all have the same set of levels of traffic load.

In order to learn the weights of the subjective utility function associated with \mathcal{G}_ρ , we assume that a set of training data Ψ is given. The examples in Ψ consist of tuples (Γ_ρ^i, m, a) specifying that a subject playing the driver's role in the game $\Gamma_\rho^i \in \mathcal{G}_\rho$ chose road $a \in H$ after receiving the message $m \in M$. We further assume that there is a predefined threshold $\tau > 0$, and for each m that appears in Ψ there are at least τ examples. Denote by $prop(\Gamma_\rho^i, m, a)$ the fraction of examples in Ψ of subjects who, when playing Γ_ρ^i and receiving message m , chose road a .

Next, given Ψ we aim to find appropriate α s that minimize the mean square error between the prediction and the actual distribution of the actions given in the set of examples Ψ . Note that we propose to learn α s across all the games in \mathcal{G}_ρ . Formally we search for α s that minimize

$$\sum_{\Gamma^i, m, h} (p(a = h | \Gamma_\rho^i, m) - prop(\Gamma^i, m, h))^2.$$

6. WHICH INFORMATION TO DISCLOSE?

One may notice that the subjective utility function that we propose does *not* depend on the meeting time θ . This is because the meeting time θ is a private value of the driver and therefore is not specified in the examples in Ψ . However, since we are interested in the expected overall response per message of the whole population and not in predicting each individual response, if the distribution of the meeting time is left unchanged, dependence on the meeting time is embedded in the utility results. (We actually learn p_A^m directly and therefore do not depend on θ).

Next, given a specific Γ_ρ , we incorporate the learned function $p(a = h|m)$ as an instantiation of p_A^m into the calculation of the expected utility of a disclosure rule:

$$U_s[\pi] = \sum_{\vec{v} \in V} \sum_{h \in H} \sum_{m \in M} \sum_{\omega \in \Omega} u_s(\vec{v}, h) p_V(\vec{v}) p_\Omega(\omega|v) \pi(m|\omega) p(h|m).$$

Unfortunately, this means that $U_s[\pi]$ has a very non-trivial shape (involving positive and negative exponential and polynomial expressions of its argument), and even such properties as convexity were hard to verify analytically. As a result, we chose to use the standard pattern search algorithm in order to find a reasonable approximation of the optimal disclosure rule with respect to $U_s[\pi]$.

6.4.5 Non-monetary Utility Estimation for the Sandwich Game with Two-Sided Uncertainty

Based on the LUQ method for human modeling we assume that the seller decides on the number of sandwiches to prepare based on the following subjective utility function: $\alpha_1 \cdot \min\{a, v\} + \alpha_2 \cdot \max\{(a - v), 0\}$. That is, the seller tries to maximize the number of sandwiches sold, and minimize the number of sandwiches thrown away (we anticipate that α_2 will be negative). For similar reasons to those mentioned in Section 6.4.4, the proposed subjective utility function does not depend on θ . According to LUQ we assume a logit quantal response. Learning the α s and building an optimal policy is conducted in a method identical to that of the road selection problem. Each of these proposed agents that rely on the LUQ method (for each of the two domains) will be called a LUQ Agent (LUQA).

6.5 Experimental Evaluation

Our experiments were aimed at answering three questions:

1. How well would the game theory-based agent that finds the optimal policy of the information disclosure game perform, assuming that people choose the best response according to u_r (GTBA)?
2. Would LUQA improve the Sender's results in comparison to GTBA?
3. Do the answers to the above questions depend on the domain and, if so, given a domain, can we provide a way to predict whether LUQA or GTBA will perform better?

6.5.1 Experimental Design

In both games the subjects were given the description of the game including the Sender's preferences. Before starting to play, the subjects were required to answer a few questions verifying that they understood the game. For each subject, the center received a state that was drawn randomly and sent a message using the disclosure rule described in section 6.2. To support the subjects' decision-making process, we presented them with the distribution over the possible states that was calculated using the Bayesian rule given the message, the prior uniform distribution and the center's policy. That is, the subjects were given $p_V^M(m)$. The subjects then selected a single action (either a number of sandwiches to prepare or a road). For motivation, the subjects received bonuses proportionate to the amount they gained in dollars. Comparisons between different means were performed using t-tests.

We considered two variations for each of the two games (the sandwich game and the road selection game). The first one was used to answer the first question and to collect data for the opponent modeling procedure. The second variation was used to answer the second question, using the collected data of the first variation as the training data set. We now describe the parameters used for both variations of the sandwich game and the road selection game.

Road Selection Game

In the first game, Γ^1 , the players had to choose one of three roads: a toll free road, a \$4 toll road or an \$8 toll road (i.e. $H = \{h_1, h_2, h_3\}$, $c(h_1) = 0$, $c(h_2) = 4$ and $c(h_3) = 8$). Each road could either have flowing traffic which would result in a 3 minute ride, heavy traffic which would take 9 minutes of travel time or a traffic jam which would cause the ride to take 18 minutes. That is, $L = \{flowing, heavy, jam\}$, and $d(h_i, flowing) = 3$, $d(h_i, heavy) = 9$ and $d(h_i, jam) = 18$, for all $h_i \in H$. An example of a state v could

6. WHICH INFORMATION TO DISCLOSE?

be $\langle \text{heavy}, \text{flowing}, \text{flowing} \rangle$, indicating that there is heavy traffic on the toll free road and traffic is flowing on the other two toll roads. Arriving on time (or earlier) yields the player a gain of \$23 and he will be penalized \$1 for every minute that he is late. Finally, the meeting could take place in either 3, 6, 9, 12 or 15 minutes, i.e., $\Theta = \{3, 6, 9, 12, 15\}$. Thus $u_r(\vec{v}, a, \theta) = 23 - \max\{d(a, \vec{v}) - w, 0\} \cdot 1$. The prior probabilities over V and W were uniform.

The center's utility was as follows: if the subject took the toll free road, the center received \$0 regardless of the state. If the subject took the \$4 toll road, the center received \$4 if the traffic was flowing, \$2 if there was heavy traffic and \$0 if there was a traffic jam. If the subject took the \$8 toll road, the center received \$8 if the traffic was flowing, \$2 if there was heavy traffic and lost \$4 if there was a traffic jam.

In the second game, Γ^2 , the meeting time was changed to take place in 12, 13, 14 and 15 minutes, i.e., $\Theta = \{12, 13, 14, 15\}$. The center's utility was also changed: the center received \$1 if the driver chose the most expensive road among those with the least traffic. Otherwise the center received \$0.

Sandwich Game

The conference size (v) had either no participants (a canceled conference), 20 participants (a small conference), 30 participants (a medium conference), 40 participants (a large conference) or 50 participants (a huge conference).

The number of sandwiches prepared by the seller (a) was in $\{0, 20, 30, 40, 50\}$ as well. Recall that the seller's utility function is given by $u_r(v, \theta, a) = \min\{a, v\} \cdot \bar{c} - \theta(a)$. We set \bar{c} (the sandwich retail price) to \$1. We used three different private types (θ), which indicate the cost of preparing each possible number of sandwiches. Table 6.1 shows the different private types used.

We considered two different utility functions for the organizer in the sandwich game. In the first game, Γ_σ^1 , the system wanted the seller to prepare more sandwiches than needed, unless the conference had 50 attendees. In Γ_σ^2 the system wanted the seller to prepare less sandwiches than needed, unless the conference was canceled (0 attendees). The utility function was chosen such that the utility for the organizer and the seller would be different and not linearly dependent. The observation table is shown in Table 6.2. As depicted in the table, if the organizer observes that the conference will be canceled, then in fact it will be. In any other case there is an 85%

Table 6.1: Seller types

Number of sandwiches	Cost for Type 1	Cost for Type 2	Cost for Type 3
None	\$0	\$0	\$0
10	\$5	\$8	\$12
20	\$9	\$12	\$15
30	\$14	\$16	\$18
50	\$20	\$20	\$20

chance that the organizer will observe the correct state. Even if the state observed is incorrect, the actual state is not too far off, unless the conference is unexpectedly canceled.

6.5.2 Human Subjects

In the experiments, subjects were asked to play either the sandwich game or the multi-attribute road selection game with two-sided uncertainty. As mentioned above, each of the games had two different variations which differed in the system utility function. Each subject played only once. All of our experiments were run using Amazon’s Mechanical Turk service (AMT) [56]¹. A total of 713 subjects from the USA, 56.2% females and 43.8% males, participated in our study. The subjects’ ages ranged from 18 to 74, with a mean of 34 and a standard deviation of 11.3. The subjects participated in the following experiments:

- 173 subjects participated in Γ_{ρ}^1 , which is the first game played in the road selection game, using GTBA.
- 102 subjects participated in Γ_{ρ}^2 , which is the second game played in the road selection game, using GTBA.
- 119 subjects participated in Γ_{ρ}^2 , which is the second game played in the road selection game, using LUQA.

¹For a comparison between AMT and other recruitment methods see [7].

6. WHICH INFORMATION TO DISCLOSE?

Table 6.2: Observation table in the sandwich game

Observation	Probability of actual conference size				
	Canceled	Small	Medium	Large	Huge
Canceled	1	0	0	0	0
Small	0.08	0.85	0.06	0.01	0
Medium	0.02	0.06	0.85	0.06	0.01
Large	0.01	0.03	0.06	0.85	0.05
Huge	0.06	0	0.03	0.06	0.85

- 100 subjects participated in Γ_σ^1 , which is the first game played in the sandwich game, using GTBA.
- 106 subjects participated in Γ_σ^2 , which is the second game played in the sandwich game, using GTBA.
- 113 subjects participated in Γ_σ^2 , which is the second game played in the sandwich game, using LUQA.

Since the experiment was based on a single multiple-choice question, we were concerned that subjects might not truly attempt to find a good solution. Recall that we took several measures to encourage truthful answers in all the experiments described in this thesis (See the introduction in Chapter 1). In addition to these measures, since this experiment was based on a single question, we removed 6 answers which were produced in less than 10 seconds as the response was considered unreasonably fast. However, since the average time needed to solve our task was 83 seconds, we concluded that the subjects considered our tasks seriously.

6.5.3 Experimental Results

In both the sandwich game and the multi-attribute road selection game with two-sided uncertainty, we first let the subjects play with the GTBA agent. This agent computes the game theory-based policy of Γ^1 , to solve the maximization problem presented in section 6.3. Note

that even though the complexity of solving this problem is high, we were able to find the optimal policy for the multi-attribute selection games in a reasonable amount of time.

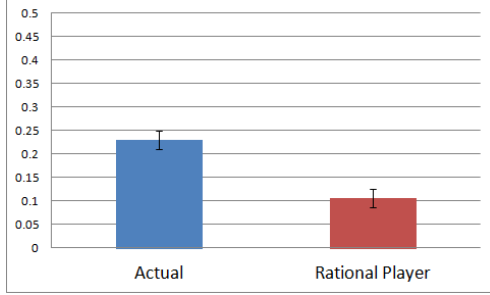


Figure 6.1: System utility in road game Γ_ρ^1 . The center gained a significantly higher utility from the actual users than the utility it would have gained if all of the users were rational ($p < 0.001$)

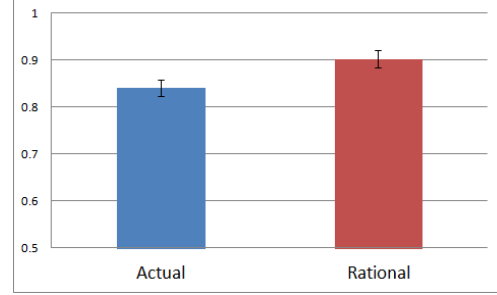


Figure 6.2: User utility in road game Γ_ρ^1 . The actual drivers gained a significantly lower utility, on average, than they would have gained if they all would have acted rationally ($p < 0.001$).

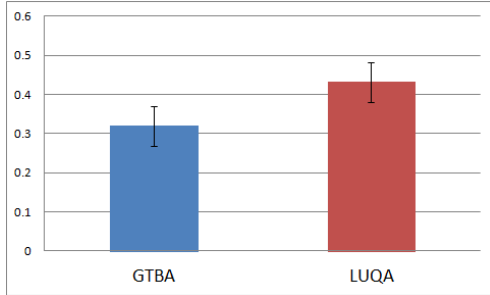


Figure 6.3: System utility in road game Γ_ρ^2 . The center performed significantly better when using LUQA rather than GTBA ($p < 0.05$).

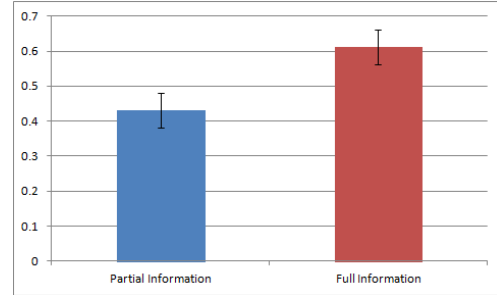


Figure 6.4: System utility for LUQA in road game Γ_ρ^2 . LUQA performed significantly better when it received full information ($p < 0.05$).

6.5.4 Results of the Multi-attribute Road Selection Game with Two-sided Uncertainty

GTBA results

The policy of GTBA using the first settings (Γ_ρ^1) included 13 messages, but 5 of them were generated with a very low probability. Thus, from the 169 subjects who participated in the

6. WHICH INFORMATION TO DISCLOSE?

experiment, most of them (166 subjects) received one of 8 messages, and 3 of the subjects each received a different message.

The center received, on average, 0.230 per driver. This result is significantly ($p < 0.001$) higher than the utility that the center would have received if all of the subjects were rational (i.e., maximizing u_r), which, in expectation, was only 0.105 per driver (see Figure 6.1). As can be seen in Figure 6.2, user performance significantly dropped from that of fully rational. Another deviation from full rationality was observed by the correlation between the time to the meeting and the road selection. For a fully rational player, the longer he has until the meeting, the less likely he is to choose a toll road. However, this negative correlation between the time to the meeting and the road selection was as low as -0.015 , suggesting that subjects almost ignored the meeting time. These observations lead to the conclusion that in the multi-attribute road selection game with two-sided uncertainty, humans tend to concentrate on the traffic on each road and its toll, but ignore the actual monetary value which supports our general opponent modeling approach for this domain.

LUQ human model

We tested four different methods of modeling human decision-making:

1. Rational, which assumes that humans always choose the road which maximizes their expected monetary value given in Equation 6.5. This method is the method assumed by the GTBA agent and does not require any additional parameters.
2. QRE (logit quantal response), which assumes that the probability that humans choose a road is proportionate to the expected monetary value from that road. This method is based on Equation 6.7, however, it assumes that the drivers base their utility function on the monetary value ($u_r(\vec{v}, a, \theta)$) given in Equation 6.5) rather than using the subjective utility function ($\bar{u}_\rho(\vec{v}, a)$) (as assumed by LUQ). Therefore, this method has a single parameter: λ .
3. LWU (Linear Weighted-Utility), which assumes that humans always choose the road which gives them the highest subjective utility (using the subjective utility function $\bar{u}_\rho(\vec{v}, a)$ given in Equation 6.6. This method has 5 parameters.
4. LUQ, which combines both linear weighted-utility function and logit quantal response, given in Equations 6.6 and 6.7. This method has 5 parameters.

Table 6.3: Mean square error of modeling human decision-making

Modeling Method	Mean Square Error (the lower the better)
Rational	0.89
QRE	0.295
LWU	0.194
LUQ	0.065

Table 6.3 presents the mean square error for all four methods on the data from Γ_ρ^1 using a leave-one-out cross validation (in which for each of the messages, when the mean square error is evaluated on a messages, the parameters are learned using data from all other messages). Clearly, LUQ’s prediction outperforms all other methods.

Comparing LUQA and GTBA

Using the settings of the second game, Γ_ρ^2 , we ran two agents, GTBA and LUQA. We used the results obtained from the 166 subjects that played Γ_ρ^1 as the training set data Ψ for LUQA. That is, the α s for $\bar{u}_r^{\Gamma_\rho^2}$ were learned from the subjects playing Γ_ρ^1 , i.e., $\mathcal{G} = \{\Gamma^1\}$. LUQA and GTBA each generated 4 messages for Γ_ρ^2 . 119 subjects played with LUQA and 102 with GTBA. LUQA performed significantly better ($p < 0.05$) than GTBA, gaining an average of 0.431 vs. 0.319 points per driver (see Figure 6.3).

We also checked the actual dollars earned by the subjects. Unfortunately, when playing with LUQA the average virtual gain per subject was only \$19.00, while when playing with GTBA the average was higher, \$21.20. These results differ significantly, hinting that the center’s gain was on account of the driver’s monetary utility. This result is compatible with our previous result in [38], where people tend to perform better when the agent confronting them assumes that they will act rationally. However, in practice, this issue isn’t of great concern, since, if the center is interested in the driver receiving a higher utility, it may implicitly add the driver’s utility to its own utility function and result with a protocol that will be better for both the center and the driver.

6. WHICH INFORMATION TO DISCLOSE?

One-sided uncertainty vs. Two-sided uncertainty

In previous work [40] we tested the performance of LUQA in the road selection problem under the exact same settings, only with full information for the center. Figure 6.4 shows these results along with our current results with partial information. As can be seen, when LUQA has full information, it significantly outperforms LUQA with partial information. This is not surprising, since additional information allows the Sender to avoid mistakes and encourages the Receiver to take actions which are more favorable to the Sender.

6.5.5 Sandwich Game Results

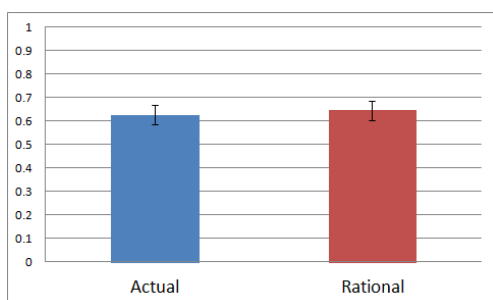


Figure 6.5: User utility in sandwich games. The difference between a fully rational seller and the actual human sellers is minor and not statistically significant.

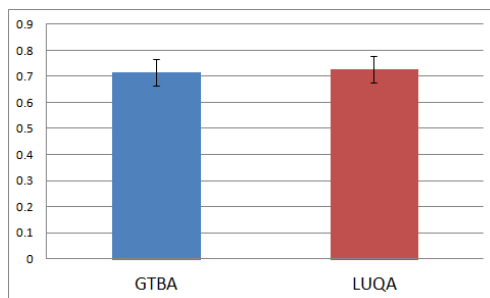


Figure 6.6: System utility in sandwich game Γ_{σ}^2 . The difference between the organizer utility when using LUQA and when using GTBA is minor and not statistically significant.

GTBA results

The monetary result plays an important role in the sandwich game. This is because the game is played in an environment in which a person's goal is to make as high a revenue as possible, which usually results in selling as many sandwiches as possible while minimizing the number of sandwiches thrown away.

The policy of GTBA in the first settings (Γ_{σ}^1) included 5 messages. The organizer received on average 0.260 per seller (Figure 6.7). The utility of the organizer was similar to the expected utility that the organizer would receive if all subjects were rational (i.e., maximizing u_r), which, in expectation, was 0.299 per seller. We suspect that this is due to the important role that the monetary value played in this game. These results differ from the correspondence results of the

road selection game and thus we hypothesized that LUQA would not be needed for this game and that the GTBA agent would do as well as LUQA in this domain.

LUQA and the LUQ human model

The learning phase for LUQA, which was based on the subjects who participated in Γ_σ^1 , revealed the following parameters in the subjective utility function: α_1 , which is the amount gained by each sandwich sold, was 0.087, and α_2 , which is the amount lost by each sandwich thrown away was -0.103 . On average (depending on the private type w), if the expected monetary values are maximized, people should be neutral to missing a sandwich or preparing one too many sandwiches. However, apparently people were a little risk averse since $|-0.103| > |0.087|$, though the numbers are very close. When testing the MSE of LUQ, the result was similar to that of QRE (quantal response under expected monetary outcome), both yielding 0.07. The similar performance for both LUQ and MSE indicates that people performed nearly rationally, which virtually obviates the usage of LUQ.

Comparing LUQA and GTBA

The comparison was done under the second set of settings (Γ_σ^2), and both GTBA and LUQA used 4 messages. The organizer received on average 0.715 when using GTBA, and 0.728 when using LUQA (Figure 6.6). Although LUQA did perform slightly better, the results do not differ significantly. This is not surprising since, as mentioned above, the subjects' subjective utility was very close to the expected monetary value and thus the GTBA's assumptions were correct. We suggest that the slight improvement shown was due to the logit quantal response assumption.

6.5.6 Deciding between LUQA and GTBA

As demonstrated in the above two games, there are situations where LUQA outperforms GTBA, while in other situations they yield similar results. One may recommend to always use LUQA since it is always as good as GTBA and sometimes even better. However, LUQA requires collecting data to learn the human utility function. Therefore, we recommend to first collect some data using GTBA and compare the agent's results and the human behavior to the rational behavior. If GTBA's results are significantly different from the expected results that it would have

6. WHICH INFORMATION TO DISCLOSE?

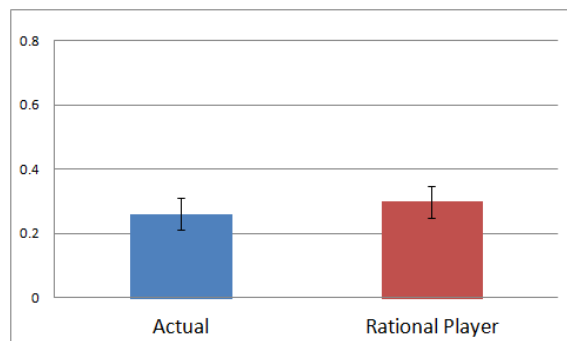


Figure 6.7: System utility in the sandwich game. The difference in the organizer’s utility between actual users and the utility it would have gained if all of the users were rational is minor and not statistically significant.

received if people would have followed a rational decision-making process, then it is worthwhile to collect more data and use LUQA. Otherwise, using GTBA seems to be an adequate heuristic. It is important to note that in the road selection game it was ample to use 10 subjects in order to obtain a significant difference between GTBA and the expected utility if people would have followed rational behavior. In the sandwich game, we did not obtain significant results even with 100 subjects. Consequently collecting 20 – 25 data points in order to make a decision whether to use LUQA or GTBA seems reasonable.

6.6 Conclusions

In this chapter we considered information disclosure games with two-sided uncertainty in which an agent tries to lead a person to take an action that is beneficial to the agent by providing him with truthful, but possibly partial, information relevant to the action selection. We first provided an algorithm to compute the optimal policy for information disclosure games with two-sided uncertainty, assuming that the human is fully rational. We also provided an innovative machine learning-based model that effectively predicts people’s behavior in these games. The model we provided assumes that people use a subjective utility function which is a linear combination for all given attributes. The model also assumes that while people use this function as a guideline, they do not always choose the action with the greatest utility value. Nonetheless, the higher an action’s utility value the more likely they are to choose that action. We integrated this model into our persuasion model in order to yield an innovative method of human behavior manipulation. Extensive empirical study in multi-attribute road

selection games with two-sided uncertainty confirms the advantage of the proposed model in that game. However, in another domain we tested, i.e., the Sandwich game, there is no significant advantage to the machine learning-based model, and using the game theory-based agent which assumes that people maximize their expected monetary values is beneficial. We propose a methodology of how to choose between the two options. We argue that, depending on the domain, people's decision-making process may vary and thus where in one domain modeling humans as rational may be sufficient, in another domain it is too far from their actual behavior and therefore an agent that assumes perfectly rational behavior may fall far behind.

6.7 Proofs of Theorems Concerning Message Space

6.7.1 Proof of Theorem 1

Proof. Let (π, M) be an optimal solution to the game so that $|M| > |\Omega| = n$. We will first show that certain transformations of π produce a left stochastic matrix structure (in which the rows correspond to messages and the columns to observations) with at least one zero row, i.e. produce disclosure rules that use less messages than the original π . We will then show a specific transformation of π that, while reducing the number of used messages, preserves the utility gained. We will thus obtain a new optimal disclosure rule with fewer messages. Since $|M| < \infty$, iterative application of the above process would lead to an optimal $(\tilde{\pi}, \tilde{M})$, where $|\tilde{M}| \leq n$ as required.

Notice again that zero rows in π correspond to the messages that were never sent, and we would be able to reduce the size of M without changing the utility in any way. Assume that after the elimination of zero rows, we still have a set of messages greater than $|\Omega|$ or there never were any.

Since π is a stochastic matrix, there can be no more than n elements in it equal to 1. If all are present, the rest of the rows are zero, and we can reduce M to have only n elements without changing π , thereby obtaining the necessary optimal solution properties. If this does not occur, i.e. there are less than n elements in π equal to 1, we can proceed with the following reasoning.

Denote π_m the m 'th row of π . It holds $\sum_{m \in M} \pi_m = \vec{1}_n^T$, where $\vec{1}_n$ is a column vector in R^n with all elements equal to 1. Since there are at least $n + 1$ rows in π , but only n columns, π has a non-trivial kernel space of left multiplication vectors. Hence, there is a non-trivial row vector $\alpha = (\alpha_m)_{m \in M}$ so that $\phi = \alpha\pi = \vec{0}_n^T$, and for all $m \in M$ $|\alpha_m| \leq 1$ and for some $m_1 \in M$ $\alpha_{m_1} = 1$. This can be achieved by taking an arbitrary non-trivial kernel row vector and scaling it appropriately.

6. WHICH INFORMATION TO DISCLOSE?

Clearly, $\pi_{m_1}(\omega) < 1$ for all $\omega \in \Omega$. Otherwise, for some $\bar{\omega} \in V$ $\pi_{m_1}(\bar{\omega}) = 0$ for all $m \neq m_1$, and $\phi(\bar{\omega}) = \alpha_{m_1} = 1 \neq 0$, hence contradicting $\phi = \vec{0}$.

Denote $\tilde{\pi}$ a matrix with rows defined by $\tilde{\pi}_m = (1 - \alpha_m)\pi_m$. Notice that all elements of $\tilde{\pi}$ are non-negative. Furthermore, they are not greater than 1, due to the following:

$$\begin{aligned}\vec{1}_n^T &= \sum_{m \in M} \pi_m = \vec{1}_{|M|}^T \pi \\ \vec{0}_n^T &= \alpha \pi \\ \vec{1}_n^T &= (\vec{1}_{|M|}^T - \alpha) \pi = \vec{1}_{|M|}^T \tilde{\pi}\end{aligned}$$

Since all elements of $(\vec{1}_{|M|}^T - \alpha)$ are non-negative, as are elements of π , the last equation means that elements of $\tilde{\pi}$ are bounded by 1, and the sum of rows is $\vec{1}_n^T$.

Hence $\tilde{\pi}$ is also a valid solution to the game. Furthermore, it uses less messages since $\alpha_{m_1} = 1$ and $\tilde{\pi}_{m_1} = (1 - \alpha_{m_1})\pi_{m_1} = \vec{0}_{|M|}$.

Applying the above reasoning in an iterative fashion, we can reduce the number of non-zero rows in $\tilde{\pi}$ to n . Denote \widetilde{M} to be the subset of M that corresponds to those rows, which will be the new set of messages.

We will now show that $\tilde{\pi}$ has the same utility as π , hence $(\tilde{\pi}, \widetilde{M})$ will be an optimal disclosure rule with $|\widetilde{M}| = n$, concluding the proof.

Denote $U_s^M[\pi_m] = \sum_{v \in V} \sum_{a \in A} \sum_{\omega \in \Omega} u_s(v, a) p_V(v) p_A(a | p_V^m) p_\Omega(\omega | v) \pi(m | \omega)$, then $U_s[\pi] = \sum_{m \in M} U_s^M[\pi_m]$. Notice that $U_s^M[\gamma \pi_m] = \gamma U_s^M[\pi_m]$, since p_V^m is insensitive to scaling of π_m .

Let us now compute $U_s[\tilde{\pi}]$, where $\tilde{\pi}$ was computed using a vector $\gamma \alpha$ with $\gamma \in R$.

$$\begin{aligned}U_s[\tilde{\pi}] &= \sum_{m \in M} U_s^M[\tilde{\pi}_m] \\ &= \sum_{m \in M} U_s^M[(1 - \gamma \alpha_m) \pi_m] \\ &= \sum_{m \in M} (1 - \gamma \alpha_m) U_s^M[\pi_m] \\ &= U_s[\pi] - \gamma \sum_{m \in M} \alpha_m U_s^M[\pi_m] \\ &= U_s[\pi] - \gamma * U_{diff}\end{aligned}$$

If $U_{diff} \neq 0$, then for $\tilde{\pi}$ computed for $\gamma = \text{sign}(U_{diff})$ we find that $U_s[\tilde{\pi}] \geq U_s[\pi]$, hence contradicting the optimality of π . Therefore, $U_{diff} = 0$, (setting $\gamma = 1$) and $U_s[\tilde{\pi}] = U_s[\pi]$, making $(\tilde{\pi}, \widetilde{M})$ an alternative optimal solution with $|\widetilde{M}| = |\Omega|$, as required.

□

6.7.2 Proof of Theorem 2

Proof. The following proof is stated for a countable infinity of messages. However, since the space of all possible conditional message probabilities π_m is compact, it is easy to recast it for continuous message indices.

Let (π, M) be an optimal solution to the problem, so that $|M| = \infty$, and furthermore for an infinite number of messages $\pi_m p_V > 0$. In other words there is an infinite number of messages that have a non-zero probability to appear. and w.l.g. assume that all messages are such. Notice also that w.l.g. we can assume that $u_s(v, a) > 0$ for all $a \in A$ and $v \in V$. Denote $u_s^{min} = \inf_{p(v,a) \in \Delta(V \times A)} \mathbf{E}[u_s(v, a)] > 0$, and notice that $U_s^M[\pi_m] \geq u_s^{min}$ for any π_m . Similarly notice that $u_s^{max} = \sup_{p(v,a) \in \Delta(V \times A)} \mathbf{E}[u_s(v, a)] < \infty$ and that $U_s^M[\pi_m] \leq u_s^{max}$.

Since the sum of all message probabilities is equal to 1, and all utilities are strictly positive, the sequence of partial sums $\sum_{i=0}^t U_s^M[\pi_{m_i}]$ monotonically increases and is bounded, hence

$U_s[\pi] = \sum_{i=0}^{\infty} U_s^M[\pi_{m_i}] < \infty$ and is well defined. Furthermore, for any $\epsilon > 0$ exists $T < \infty$ so

that $\sum_{i=T+1}^{\infty} U_s^M[\pi_{m_i}] \leq \epsilon$. Consider setting $\epsilon = u_s^{min}$ and set $\tilde{\pi}_m = \begin{cases} \pi_{m_i} & m = m_i, i \in [0, T] \\ \sum_{i=T+1}^{\infty} \pi_{m_i} & m = \tilde{m} \end{cases}$.

It holds that $U_s^M[\tilde{\pi}_{\tilde{m}}] \geq u_s^{min} \geq \sum_{i=T+1}^{\infty} U_s^M[\pi_{m_i}]$. Therefore, $U_s[\pi] \leq U_s[\tilde{\pi}]$, and $(\tilde{\pi}, \tilde{M})$ is a finite disclosure rule with a utility at least as good as the original solution (π, M) . Hence, if the optimal U_s is obtainable, then there is a finite disclosure rule that achieves it. □

6.8 List of Notations

notation	meaning
a	Receiver action.
a^*	Receiver <i>optimal</i> action.
A	set of possible actions for Receiver.
$c(h)$	toll cost.
\bar{c}	fixed price for sandwiches.
d	time duration of a trip.

6. WHICH INFORMATION TO DISCLOSE?

f	a function from Ω to A .
F	set of functions from Ω to A .
h	highway road.
H	set of highways.
l	road load.
m	message.
m_f^i	message.
M	set of messages.
n	number of highways ($ H $).
p_V	distribution over the sates of the world.
p_Ω	distribution over the observations.
p_θ	distribution on receiver type.
p_A	Receiver response function.
p_A^m	Receiver response function <i>given</i> message m .
p_V^b	Receiver's belief over the state of the world.
p_V^m	Receiver's belief over the state of the world given message m .
u_s	utility for Sender.
U_s	optimal expected utility for Sender.
u_r	utility for Receiver.
v	state of the world.
\vec{v}	highway network state (state of the world in the road selection problem).
V	set of possible states of the world.
\circ	entry-wise product.
$\Delta(\cdot)$	the space of all distributions over a set.
Γ	an information disclosure game.
π	Sender disclosure rule.
π_Ω	Sender <i>effective</i> disclosure rule.
π^*	Sender <i>optimal</i> disclosure rule.
θ	Receiver type.

6.8 List of Notations

Θ	Receiver set of types.
ω	observation on the state of the world.
Ω	set of possible observations.

Table 6.4: List of Notations

6. WHICH INFORMATION TO DISCLOSE?

Persuasion Method Matters

7.1 Introduction

In this chapter we face a new challenge. While in all previous chapters the main challenge of the system or automated agent was which information (or advice) to present to the user, in this chapter we focus on *how* to present information to the user. While many systems are designed to encourage users to accept beneficial proposals, complex propositions may often confuse the user and make the decision non-trivial. It is well known that problem presentation [31, 33, 81] may have an impact on the human decision-making process. In our work we consider beneficial proposals that comprise several gains or losses, which are associated with varying probabilities and must be accepted or rejected together. We will compare two possible presentation methods (for the possible outcomes and their associated probabilities), a separate presentation and a combined presentation, for each proposal. Many real life situations resemble our problem.

Our first example is a medical system which assists a doctor in encouraging a patient to take a certain medication. The medication is associated with one or more benefits, such as curing the infection or reducing pain, and also with several side effects, such as headaches, nausea, a rash or an allergic reaction. These outcomes have varied significance; for instance, a headache might be slightly unpleasant whereas an allergic reaction could be life threatening. Each of these outcomes is also associated with a certain probability; for example, the probability of overcoming the infection may be 90% while the side effect of a headache might occur in 20% of the patients, whereas an allergic reaction might only become evident in 0.5% of the patients. The expected overall reaction to the medication must be positive (otherwise it is more harmful than helpful). In order to decide whether to use a medication, all of the potential benefits

7. PERSUASION METHOD MATTERS

and the risk of side effects must be evaluated together. Combining the various components is associated with a cost, since it is unclear how to quantify a headache compared to a rash. Different people may associate different values with each benefit or side effect. Therefore assigning values to the components using a joined metric would require some effort, such as questioning many people on their preferences, and thereafter impose a cost.

Another example is an investment adviser who is trying to build an investment portfolio for one of his customers. Some stocks have a higher risk but also offer an opportunity to receive greater interest, while on the other hand a bond may have a lower risk however with a lower interest level. Most people combine different stocks and bonds, which results in a combination of different levels of risk. The investment adviser's primary goal is to get the customer to invest her money. Consequently the adviser would like to show the portfolio to the customer using the most appealing presentation. Should the investment adviser show the expected probability and value of revenue (or loss) for each stock, or should he try to combine all stocks in a single chart which presents the total investment?

Our last example is a travel agent who would like to promote the sales for a specific vacation package. Every day of a multiple-day vacation has some probability of rain or heat load (the strength of the rain or heat load may also vary). The travel agent wants to show the customer the probabilities for rain on each of the planned days. How should the travel agent present these probabilities (while his goal is to sell the package)? Should he present them for every day separately, or should he combine them all into one chart?

In order to determine how to present complex proposals, we propose an automated agent that utilizes behavioral economic theory. A *prospect* is a lottery (possibly with several outcomes, where each outcome has its own probability) [82], and a *simple prospect* is a prospect with some probability p to gain or lose some amount x and otherwise to gain or lose nothing. The problems we study in this chapter are composed of several simple prospects. These prospects must either **all** be accepted or **all** be rejected (there is no option to accept a partial set of prospects) and the system gains from accepted proposals. In the medical system we described earlier, the benefit of *cure infection* with a probability of 90% (and 10% of not curing the infection) is an example of a prospect, and similarly so is the side effect of acquiring a *headache* with a probability of 20% (and 80% of not resulting in a headache). All of the prospects must be selected or rejected together since a patient either takes the medication or does not. The agent we propose must decide whether to present the proposal in a separate method, as is, or in a combined method, combining all of the simple prospects into a single

(more complex) prospect. For example, in the medical system the separate method would list all of the separate prospects. In combined presentation combining the curing prospect with the headache prospect results with a 72% chance that infection will be cured without the headache side effect appearing, an 18% chance that the infections will be cured and a headache will appear, an 8% that the infection will not be cured and no headache will appear and finally a 2% chance that the infection will not be cured and that a headache will appear. Note that in the combined method all of the probabilities add up to 100%. We assumed, and show experimentally, that presenting the problem as a set of simple prospects or as a combined prospect is not necessarily equivalent and can affect people's choices. Thus the automated agent will determine when to use a separate presentation and when to use a combined presentation in order to encourage the users to accept the propositions.

When several prospects are proposed, the issue of bracketing arises. Read et al. [83] introduced the term "Choice Bracketing" to mean the grouping of choices. It has been shown that when people face several choices in which each choice has several options, they tend to treat such choices separately rather than treat them as a single decision. Our agent must take this into account when considering whether to use a combined or separate presentation for a problem, since similarly to what has been shown on separate choices, people might also treat each prospect separately even if the prospects are presented as part of a group.

Behavioral economic theory describes the decision processes that people use when deciding whether to accept a prospect or reject it. The most significant theories in this field are the Expected Utility Hypothesis [84], the Prospect Theory [82] and the Cumulative Prospect Theory [85]. We embed these theories into our agent in order to model the expected human choice that will be made for a given set of prospects, in order to determine if the separate or combined presentation should be used.

We introduce the Prospect Presentation Problem, along with its formal description. This problem requires selecting whether to represent the multiple prospects in a separate presentation or a combined presentation, while maximizing the system's utility. We use different decision process models and settings in order to compose an agent that is capable of solving the Prospect Presentation Problem. We demonstrate the efficiency of the agent, in choosing the better presentation method, using an extensive experimental evaluation.

7.2 Human Decision Making Under Uncertainty Hypotheses

7.2.1 Expected Utility Hypothesis

The Expected Utility Hypothesis (EUH) was initiated by Bernoulli in 1738 [84]. Under this hypothesis, people have a utility function, u , which associates any possible total wealth with some utility. People use this function when deciding whether to accept or reject a lottery simply by maximizing their expected utility. For example, a person with a current total wealth of $\$W$ facing a prospect (lottery), P , with a probability of p to win $\$x$ and a probability of $1 - p$ to lose $\$y$, will compare the expected utility from accepting the offer:

$$U(P) = u(W + x) \cdot p + u(W - y) \cdot (1 - p) \quad (7.1)$$

with the expected utility from rejecting the offer, which is simply $u(W)$. The person will accept the lottery if the former is greater and otherwise reject it. A common utility risk averse function, suggested by Bernoulli, is the log function:

$$u(X) = \log(X) \quad (7.2)$$

7.2.2 Prospect Theory

The Prospect Theory was presented by Kahneman and Tversky in [82] and later refined to the Cumulative Prospect Theory (CPT) in [85]. The Prospect Theory is based on three principles. The first is that people do not take into account their total wealth when accepting or rejecting an uncertain opportunity (as suggested by the expected utility hypothesis [84]), but rather use their current wealth as a baseline, and will be happy if they win an amount and become upset if they lose an amount. The second principle is loss aversion, where people hate losing more than they like winning. The third principle is that people have a subjective representation of probabilities and do not interpret probabilities fully rationally, but rather use their own decision weights when deciding whether to reject or accept a gamble. In his book, Kahneman [86] (p.314) gives the following examples: The decision weight that corresponds to a 90% chance is 71.2%, while the decision weight that corresponds to a 10% chance is 18.6%. According to these examples, people are likely to prefer a guaranteed outcome of \$80 than to gamble with a 90% chance of winning \$100, since the latter is only worth \$71.2 to them. Tversky and Kahneman elicited these weights by sequentially asking subjects to choose between a specific lottery and many different guaranteed outcomes. The equivalent to the given lottery for a certain subject was

set to the average between the greatest rejected guaranteed outcome and the smallest accepted guaranteed outcome [85]. However, these decision weights depend on people's personalities, their wealth, culture and the scope of the payoff in question. The cumulative prospect theory determines the value of any prospect based on its possible outcomes and the probability of each of its outcomes. Given a prospect P which comprises T possible ordered outcomes (as defined by Tversky and Kahneman), $\{x_1, x_2, \dots, x_T\}$, and the first t are negative outcomes, i.e. $x_1 < x_2 < \dots < x_t < 0 \leq x_{t+1} < \dots < x_T$. Each outcome is associated with some probability $p(x)$. The value of the prospect is given by the following formula:

$$U(P) = \sum_{i=1}^t v(x_i) \cdot \left(w\left(p_i + \sum_{j=1}^{i-1} p(j)\right) - w\left(\sum_{j=1}^{i-1} p(j)\right) \right) + \sum_{i=t+1}^T v(x_i) \cdot \left(w\left(p_i + \sum_{j=i+1}^T p(j)\right) - w\left(\sum_{j=i+1}^T p(j)\right) \right) \quad (7.3)$$

where $v(x)$ stands for the value function and w is the weighting function (the decision weight function described above). Both of these functions must be non-decreasing, and $w(0) = 0, w(1) = 1$. v is negative for losses and positive for gains, and $v(0) = 0$. The intuition behind this formula is that every possible value of the output is assumed to have an impact which is proportionate to the marginal affect that its accumulated probability has on the weighting function. For example, given a prospect P' with three possible outcomes, \$2, \$3 and \$10 (no negative outcomes) with probabilities of $p(\$2) = 0.1, p(\$3) = 0.7, p(\$10) = 0.2$, the value of the prospect is given by:

$$U(P') = v(\$10) \cdot w(0.2) + v(\$3) \cdot (w(0.9) - w(0.2)) + v(\$2) \cdot (w(1) - w(0.9))$$

Note that due to the nature of w and v , the value of P' is at least $v(\$2)$. This corresponds with the fact that the prospect guarantees a win of at least \$2.

Tversky and Kahneman suggested the value function:

$$v(x) = \begin{cases} x^\alpha & \text{if } x \geq 0 \\ -\mu(-x)^\beta & \text{if } x < 0 \end{cases} \quad (7.4)$$

where α, β and μ are parameters, and the weighting function is:

$$w(p) = \frac{p^\gamma}{(p^\gamma - (1-p)^\gamma)^{1-\gamma}} \quad (7.5)$$

7. PERSUASION METHOD MATTERS

where γ is a parameter used for positive payoffs and is replaced by a different parameter, δ , for negative payoffs. Several studies try to estimate parameters for the Prospect Theory [87, 88, 89], however most studies try to maximize the likelihood of the results obtained by each subject individually. This approach could not be applied in our work since we built a model based on a group of users and apply the model to new users (for whom we have little or no data). Models that are built for each and every user will not allow us to generalize them to new subjects.

7.2.3 Bracketing

“Choice Bracketing”, termed by Read et al. [83], designates the grouping of individual choices together into sets. “Broadly Bracketing” indicates that the decision-maker takes all choices into account when making his decision, while “Narrow Bracketing” indicates that the decision-maker isolates each choice from all other choices. When humans face a broad spectrum of topics, where each topic consists of several options, they usually make a decision on each topic separately. A classic experiment that illustrates narrow bracketing was done by Tversky and Kahneman [49]. They asked their subjects the following question:

”Imagine that you face the following pair of concurrent decisions. First examine both decisions, then indicate the options you prefer:

Choice I. Choose between:

- A. A guaranteed gain of \$240.
- B. A 25% chance to gain \$1000 and a 75% chance to gain nothing.

Choice II. Choose between:

- C. A guaranteed loss of \$750.
- D. A 75% chance to lose \$1000 and a 25% chance to lose nothing.”

Since people tend to be risk averse with a positive payoff and risk seeking with a negative payoff, a large majority of subjects (73%) chose both A and D. Only 3% of the subjects chose B and C. Combining A and D yields a 25% chance to gain \$240 and a 75% chance to lose \$760. However, combining B and C dominates this with a 25% chance of gaining \$250 and a 75% chance of losing \$750. Tversky and Kahneman performed an additional experiment in which they presented only the following combined choices to the subjects, that is:

Choose between:

A+C. A guaranteed loss of \$510.

A+D. A 25% chance to gain \$240 and a 75% chance to lose \$760.

B+C. A 25% chance to gain \$250 and a 75% chance to lose \$750.

B+D. A 6.25% chance to gain \$1000, a 56.25% chance to lose \$1000 and a 37.5% chance to gain or lose nothing.”

This time, not a single subject chose the dominated option (A+D). This experiment demonstrates that people tend to treat each decision on its own and do not combine the choices, unless they are explicitly combined for them.

7.3 Prospect Presentation Problem

The *prospect presentation problem* is a decision problem for a system and is defined as follows. We first define a set of simple prospects $s = \{P_1, P_2, \dots, P_{k_i}\}$; recall that a simple prospect P is composed of a probability P_p of gaining or losing a certain amount, P_x . In the prospect presentation problem, a system has a set of n sets of simple prospects, $S = \{s_1, s_2, \dots, s_n\}$. Each set $s \in S$ must be offered to h human clients. Each of the human clients may either accept the set of prospects s (and participate in the lotteries associated with the prospects) or reject it. Each of the sets, s , may be presented to the human clients in two different presentation modes $m(s)$; the presentation mode may either be *separate*, which indicates that the set of prospects are presented separately (as they are), or *combined*, which indicates that the prospects are *validly* combined into a single prospect. The separate presentation mode of s is denoted s^s and the combined presentation mode of s is denoted s^c . The probability for any possible outcome must be identical in both s^s and s^c . A cost $c(m(s))$ may be applied to the system and may depend on the method of presentation. The system gains a utility of $1 - c(m(s))$ every time a human client accepts a set of prospects. The human clients are assumed to follow a stochastic decision policy, in which, given a set of prospects, s , and a presentation mode, m , $p(s, m)$ determines the probability that the humans will accept the set of prospects. The *prospect presentation problem* is intended for the system to determine the presentation mode, $m(s)$, for each of the sets of prospects, s , in order to maximize:

$$\sum_{s \in S} p(s, m(s)) \cdot h \cdot (1 - c(m(s))) \quad (7.6)$$

7.4 An Agent for the Prospect Presentation Problem

In this section we introduce an Agent for the **P**rospect **P**resentation **P**roblem (APPP). Section 7.4.1 describes how APPP calculates the combined presentation for a set of prospects and solves the prospect presentation problem. However, this solution relies on a component that accurately models human decision policy. We therefore propose several alternatives for APPP’s composition of the human model in Section 7.4.2.

7.4.1 Solving the Prospect Presentation Problem

The first component of the APPP agent is described in Algorithm 1, which handles the task of efficiently (linear in output length) calculates a combined presentation for a set of prospects s^c . The algorithm receives a set of simple prospects and outputs a hash map with all of the possible outcomes (as keys) and probabilities (as values). The algorithm iterates via all prospects. In every iteration the algorithm takes the previous iteration’s result and doubles it, once assuming that the current prospect obtained its outcome and once assuming that the current prospect did not yield its outcome. For example, consider a set of simple prospects in which one prospect has a 25% chance of winning \$37 and another prospect has a 60% chance of losing \$10, i.e. $s = \{(0.25, \$37), (0.60, \$ - 10)\}$. In the first iteration, with the prospect of $(0.25, \$37)$, there will be only two possible outcomes, 0 with a probability of 0.75 and \$37 with a probability of 0.25. In the second (and last) iteration, with the prospect of $(0.60, \$ - 10)$, the algorithm will first assume that the (negative) outcome was not obtained (with a probability of 0.40), and thus there will be two possible outcomes: \$37 with a probability of $0.25 \cdot 0.40 = 0.10$ and \$0 with a probability of $0.75 \cdot 0.40 = 0.30$. Then the algorithm will add two additional outcomes, assuming that the outcome of the second prospect, $-\$10$, was obtained (with a probability of 0.60): $\$27 - \$10 = \$17$ with a probability of $0.25 \cdot 0.60 = 0.15$ and $\$0 + -\$10 = -\$10$ with a probability of $0.75 \cdot 0.60 = 0.45$.

The second component of the APPP agent is described in Algorithm 2. This is the procedure that solves the prospect presentation problem. The input for Algorithm 2 is a set of prospect sets, a cost function and a human decision policy. The output is a determination policy for each set of prospects. The determination policy determines whether to use the *separate* or the *combined* method for each prospect set. This algorithm simply iterates via all sets of prospects and calculates, for each of the sets, which of the presentation methods is more profitable for the system, i.e. whether $p(s, \text{separate}) \cdot (1 - c(\text{separate}))$ is larger than

7.4 An Agent for the Prospect Presentation Problem

Algorithm 1 Calculation of the combined presentation for a set of prospects.

Input: s - A set of simple prospects $s = \{P_1, P_2, \dots, P_k\}$, where $P_i = (P_i^p, P_i^x)$.

Output: s^c - A hash map holding all possible outcomes as keys and their associated probabilities as values.

```

1:  $s^c[0] \leftarrow 1$ 
2: for each prospect  $P$  in  $s$  do
3:    $s^{c'} \leftarrow s^c$ 
4:   clear  $s^c$ 
5:   for each outcome in  $s^{c'}$  do
6:      $s^c[\text{outcome}] \leftarrow s^{c'}[\text{outcome}] \cdot (1 - P^p)$  1
7:      $s^c[\text{outcome} + P^x] \leftarrow s^{c'}[\text{outcome}] \cdot P^p$ 
8: return  $s^c$ 

```

$p(s, \text{combined}) \cdot (1 - c(\text{combined}))$ or vice versa. If the human decision policy were known,

Algorithm 2 Procedure for solving the prospect presentation problem.

Input: A set composed of prospect sets S , a cost function $c(m(s))$, and a human decision policy $p(s, m(s))$.

Output: A determination policy, m , for each of the prospect sets.

```

1: for each problem  $s$  in  $S$  do
2:   if  $p(s, \text{separate}) \cdot (1 - c(\text{separate})) > p(s, \text{combined}) \cdot (1 - c(\text{combined}))$  then
3:     set  $m(s) \leftarrow \text{separate}$ ;
4:   else
5:     set  $m(s) \leftarrow \text{combined}$ ;

```

the algorithm would have fully solved the prospect presentation problem. However, in real life, a system agent facing the prospect presentation problem is not likely to have access to the human decision policy, $p(s, m(s))$. Therefore, the major concern for an agent facing the prospect presentation problem is to accurately model the human decision policy.

7.4.2 Decision Policy Modeling in APPP

An agent facing the prospect presentation problem does not have specific information about the human user, and therefore must use a general model for modeling human decision policy.

¹In 6 and 7, if $s^c[\text{outcome}]$ or $s^c[\text{outcome} + P^x]$ already have a value, increment that value by $s^{c'}[\text{outcome}] \cdot (1 - P^p)$ or $s^{c'}[\text{outcome}] \cdot P^p$, respectively.

7. PERSUASION METHOD MATTERS

While several theories may be considered for building this model, the Expected Utility Hypothesis and Cumulative Prospect Theory definitely stand out as being significant in their field. We therefore decided to embed each of these theories in APPP’s decision policy model. Nevertheless, it is possible to embed other theories into the agent. However, in both the expected utility hypothesis and the cumulative prospect theory, a human’s response does not depend on the form of the presentation of the problem. This implies that both presentation methods (*combined* and *separated*) would yield the same probability that the user will accept a given lottery regardless of the lottery, i.e. for any s , $p(s, \textit{combined}) = p(s, \textit{separate})$. This assumption is clearly inappropriate for our work (and is shown to be false in the results section). Consequently, each of the theories requires slight modification when considering the *separate* representation method, by taking the bracketing effect (see section 7.2.3) into account.

In the following subsections we describe in detail how each of the theories can be embedded into APPP. All of the methods we tested needed to set some parameters, therefore APPP required training data. The data set, ψ , was composed of a set of tuples $\langle s, m(s), d \rangle$, in which s is a set of simple prospects presented to a human user, $m(s)$ determines whether the set of prospects were presented in the *combined* method or in the *separated* method, and d is a Boolean, indicating whether or not the user decided to participate in the lottery. In order to accurately model human decision-making, it is essential to assume stochastic decision-making, since the agent is required to evaluate the probability that a user will accept a lottery. Not assuming stochastic decision-making would mean that a lottery (possibly depending on its presentation method) would either be accepted by everyone or rejected by everyone, i.e. $p(s, m(s)) \in \{0, 1\}$. (It is not necessary to assume that every individual actually uses stochastic decision-making, but that the population as a whole can be modeled as using stochastic decision-making.)

APPP assumes a logit quantal response and thus relies on Equation 7.7. Recall that in the prospect presentation problem, the user needs to choose between participating in a prospect (or a set of simple prospects in the *separate* presentation method) or not. Thus, the user must actually choose between the lottery and the value of not participating in it, denoted $U(\textit{null})$. In EUH, $U(\textit{null}) = u(\$W)$, where W is the person’s initial wealth, and in CPT, $U(\textit{null}) = v(0) = 0$. Therefore, given a lottery L , according to Equation 7.7, the probability that a user will accept the lottery is:

$$p(L) = \frac{1}{1 + e^{\lambda(U(\textit{null}) - U(L))}} \quad (7.7)$$

Modeling using the Prospect Theory with Learned Parameters

The first model we consider for APPP's decision policy is the cumulative prospect theory. We used the cumulative prospect theory, CPT, (see section 7.2.2) in order to evaluate $U(L)$ for a user facing the *combined* method (a single prospect). However, as mentioned above, when considering the *separated* method (a set of simple prospects), APPP deviates from CPT. For the instances in the data set using the *combined* method of presentation, in which the user is only presented with a single prospect (s^c), APPP calculates $U(s^c)$ according to Equation 7.3 (and Equations 7.4 and 7.5 for the value and weighting functions, respectively). APPP assumes that people who face the separated method evaluate each prospect separately and then combine all values together to receive the total value of the lottery¹. This assumption is based on the bracketing effect (see section 7.2.3), which suggests that people treat each problem separately, and thus we assume that they will evaluate each prospect separately. Formally, given a set of prospects, s , presented in the *separated* method, the value of the set of prospects is given by:

$$U(s^s) = \sum_{P \in s} U(P) \quad (7.8)$$

APPP searches for parameters $\alpha, \beta, \mu, \gamma, \delta$ and λ that minimize the mean squared error (MSE) between $p(s, m(s))$ and the actual fraction of users in (ψ) who accepted lottery s ($d = true$) from all of those who were shown the lottery using $m(s)$ as the presentation method. Once APPP has set the parameters to use for the decision policy, it can be used to determine how to present a new set of prospects. Given a set of prospects, s , and a presentation mode, $m(s)$; if the presentation mode is *combined*, APPP uses Equations 7.7 and 7.3 and parameters $\alpha, \beta, \mu, \gamma, \delta$ and λ . If the presentation mode is *separated*, APPP uses Equations 7.7, 7.8 and 7.3 and the parameters ($\alpha, \beta, \mu, \gamma, \delta$ and λ) to evaluate the probability that the user will accept the associated lottery.

Modeling using the Prospect Theory with Original Parameters

We also investigated another possible model; it is the same model described in 7.4.2, but rather than learning the parameters $\alpha, \beta, \mu, \gamma$ and δ , APPP uses the parameters proposed by Kahneman and Tversky. That is, $\alpha = 0.88, \beta = 0.88, \mu = 2.25, \gamma = 0.61$ and $\delta = 0.69$. APPP only uses the data to evaluate λ .

¹This approach in an environment of only simple prospects is actually similar to the assumptions made by the prospect theory rather than CPT.

7. PERSUASION METHOD MATTERS

Modeling using Expected Utility Hypothesis

As the Expected Utility Hypothesis is very well known, we also show how to embed it into APPP. APPP, based on EUH, uses Equations 7.1 and 7.2¹. As in the model assuming CPT, we must determine how to account for sets of prospects presented in the *separated* method. However, when using EUH, APPP may not simply use the exact same approach as when using CPT, since in EUH, $U(P)$ includes the initial wealth. Therefore, if we were simply to add up the utilities from all of the prospects (using Equation 7.8), eventually we would be adding the initial wealth several times. Thus we propose a simple modification, using the following equation:

$$U(s^s) = u(W) + \sum_{P \in s} (U(P) - u(W)) \quad (7.9)$$

APPP based on EUH, searches for parameters W and λ that minimize the mean squared error (MSE) between $p(s, m(s))$ and the actual fraction of users in (ψ) who accepted lottery s ($d = true$) from all those who were shown the lottery using $m(s)$ as the presentation method. Given a set of prospects, s , and a presentation mode, $m(s)$; if the presentation mode is *combined*, APPP uses Equations 7.7 and 7.1 and parameters W and λ . If the presentation mode is *separated*, APPP uses Equations 7.7, 7.9 and 7.1 and the parameters (W and λ) to evaluate the probability that the user will accept the associated lottery.

7.5 Evaluation

7.5.1 Experimental Setup


We ran our experiments using Amazon’s Mechanical Turk (AMT) [56]. We constructed a total of 120 sets of simple prospects, with k (the number of simple prospects in a set) varying between 3 and 5. The upper boundary of 5 was chosen since the number of possible outcomes is exponential in the number of k ($2^5 = 32$) and we didn’t want to present too many possible outcomes to the subjects. Each simple prospect (P) had a random probability (P_p) between 1% and 100% (only whole probabilities) and a random expected outcome ($P_p \cdot P_x$) between $-\$15.00$ and $+\$15.00$. For the human decision not to be trivial, every set of prospects had at least one prospect with a negative outcome and one with a positive outcome. For ethical

¹In practice we also tried two different power functions, but they yielded the exact same results as the log function (presented in Section 7.5.2).

Suppose you are facing the following set of lotteries,
you may either participate in all lotteries or reject them all: (# 7 of 20)

- A 1.00% chance to **lose** \$445.40, (and a 99.0% chance to gain/lose nothing).
- A 79.00% chance to **gain** \$2.40, (and a 21.0% chance to gain/lose nothing).
- A 83.00% chance to **gain** \$3.30, (and a 17.0% chance to gain/lose nothing).

The following pie charts which present the lotteries may assist you in making your decision:



Which option would you choose:

☐ Reject all lotteries.

☐ Participate in all lotteries.

Figure 7.1: A subject facing a set of prospects presented separately.

reasons, since we did not want to encourage traditional gambling, we ensured that all gambles had a positive expected utility. Therefore, a player who was trying to maximize her expected outcome should have accepted all gambles. Thus, the subjects were not urged into a gamble which was not good for them. We recruited a total of 612 participants. 58.1% of the subjects were males and 41.9% were females. Subjects' ages ranged from 18 to 73, with a mean of 31.6, a median of 29 and a standard deviation of 11.0. All subjects were residents of the USA. The subjects were paid 30 cents to participate in the experiment. Each subject was presented with 20 sets of prospects, half in their original form and half as combined prospects. This resulted in an average of approximately 50 instances for each of the 120 sets of prospects for each of the two different presentation modes. The subjects were given the following instructions: "Suppose you are facing the following lottery / set of lotteries, you may either participate in it / in all lotteries or reject it / them all". (The exact text depended on the mode of presentation of the current set of prospects.) To enhance comprehension, we also provided the subjects with pie-charts presenting the prospects (as in [90]). The following explanation was provided: "The following pie chart(s) which present(s) the lottery / lotteries may assist you in making your decision." Figure 7.1 presents an example of a screen-shot of a subject facing a set of prospects presented separately, and Figure 7.2 presents an example of a screen-shot of a subject facing

7. PERSUASION METHOD MATTERS

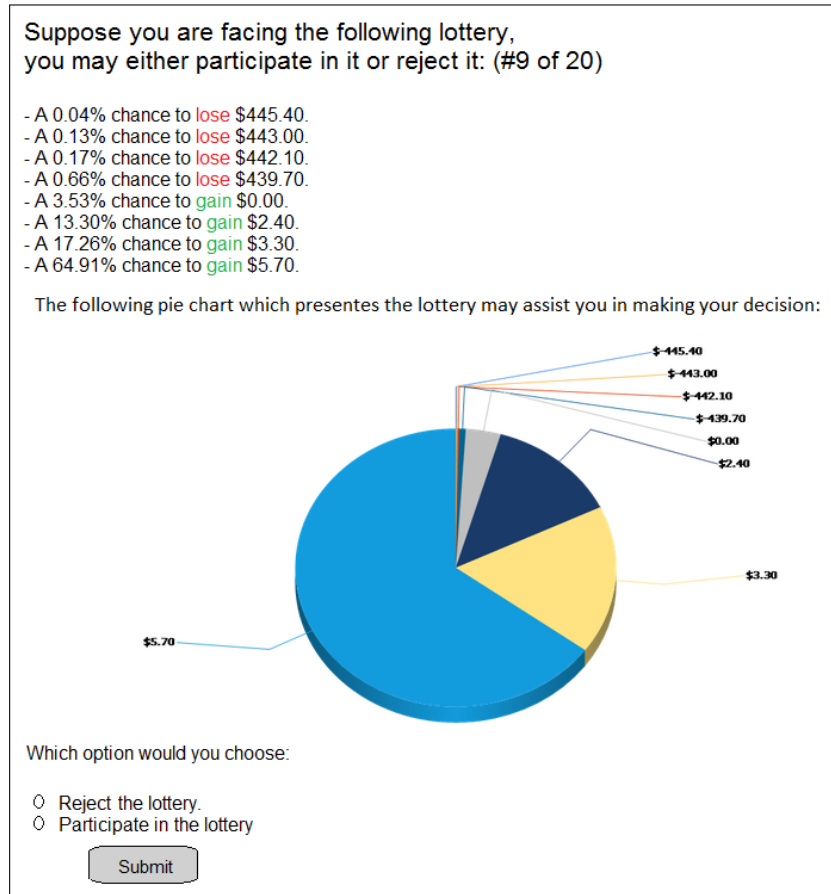


Figure 7.2: A subject facing a set of prospects presented in the combined mode.

the same set of prospects presented in the combined mode. Note that there are 3 prospects in the separate mode, and thus $2^3 = 8$ in the combined mode. We set the cost function at 0 for the separated method and at 0.15 for the combined method. This setting was chosen since we assume that the problem is provided as separate prospects, and therefore some cost is associated with presenting a combined prospect to the user. A cost of 0.15 for the *combined* mode, equalizes the performance of the *combined* mode to the *separate* mode.

7.5.2 Results

We ran APPP using 10-fold cross-validation on the data. In every fold APPP was trained on 108 sets of prospects, and tested on the remaining 12 sets of prospects. For each of these sets, APPP had to decide whether to present the prospects separately or combined. Figure 7.3

presents the performance of APPP in three modes. The APPP refers to the agent using the learned parameters as described in Section 7.4.2, APPP-KT refers to the agent that uses the parameters described by Kahneman and Tversky as explained in Section 7.4.2 and APPP-EUH is the agent described in 7.4.2. These versions of the agent are compared to an agent that always presents the *combined* mode and an agent that always presents the *separated* mode. The black bars represent the 95% confidence interval using a paired t-test. APPP significantly ($t(119) = 2.3645$, $p < 0.01$ using a paired t-test on the score obtained from every set of prospects) outperformed all other methods, and yielded an increase of 6% in the average score over the two baselines. Recall that APPP may only control the method of presentation to the users (and not the actual lotteries), therefore this achievement is very impressive. Table 7.1 provides additional details regarding the acceptance rate of APPP and the baseline agents. As can be observed by the table, the combined mode enjoyed a much greater acceptance rate than the separate mode. This clearly demonstrates that the presentation mode has an impact on the human acceptance rate, justifying our initial assumptions. However, recall that presenting the combined method requires some additional effort and is therefore assumed to be associated with some cost. If this cost is reduced, the *combined* mode becomes much more appealing, and vice versa, as the cost increases, the *separated* mode becomes more appealing. It is not surprising that APPP yields a slightly lower acceptance rate than the combined agent, since, as mentioned, many more subjects accept the combined mode than the separated mode, and APPP chose to present the combined mode only in 37.5% of the sets. APPP-EUH did not present the combined mode in any of the sets. This was not because it assumed that using the separate mode is more appealing to the user, but because it was not willing to pay the cost associated with presenting the combined mode (if the cost was totally removed, APPP-EUH would present the combine mode in 94% of the sets). Recall that the expected outcome on all lotteries was always positive, yet on average people still accepted less than 50% of the lotteries.

Regarding the pie-charts, 79.1% of the subjects said that the pie-charts helped them and only 19.6% said that they did not help them. Only 6 subjects (less than 1%) said that they did not understand the pie-charts. While the total average of participation in the lotteries was 41.5%, females participated on average in only 37.9%, which is significantly lower ($p < 0.01$ using ANOVA test) than males who participated on average in 44.1% of the lotteries. This indicates that females are more risk averse than males (this finding was also present in many studies such as [91]). We also found that young people, up to age 29 (which was our median),

7. PERSUASION METHOD MATTERS

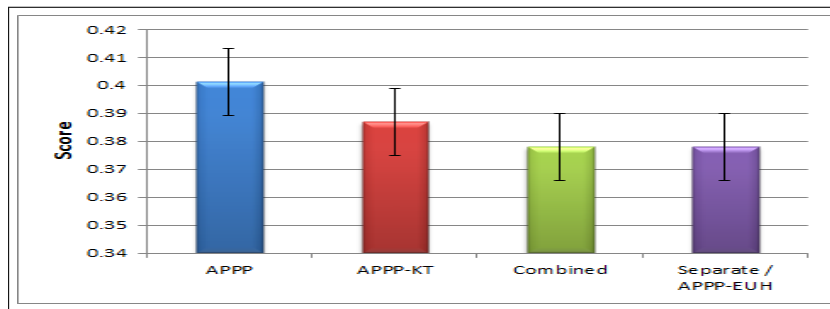


Figure 7.3: Average score obtained with each of the methods.

Agent	Acceptance rate	Combined presentation	Average cost	Score	Number of parameters
APPP	42.8%	37.5%	0.0266	0.401	6
APPP-KT	39.6%	16.7%	0.009	0.387	1
Combined	44.4%	100%	0.066	0.378	0
Separate/APPP-EUH	37.8%	0%	0	0.378	0 – 2

Table 7.1: Average performance of APPP compared to the other agents

are significantly ($p < 0.01$) more likely to participate in the lottery than people aged 30 and above (44.0% vs. 37.7%). Risk aversion also seems to increase with education, as subjects with only a high school education (49.6%) participated in 43.3% of the lotteries, while subjects with a bachelor's or a master's degree or a PhD participated in only 39.7% of the lotteries (these results differ statistically with $p < 0.05$, however, these results may be explained by the fact that the younger subjects tend to have lower education). Interestingly, while the gap between the males and females participating in the lotteries was smaller when the prospects were combined (6.93% vs. 5.62%), the gap between the young subjects and the older ones almost doubled in size when the presentation mode was *combined* (4.29% vs. 8.31%). This finding may encourage future work on a personalized agent based on demographic data alone.

7.5.3 Discussion

In this study we used $k \leq 5$ (the number of simple prospects in a set). What affect an increase in the number of prospect selection problems will have remains indeterminate. In such a case, the pie chart may become very complex as the number of outcome options increases exponentially. One option is to group together similar outcomes, but still it is unclear how this should be done. Another interesting question is what will happen with lower values of k . Lower values of k should enable a better evaluation of the problem and a better comprehension of the agent's advice; it remains to be seen how this will affect the users' interaction with the agent. We chose the pie charts (combined and separate) in order to enhance comprehension of the proposed prospects. The users provided positive feedback to the charts. It would be interesting to study how much of a role they played in the user acceptance rates. Enhancing APPP to include more presentation modes or other visual enhancements would be an interesting extension of this study. Using CPT with learned parameters was shown to outperform other methods we tested. In the future we would like to consider modifying the functions suggested by Tversky and Kahneman to see if it is possible to further improve APPP's performance. Another possible extension would be to study how to apply alternative modeling theories such as the priority heuristic [92].

7.6 Conclusion

In this chapter we introduced the *prospect presentation problem*, in which users are presented with sets of prospects that must be accepted or rejected as a group (such as an investment

7. PERSUASION METHOD MATTERS

portfolio). We refer to a system that gains a positive utility when clients accept the prospects (decide to invest). We defined the *prospect presentation problem* as determining, for each set of prospects, which presentation mode to use in order to maximize the system's utility. We presented the APPP agent that solves the *prospect presentation problem* and chooses between representations of complex beneficial proposals. APPP uses the Cumulative Prospect Theory (CPT) in order to model human decision-making and uses this model to select the better representation method for a set of prospects. We demonstrated that fixing the presentation method to always present the prospects separately, or always present them combined, results in a lower score than using APPP to select the presentation mode based on the human model and a given set of prospects. We investigated several variations on decision process models and found that using CPT results in the most successful agent for selecting the presentation mode. Furthermore, learning the parameters for CPT from sets of prospects presented in the two presentation modes resulted in a better agent than simply using the parameters elicited by Kahneman and Tversky in their original study. We show that the combined method yields much higher acceptance rates, and thus, if the cost associated with presenting the combined method is very low, it may be more beneficial for the system to always use the combined mode. If the cost is not negligible, we suggest that APPP be used to determine which sets to present separately and which to present combined.

7.7 List of Notations

notation	meaning
$c(m(s))$	cost applied to a presentation mode.
h	number of human clients.
$m(s)$	presentation mode of a simple prospect.
n	number of simple prospects.
p	probability.
P	prospect (lottery).
s	simple prospect.
S	set of simple prospects.
s^c	combined presentation mode of s .

s^s	separate presentation mode of s .
T	set of possible outcomes of a prospect.
$u(X)$	utility of amount of money.
$U(P)$	utility of prospect.
v	value function.
W	wealth.
x	a possible monetary outcome.
α	cumulative prospect theory parameter for positive payoffs.
β	cumulative prospect theory parameter for negative payoffs.
γ	cumulative prospect theory parameter for positive payoffs.
δ	cumulative prospect theory parameter for negative payoffs.
λ	logit quantal response parameter.
μ	cumulative prospect theory parameter for negative payoffs.

Table 7.2: List of Notations

7. PERSUASION METHOD MATTERS

Final Remarks

In this thesis we study the possibility of deploying automated agents for human persuasion. In our settings, the automated agents and the humans have different goals. Nonetheless we assume that the automated agents do not compete with the humans either.

We presented a methodology for the automated agent. This methodology relies heavily on building a human model. The human model relies on running machine learning techniques on collected data. It also benefits from social science insights, such as hyperbolic discounting, logit quantal response and risk aversion. Using the human model the automated agent searches for an optimal action for the system to perform.

We considered three different types of persuasion methods: advice provision, information disclosure and presentation methods. We used our methodology for all of these presentation methods. We also showed the success of this methodology in several different settings, including one in which the system presented advice on several different parameters, another in which the system presented a set of actions to the user, a setting in which the system presented long-lasting advice and a setting in which the system presented advice repeatedly to the user. We also tested this methodology in several different domains, such as navigation systems, climate control systems and movie recommender systems. We conclude that to build an effective method for human persuasion one must build a human model, which is best achieved when using machine learning on given data and relying on principles known from social science.

Most of the work presented in this thesis was conducted using Amazon's Mechanical Turk platform, which is a crowd sourcing web service that coordinates the supply and demand of tasks that require human intelligence to complete. Amazon's Mechanical Turk has become an

8. FINAL REMARKS

important tool for running experiments with human subjects and was established as a viable method for data collection [7]. We took several actions to encourage subjects to truly attempt to answer seriously: we only selected workers with a good reputation; a set of questions, designed to verify comprehension of the task, was presented to the subjects prior to executing the task; and as a stimulus, all subjects were guaranteed a monetary bonus proportionate to their performance. Amazon Mechanical Turk enabled us to recruit hundreds of people for every domain we considered. It allowed us to build more accurate human models and evaluate the methodology by means of an extensive study. Our experience in running experiments on Mechanical Turk demonstrated that almost all subjects considered our tasks seriously.

However for one of the experiments presented in this thesis we did not rely on Amazon's mechanical Turk but rather on real drivers, as explained in chapter 2. The drivers were provided advice from an agent regarding the settings of a climate control system in a real car.

Throughout this thesis, we used a general human model which, in most cases, did not take into account the differences among humans. Even in those cases in which we did consider different types of humans, there was no intent to model a specific person as the system keeps interacting with him. However, it is well known that people are different from each other and therefore personalization of the human model, and thus the way the system interacts with the human, is an important topic for future work. Another topic for future work are scenarios in which the user has other agents providing advice or information to them with different cost functions. In such cases, our agent may need to compete with other systems or agents. Furthermore future work may allow users to receive partial information from other sources and the ability to turn on or off advice or information received from our agent.

Bibliography

- [1] N. PELED, Y. GAL, AND S. KRAUS. **A Study of Computational and Human Strategies in Revelation Games.** In *Proc. of AAMAS*, 2011.
- [2] P. HOZ-WEISS, S. KRAUS, J. WILKENFELD, D. R. ANDERSEND, AND A. PATE. **Resolving crises through automated bilateral negotiations.** *AIJ*, **172(1)**:1–18, 2008.
- [3] A. AZARIA, Z. RABINOVICH, S. KRAUS, AND C. GOLDMAN. **Strategic Information Disclosure to People with Multiple Alternatives.** In *Proceedings of AAAI 2011*, 2011.
- [4] C. F. CAMERER. *Behavioral Game Theory. Experiments in Strategic Interaction*, chapter 2. Princeton University Press, 2003.
- [5] 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.
- [6] X. J. KUANG, R. A. WEBER, AND J. DANA. **How effective is advice from interested parties?** *J. of Economic Behavior and Organization*, **62(4)**:591–604, 2007.
- [7] G. PAOLACCI, J. CHANDLER, AND P. G. IPEIROTIS. **Running experiments on Amazon Mechanical Turk.** *Judgment and Decision Making*, **5(5)**, 2010.
- [8] B. J. FOGG. **Persuasive technology: using computers to change what we think and do.** *Ubiquity*, **2002(December)**:5, 2002.

BIBLIOGRAPHY

- [9] J. FROELICH, L. FINDLATER, AND J. LANDAY. **The design of eco-feedback technology.** In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2010.
- [10] P. MILGROM AND J. ROBERTS. **Relying on the Information of Interested Parties.** *Rand J. of Economics*, **17**:18–32, 1986.
- [11] V. CRAWFORD AND J. SOBEL. **Strategic Information Transmission.** *Econometrica*, **50**:1431–1451, 1982.
- [12] I. SHER. **Credibility and determinism in a game of persuasion.** *Games and Economic Behavior*, **71**(2):409 – 419, 2011.
- [13] J. GLAZER AND A. RUBINSTEIN. **A Study in the Pragmatics of Persuasion: A Game Theoretical Approach.** *Theoretical Economics*, **1**:395–410, 2006.
- [14] J. SOBEL. **Giving and receiving advice.** In *Econometric Society 10th World Congress*, 2010.
- [15] J. RENAULT, E. SOLAN, AND N. VIEILLE. **Dynamic Sender-Receiver Games**, 2011. Unpublished manuscript.
- [16] F. RICCI, L. ROKACH, B. SHAPIRA, AND P. KANTOR, editors. *Recommender Systems Handbook*. Springer, 2011.
- [17] J. B. SCHAFER, J. KONSTAN, AND J. RIEDL. **Recommender systems in e-commerce.** In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.
- [18] L. CHEN, F. HSU, M. CHEN, AND Y. HSU. **Developing recommender systems with the consideration of product profitability for sellers.** *Information Sciences*, **178**(4):1032–1048, 2008.
- [19] B. PATHAK, R. GARFINKEL, R. D. GOPAL, R. VENKATESAN, AND F. YIN. **Empirical analysis of the impact of recommender systems on sales.** *Journal of Management Information Systems*, **27**(2):159–188, 2010.
- [20] A. DAS, C. MATHIEU, AND D. RICKETTS. **Maximizing profit using recommender systems.** *ArXiv e-prints*, August 2009.

- [21] K. HOSANAGAR, R. KRISHNAN, AND L. MA. **Recomended for You: The Impact of Profit Incentives on the Relevance of Online Recommendations**. 2008.
- [22] G. SHANI, D. HECKERMAN, AND R. I. BRAFMAN. **An MDP-Based Recommender System**. *J. Mach. Learn. Res.*, **6**:1265–1295, 2005.
- [23] M. HIPPI, F. SCHAUB, F. KARGL, AND M. WEBER. **Interaction weaknesses of personal navigation devices**. In *Proc of AutomotiveUI*, 2010.
- [24] M. DUCKHAM AND L. KULIK. **"Simplest" paths: Automated route selection for navigation**. In *LNCSE*, **2825**, pages 169–185, 2003.
- [25] K. PARK, M. BELL, I. KAPARIAS, AND K. BOGENBERGER. **Learning user preferences of route choice behaviour for adaptive route guidance**. *IET Intelligent Transport Systems*, **1**(2):159–166, 2007.
- [26] R. J. HANOWSKI, S. C. KANTOWITZ, AND B. H. KANTOWITZ. **Driver acceptance of unreliable route guidance information**. In *Proc. of the Human Factors and Ergonomics Society*, pages 1062–1066, 1994.
- [27] D. ANTOS AND A. PFEFFER. **Using Reasoning Patterns to Help Humans Solve Complex Games**. In *IJCAI*, pages 33–39, 2009.
- [28] L. RAYO AND I. SEGAL. **Optimal Information Disclosure**. *Journal of Political Economy*, **118**(5):949–987, 2010.
- [29] S. K. HUI, P. S. FADER, AND E. T. BRADLOW. **Path data in marketing**. *Marketing Science*, **28**(2):320–335, 2009.
- [30] C. G. CHORUS, E. J. MOLIN, AND B. VAN WEE. **Travel information as an instrument to change car drivers travel choices**. *EJTIR*, **6**(4):335–364, 2006.
- [31] S. W. ROSENBERG, L. BOHAN, P. MCCAFFERTY, AND K. HARRIS. **The image and the vote: The effect of candidate presentation on voter preference**. *American Journal of Political Science*, pages 108–127, 1986.
- [32] S. SEUKEN, D. C. PARKES, E. HORVITZ, K. JAIN, M. CZERWINSKI, AND D. TAN. **Market user interface design**. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 898–915. ACM, 2012.

BIBLIOGRAPHY

- [33] M. FENSTER, I. ZUCKERMAN, AND S. KRAUS. **Guiding User Choice During Discussion by Silence, Examples and Justifications.** In *Proc. of ECAI*, pages 330–335, 2012.
- [34] A. AZARIA, S. KRAUS, C. V. GOLDMAN, AND O. TSIMHONI. **Advice Provision for Energy Saving in Automobile Climate Control Systems.** In *IAAI*, 2014.
- [35] A. AZARIA, A. ROSENFELD, S. KRAUS, C. V. GOLDMAN, AND O. TSIMHONI. **Advice Provision for Energy Saving in Automobile Climate Control Systems.** *AI Magazine*, 2015.
- [36] A. AZARIA, A. HASSIDIM, S. KRAUS, A. ESHKOL, O. WEINTRAUB, AND I. NETANELY. **Movie Recommender System for Profit Maximization.** In *RecSys*, 2013.
- [37] A. AZARIA, Z. RABINOVICH, S. KRAUS, C. V. GOLDMAN, AND O. TSIMHONI. **Giving Advice to People in Path Selection Problems.** In *AAMAS*, 2012.
- [38] A. AZARIA, Z. RABINOVICH, S. KRAUS, C. V. GOLDMAN, AND Y. GAL. **Strategic Advice Provision in Repeated Human-Agent Interactions.** In *AAAI*, 2012.
- [39] A. AZARIA, Y. GAL, S. KRAUS, AND C. GOLDMAN. **Strategic advice provision in repeated human-agent interactions.** *Autonomous Agents and Multi-Agent Systems*, pages 1–26, 2015.
- [40] A. AZARIA, Z. RABINOVICH, S. KRAUS, AND C. V. GOLDMAN. **Strategic Information Disclosure to People with Multiple Alternatives.** In *Proc. of AAAI*, 2011.
- [41] A. AZARIA, Z. RABINOVICH, S. KRAUS, AND C. V. GOLDMAN. **Strategic Information Disclosure to People with Multiple Alternatives.** 2014.
- [42] A. AZARIA, A. RICHARDSON, AND S. KRAUS. **An Agent for the Prospect Presentation Problem.** In *AAMAS*, 2014.
- [43] T. NGUYEN, R. YANG, A. AZARIA, S. KRAUS, AND M. TAMBE. **Analyzing the effectiveness of adversary modeling in security games.** In *AAAI*, 2013.
- [44] L. BACKSTROM AND J. LESKOVEC. **Supervised random walks: predicting and recommending links in social networks.** In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

- [45] G. LINDEN, B. SMITH, AND J. YORK. **Amazon. com recommendations: Item-to-item collaborative filtering.** *Internet Computing, IEEE*, 7(1):76–80, 2003.
- [46] Y. KOREN, R. BELL, AND C. VOLINSKY. **Matrix factorization techniques for recommender systems.** *Computer*, 42(8):30–37, 2009.
- [47] W. SHIH, S. KAUFMAN, AND D. SPINOLA. **Netflix.** *Harvard Business School Case*, 9:607–138, 2007.
- [48] D. READ. **Monetary incentives, what are they good for?** *Journal of Economic Methodology*, 12(2):265–276, 2005.
- [49] A. TVERSKY AND D. KAHNEMAN. **The Framing of Decisions and the Psychology of Choice.** *Science*, 211(4481):453–458, 1981.
- [50] A. TVERSKY AND D. KAHNEMAN. **Loss Aversion in Riskless Choice: A Reference-Dependent Model.** *The Quarterly J. of Economics*, 106(4):1039–1061, 1991.
- [51] H. RACHLIN AND L. GREEN. **Commitment, choice and self-control.** *J. Exp. Anal. Behav.*, 17:15–22, 1972.
- [52] L. DE ALFARO, T. HENZINGER, AND R. MAJUMDAR. **Discounting the Future in Systems Theory.** In J. BAETEN, J. LENSTRA, J. PARROW, AND G. WOEGINGER, editors, *Automata, Languages and Programming*, 2719 of *Lecture Notes in Computer Science*, pages 192–192. Springer Berlin / Heidelberg, 2003.
- [53] K.-F. RICHTER AND M. DUCKHAM. **Simplest instructions: Finding easy-to-describe routes for navigation.** In *Proc. of the Geographic Information Science*, 5266 of *LNCS*, pages 274–289, 2008.
- [54] H. H. HOCHMAIR AND V. KARLSSON. **Investigation of preference between the least-angle strategy and the initial segment strategy for route selection in unknown environments.** In *Spatial Cognition IV*, 3343 of *LNCS*, pages 79–97, 2005.
- [55] P. HART, N. NILSSON, AND B. RAPHAEL. **A Formal Basis for the Heuristic Determination of Minimum Cost Paths.** *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, February 1968.
- [56] AMAZON. **Mechanical Turk Services.** <http://www.mturk.com/>, 2013.

BIBLIOGRAPHY

- [57] J. VERMOREL AND M. MOHRI. **Multi-armed bandit algorithms and empirical evaluation.** In *Proc. of European Conference on Machine Learning*, pages 437–448. Springer, 2005.
- [58] N. GANS, G. KNOX, AND R. CROSON. **Simple Models of Discrete Choice and Their Performance in Bandit Experiments.** *M&SOM*, **9**(4):383–408, December 2007.
- [59] C. CHABRIS, D. LAIBSON, AND J. SCHULDT. **Intertemporal Choice.** *The New Palgrave Dictionary of Economics*, 2006.
- [60] A. DEATON AND C. PAXSON. **Intertemporal Choice and Inequality.** Work. P. 4328, NBER, April 1993.
- [61] P. A. HAILE, A. HORTASU, AND G. KOSENOK. **On the Empirical Content of Quantal Response Equilibrium.** *American Economic Review*, **98**(1):180–200, 2008.
- [62] P. AUER, N. CESA-BIANCHI, Y. FREUND, AND R. E. SCHAPIRE. **Gambling in a rigged casino: the adversarial multi-armed bandit problem.** In *Proc. of FOCS*, pages 322–331, 1995.
- [63] 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.
- [64] J. E. LISMAN AND M. A. P. IDIART. **Storage of 7 ± 2 Short-Term Memories in Oscillatory Subcycles.** *Science*, **267**:1512–1515, March 1995.
- [65] J. MARECKI, S. KOENIG, AND M. TAMBE. **A fast analytical algorithm for solving Markov decision processes with real-valued resources.** In *Proc. of IJCAI*, 2007.
- [66] Z. FENG, R. DEARDEN, N. MEULEAU, AND R. WASHINGTON. **Dynamic programming for structured continuous Markov decision problems.** In *Proc. of UAI*, 2004.
- [67] D. ORMONEIT AND S. SEN. **Kernel-based reinforcement learning.** *Machine Learning*, **49**(2):161–178, 2002.
- [68] N. METROPOLIS AND S. ULAM. **The monte carlo method.** *Journal of the American statistical association*, **44**(247):335–341, 1949.

- [69] W. K. HASTINGS. **Monte Carlo sampling methods using Markov chains and their applications.** *Biometrika*, **57**(1):97–109, 1970.
- [70] Y. GAL, S. KRAUS, M. GELFAND, H. KHASHAN, AND E. SALMON. **Negotiating with People across Cultures using an Adaptive Agent.** *ACM TIST*, **3**(1), 2012.
- [71] D. SILVER AND J. VENESS. **Monte-Carlo planning in large POMDPs.** In *Advances in Neural Information Processing Systems*, pages 2164–2172, 2010.
- [72] E. KAMENICA AND M. GENTZKOW. **Bayesian persuasion.** Technical report, University of Chicago, 2010. under review.
- [73] GOOGLE. **Google Maps.** <http://maps.google.com/>, 2013.
- [74] GOOGLE. **WAZE.** <http://www.waze.com/>, 2013.
- [75] D. SARNE, A. ELMALECH, B. J. GROSZ, AND M. GEVA. **Less is more: restructuring decisions to improve agent search.** In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 431–438. International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- [76] R. A. HORN AND C. R. JOHNSON. *Topics in Matrix Analysis.* Cambridge University Press, 1991.
- [77] M. J. OSBORNE AND A. RUBINSTEIN. *A course in Game Theory.* MIT Press, 1994.
- [78] D. LEE. **Best to go with what you know?** *Nature*, **441**(15):822–823, 2006.
- [79] N. D. DAW, J. P. O’DOHERTY, P. DAYAN, B. SEYMOUR, AND R. J. DOLAN. **Cortical substrates for exploratory decisions in humans.** *Nature*, **441**(15):876–879, 2006.
- [80] T. NGUYEN, R. YANG, A. AZARIA, S. KRAUS, AND M. TAMBE. **Analyzing the effectiveness of adversary modeling in security games.** In *Conf. on Artificial Intelligence (AAAI)*, 2013.
- [81] Y. GAL, B. GROSZ, S. KRAUS, A. PFEFFER, AND S. SHIEBER. **Agent Decision-Making in Open Mixed Networks.** *Artificial Intelligence*, **174**(18):1460–1480, 2010.
- [82] D. KAHNEMAN AND A. TVERSKY. **Prospect Theory: An Analysis of Decision under Risk.** *Econometrica*, **47**(2):263–291, 1979.

BIBLIOGRAPHY

- [83] D. READ, G. LOEWENSTEIN, AND M. RABIN. **Choice Bracketing.** *Journal of Risk and Uncertainty*, **19**(1):171–197, December 1999.
- [84] M. FRIEDMAN AND L. J. SAVAGE. **The expected-utility hypothesis and the measurability of utility.** *The Journal of Political Economy*, **60**(6):463–490, 1952.
- [85] A. TVERSKY AND D. KAHNEMAN. **Advances in prospect theory: Cumulative representation of uncertainty.** *Journal of Risk and Uncertainty*, **5**(4):297–323, October 1992.
- [86] D. KAHNEMAN. *Thinking, fast and slow*. Allen Lane, London, 2011.
- [87] G. HARRISON AND E. RUTSTROM. **Expected utility theory and prospect theory: one wedding and a decent funeral.** *Experimental Economics*, **12**:133–158.
- [88] J. RIESKAMP. **The probabilistic nature of preferential choice.** *Journal of Experimental Psychology*, **34**:1446–1465, 2008.
- [89] D. W. HARLESS AND C. F. CAMERER. **The Predictive Utility of Generalized Expected Utility Theories.** *Econometrica*, **62**:1251–1289, 1994.
- [90] A. AZARIA, S. KRAUS, AND A. RICHARDSON. **A System for Advice Provision in Multiple Prospect Selection Problems.** In *RecSys*, 2013.
- [91] L. BORGHANS, B. H. GOLSTEYN, J. J. HECKMAN, AND H. MEIJERS. **Gender Differences in Risk Aversion and Ambiguity Aversion.** Working Paper 14713, National Bureau of Economic Research, 2009.
- [92] E. BRANDSTÄTTER, G. GIGERENZER, AND R. HERTWIG. **The priority heuristic: making choices without trade-offs.** *Psychological review*, **113**(2):409, 2006.