ACTING RATIONALLY WITH INCOMPLETE UTILITY INFORMATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Urszula Chajewska
March 2002

Daphne Koller
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Yoav Shoham

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Eric Horvitz
Microsoft Research

Approved for the University Committee on Graduate Studies:

Abstract

Application of the decision-theoretic paradigm implicitly assumes knowledge of the utility values assigned to the relevant outcomes by the person affected by the decisions. However, in many cases, the task of acquiring such knowledge is infeasible due to the size of the outcome space and the complexity of the preference elicitation process. We argue that a person's utility values for the outcomes under consideration can be treated as we treat other domain attributes: as random variables with density functions over their possible values. A probabilistic framework is a natural way to express the uncertainty over utility values. It allows us to use a variety of well-known tools, such as expectation, value of information and conditioning.

We show that we can apply statistical density estimation techniques to learn such a probabilistic model from a database of partially elicited utility functions. The Bayesian learning framework we define for this problem also allows us to discover the number of statistically coherent subpopulations — groups of people with similar utility functions. Each subpopulation may have a different utility model with its own set of independence properties among utility attributes. We concentrate on a class of independence properties that correspond to the additive decomposition of the utility function. A decomposed utility function can be represented more compactly, making knowledge acquisition, inference, and learning much more efficient. Using our learning framework, we select the set of utility models that best matches the data. The factorization of the utilities in the learned model and the generalization obtained from density estimation allow us to provide a compact and robust representation of the utility functions in the population.

behavior is used to adjust the model to the user by Bayesian conditioning. The longer the interaction, the more information we have and the closer we get to knowing the user's utility function. At any point, we can use the probabilistic model to quantify the confidence in the accuracy of our utility estimates.

We present two applications of such customizable utility models. The first application, designed for a decision support system, directs the utility elicitation process by choosing questions most relevant for the current user and the current decision problem. The relevance of a utility elicitation question for the given problem can be measured by using its value of information. Having a probabilistic model of utilities, we can compute the best decision recommendation at any point relative to our current beliefs about the user's utilities. We can also compute the expected utility loss resulting from our recommendation, which allows us to stop the elicitation process as soon as the loss falls below a prespecified threshold.

The second application focuses on non-cooperative situations and considers the task of predicting the future decisions of an agent based on his past decisions. In such cases, we cannot rely on utility elicitation and must instead update our model with information gathered by observing the user's behavior. First, we show that the user's past decisions can be viewed as constraints on his utility function. We can condition our utility model on these constraints to obtain a posterior distribution. Using this distribution, we can predict the user's future behavior. We show that this capability is particularly useful in a two-player setting where a second agent is trying to optimize his payoff which also depends on the first agent's actions and utilities.

This work extends the traditional formulation of rational decision making to deal with the uncertainty over utility information. Such uncertainty can be expressed in probabilistic terms just as we express our uncertainty over domain events and thus can be integrated in a natural way into the traditional decision-theoretic framework. We believe that the tools we have developed can bring us closer to rational decision making in the real world.

Acknowledgments

My work at Stanford benefited from help from many people. I'd like to thank:

- my dissertation advisor, Daphne Koller, for supporting my research, providing feedback, for her relentless attention to detail, and especially for many technical insights, which helped improve my work;

- Miriam Kuppermann, for generously sharing with me her database of utility functions and her knowledge and experience in the prenatal diagnosis domain;

- my co-authors on various papers — Denise Draper, Lise Getoor, Joseph Halpern, Pedrito Maynard-Reid II, Joseph Norman, Daphne Koller, Dirk Ormoneit, Ronald Parr and Yuval Shahar — for many lessons learned in the course of our interactions;

- members of research and study groups I belonged to during my years at Stanford: Quail '97, for helping me acquire a solid overview of the field and find my place at Stanford at the same time; PANDA, for alerting me to the problem of utility modeling and inspiring me to do the work which resulted in this thesis; Dags, for exposing me to other people's work and for the use of the Phrog code; and TSG, for keeping me sane in the last stages of my thesis work;

- friends — in particular Eyal Amir, Patrick Doyle, Pedrito Maynard-Reid II, and Karl Pfleger — for proofreading my drafts, advice in technical and non-technical matters, for letting me vent frustrations, making me laugh, and supporting me in many different ways;

- most of all, my husband, Leszek, whose idea it was I should study Computer Science and without whom I would have never achieved what I did.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Rational Decision Making

Every agent, human or artificial, makes decisions in the course of his interactions with the world and other agents acting in the world. To make good decisions, we need knowledge about events that occur in the world, as they may affect the situation we find ourselves in. We also need knowledge about our own preferences capturing the extent to which we value different situations. In order to interact with other agents, we need to be able to predict their behavior, which also depends on world knowledge and preference information.

We are never certain about events happening in the world. Our uncertainty about possible events is best described within a probabilistic framework. Recent work on compact probabilistic models such as Bayesian networks (Pearl 1988) provides the basis for acquiring such models, whether by elicitation from experts, or by learning from databases of sample data (Jordan 1998).

The decision maker's preferences over outcomes (final states) are typically assumed to be known or easily acquired.[1] In fact, we usually assume not only that we know the preference information, but also that we have at our disposal the numerical values, or *utilities*, an agent assigns to every relevant outcome.

Given the probability distribution and the utility values, we can recommend the

[1]Except for some game-theoretic contexts; see Section 2.2.4.

of these outcomes given the decision. The decision with the highest expected utility is considered to be the best course of action. An agent following the principle of maximizing expected utility is commonly called a *rational* agent.

1.2 Uncertainty over Utilities

The use of the maximum expected utility principle assumes easy access to the decision maker's utilities for all relevant outcomes. In the real world, however, we rarely have full utility information at our disposal.

In contrast to the task of learning probabilistic models, the problem of acquiring utility functions is not well understood. In some sense, utility elicitation is innately harder; there are no experts to ask, and every person's utility function may be different. Thus, each individual's utility for each possible outcome must be elicited.

The process of utility elicitation is time consuming, cognitively difficult, and noisy. It is also prone to errors. People need to be trained before they can participate in a utility assessment procedure, and even after training, some people find answering elicitation questions difficult. Assessment is highly sensitive to an adequate description of outcomes. However, even with very precise descriptions, people tend to assign imaginary negative outcomes a much lower and imaginary positive outcomes a much higher utility value than the outcomes they have experienced (Schkade and Kahneman 1998; Lenert, Treadwell, and Schwartz 1999; Jansen, Stiggelbout, Wakker, Nooy, Noordijk, and Kievit 2000). Also, in long interviews, fatigue can influence the accuracy of the assessment.

There are many elicitation methods, and the fact that they produce very different utility values for the same outcomes when applied to the same person is well documented (Llewellyn-Thomas, Sutherland, Tibshirani, Ciampi, Till, and Boyd 1982; Fromberg and Kane 1989a; Fromberg and Kane 1989b; Read, Quinn, Berwick, Fineberg, and Weinstein 1984; O'Leary, Fairclough, Jankowski, and Weeks 1995).

In complex decision problems, the outcome space may become quite large. In the domain where we have focused our attention — medical problems — instances with dozens of possible outcomes are not uncommon. In problems of this size, full utility elicitation is infeasible — it would take many hours. Therefore, we can either elicit utilities for a subset of outcomes or make some simplifying assumptions about the structure of the utility function.

The use of structure (implied by conditional independence properties among random variables) has been shown to be crucial for modeling uncertainty over world events, simplifying both the representation and the associated knowledge acquisition process. Structure also exists in utilities. Utility functions can often be decomposed as a linear combination of *subutility functions*, each of which involves only a few of the relevant variables. Decomposable utility functions can be used to support more efficient inference (Tatman and Shachter 1990; Jensen, Jensen, and Dittmer 1994). In principle, as they require fewer parameters to be specified, they should also ease the knowledge acquisition process (Keeney and Raiffa 1976).

In practice, however, decomposable utility functions are rarely used (except in certain settings where everything easily reduces to a common basis, such as money). Why? Unfortunately, the task of discovering the structure of utility functions differs from the task of discovering structure of probability distributions. One might hope that there is some structure that holds for the entire population. Thus, we can elicit a database of utility functions, and try to find a common structure. Several researchers take this approach, usually trying to detect simple additive decompositions via a process of linear regression (Fromberg and Kane 1989a; Kuppermann, Shiboski, Feeny, Elkin, and Washington 1997). Unfortunately, such structure rarely seems to exist, so one typically resorts to explicit utility elicitation for the entire outcome space.

At the individual level, one could ask individuals about their decompositions. However, this approach is difficult to implement. Unlike probabilities, utilities cannot be marginalized. The utility of a specific instantiation of one state attribute does not functions.

Finally, in some situations, we may want to reason about the decision making process of other agents. These agents may prefer to keep information about their utility functions to themselves. In such cases, we cannot rely on utility elicitation at all.

Thus, in real world domains, complete and precise utility information is never available. The values for some parameters may be missing, for some others known only with a certain error margin. Some parameters may be only loosely constrained. How can we make rational decisions in these cases?

1.3 Utilities as Random Variables

Traditional decision theory accepts our uncertainty over events and allows us to represent it with a probability distribution. At the same time, it assumes that we have full utility information at our disposal. We will argue that the asymmetric treatment of events and utilities in the traditional framework is not justified. Just as we are not sure which situations we may find ourselves in, we have uncertainty over the values we (or other agents) assign to these situations. These two types of uncertainty can be dealt with in the same manner: by using the probabilistic framework.[2]

Treating utilities as random variables allows us to model our uncertainty in a principled way. In addition, it offers many benefits:

- It allows us to represent our uncertainty over utility function structure. Uncertainty over numerical utility values and uncertainty over the structure can be integrated in one model, which can be learned together from a database of utility data.

[2]The probabilistic framework is sometimes used to represent uncertainty over utility information in game theory; see (Fudenberg and Tirole 1991). In Artificial Intelligence the early work in this area includes (Jimison, Fagan, Shachter, and Shortliffe 1992; Poh and Horvitz 1993). We discuss it in detail in Section 9.2.

- Any prior knowledge can easily be incorporated into the model.
- Our model can represent utilities of different subpopulations with different utility function structures and encode their prevalence in the entire population.
- Our model can be easily updated based on new utility information.
- We can compute a point estimate of an agent's utility and express our confidence in that estimate.
- We can choose optimal actions with respect to our current utility information as represented in the model.
- We can apply various tools developed for the probabilistic framework, such as the value of information.
- Utility elicitation can be made more efficient, less sensitive to noise and more robust. It can also be customized for a specific patient and a specific decision problem.

Most importantly, the probabilistic framework allows us to make rational decisions under utility uncertainty without resorting to unreasonable simplifying assumptions, approximations, heuristics, or ad hoc solutions.

To motivate the problems of acting rationally in the presence of utility uncertainty we discuss briefly two possible settings — one-person decision problems and non-cooperative interactions — and illustrate each by introducing an application domain.

1.4 One-Person Decision Problems

Consider the task of building a medical expert system designed to advise patients on a best sequence of decisions. The probabilistic model is too complex for humans to a different utility function. Each will be willing to participate in a short interview. During that interview, we need to elicit enough utility information to be able to decide on the best course of action for a given patient.

Example: Prenatal Diagnosis

Such a decision support tool was developed for the domain of prenatal diagnosis by the PANDA project at Stanford Medical Informatics.[3] PANDA is a loose acronym for "prenatal testing decision analysis." PANDA uses knowledge gained from many studies and from practicing clinicians to advise patients on which prenatal diagnostic tests they should choose during their pregnancies.

The goal of prenatal diagnosis is to detect chromosomal abnormalities present in the fetus in the early stages of the pregnancy. Different abnormalities (diseases) have different prevalence rates and can affect the child with varying severity. There are several tests available to diagnose these diseases. These tests have different sensitivities and specificities (i.e., rates of false negatives and false positives), costs, and health risks.

In the real world, the decision about the choice of tests is rarely easy. The patient's risk for having a child with a serious disease depends on the mother's age, child's sex and race, and the family history. Some tests are not very accurate; others carry a significant risk of inducing miscarriages. Both a miscarriage (spontaneous abortion or SAB) and an elective termination of the pregnancy (induced abortion or IAB) can affect the woman's chances of conceiving again. The decision model for prenatal diagnosis is presented in Figure 1.1.

The outcomes in this domain have many attributes: the inconvenience and expense of fairly invasive testing, the possibility of test-induced miscarriage, knowledge

[3]See http://smi-web.stanford.edu/projects/panda/.

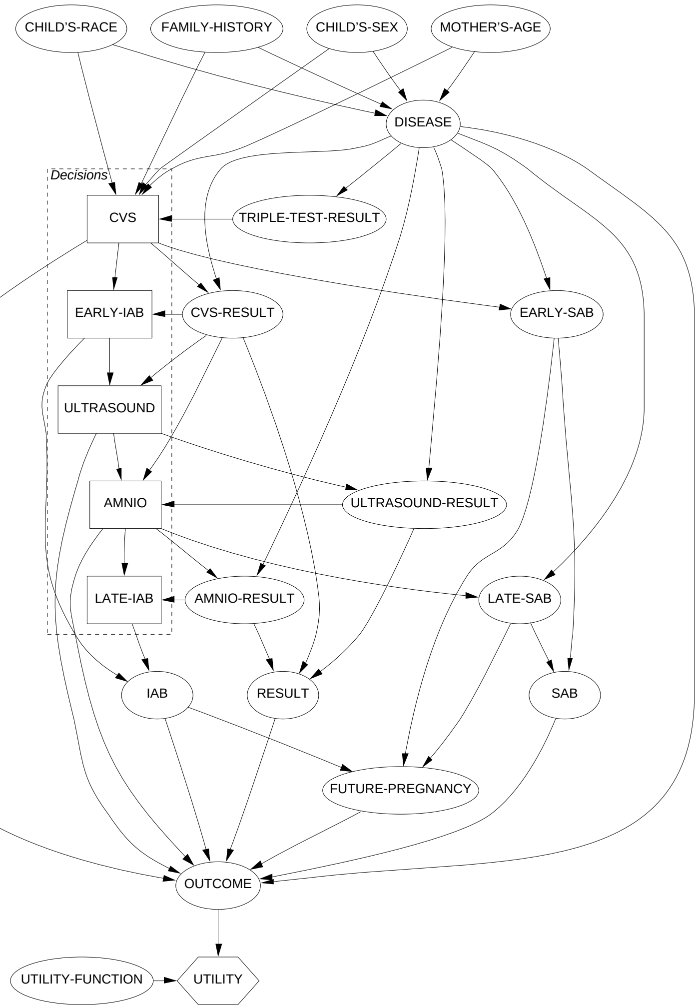


Figure 1.1: Decision model for prenatal diagnosis (courtesy of Joseph Norman, Stanford Medical Informatics)

lated and the utility of an outcome cannot be predicted from the utilities of the individual attributes. For example, consider the attributes "future pregnancy" and "miscarriage". While it is generally the case that a woman would like to conceive again following a miscarriage, and thus the attribute "future pregnancy" will be preferred to its negation, we can make no such assumption when a miscarriage has not occurred. The initial analysis of the model revealed the considerable influence of the utility function (especially the patient's attitude towards the risk of having a child with a serious disease and towards a miscarriage) on the optimal choice of actions.

The simplified model for the prenatal diagnosis domain we use in our work (shown in Figure 1.2) considers only one possible chromosomal abnormality which can be diagnosed during the pregnancy — Down's syndrome — and two tests commonly used to diagnose it: chorionic villus sampling (CVS) and amniocentesis (AMNIO).

We use five utility attributes:

- testing T (none, CVS or amniocentesis),
- fetus's status D (normal, affected by Down's syndrome),
- possible loss of pregnancy L (no loss, miscarriage, elective termination),
- knowledge of the fetus's status K (none, accurate, inaccurate),[4] and
- future successful pregnancy F (true, false).

The utility is a function of all of these variables.

Clearly, the number of outcomes is exponential in the number of attributes. Thus, the specification of the utility function in full can become expensive. In our model, there are 108 distinct outcomes; even after simplification and elimination of very unlikely outcomes, 66 outcomes remain. Utility elicitation, which in the best of cases is a long and tiring process, is extremely difficult for outcome spaces of this size.

[4]Knowledge is a deterministic function of fetus's Down's status and test result.

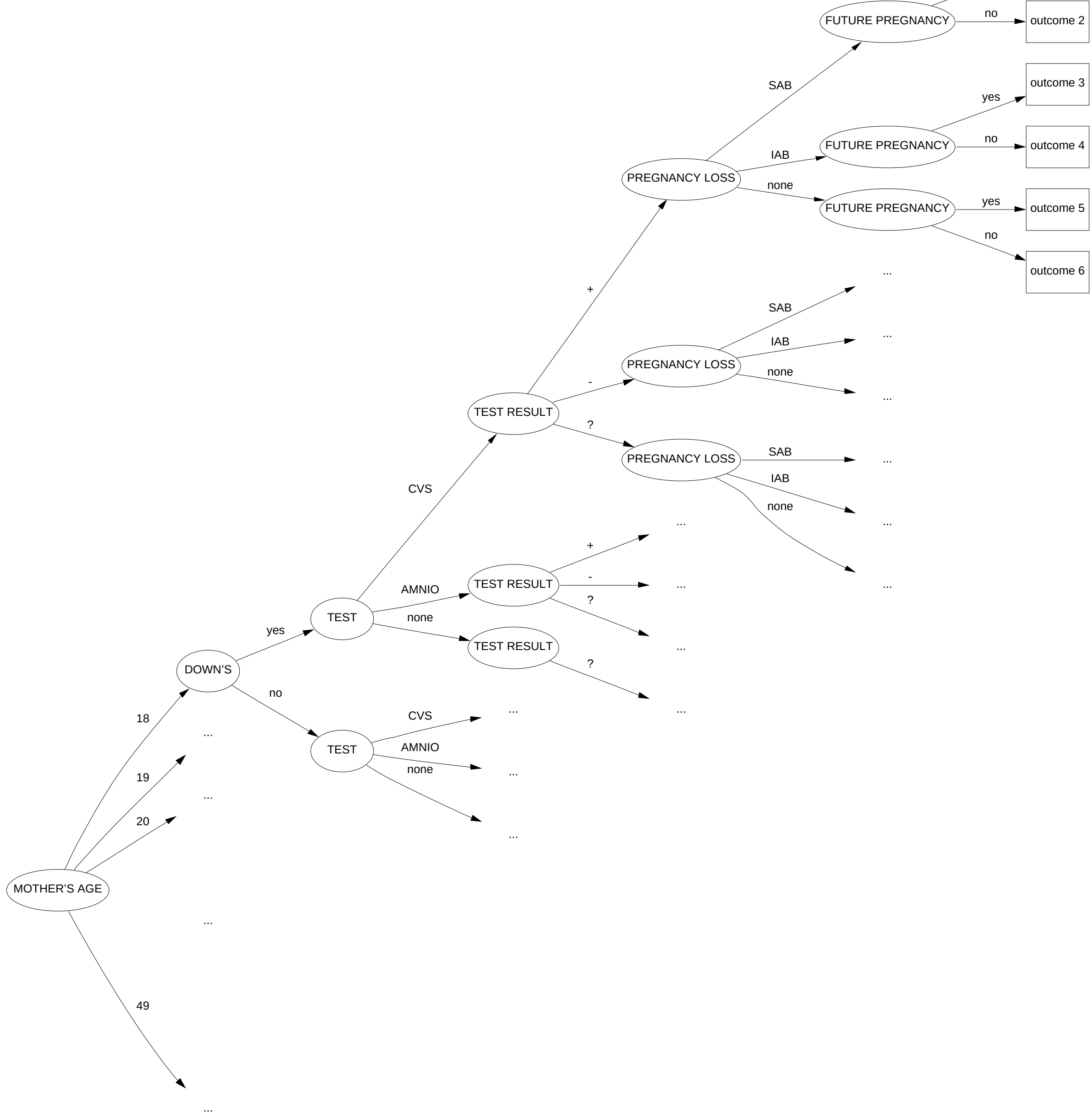


Figure 1.2: Prenatal diagnosis — simplified decision tree model

the first few questions, the probability of inconsistent answers rises sharply.

1.5 Non-cooperative Settings

Consider the problem of trying to predict the future actions of an agent A. There are many settings in which this capability is useful. In some settings, we might want to help the agent make good decisions. In other, more competitive settings, we might want to predict the agent's actions so as to better optimize our own actions.

In a direct approach to this task, we might simply try to learn a mapping from certain features of the current decision problem to actions. However, if the decision problems that A encounters are varied, it is not clear how we can extrapolate from one to the other. A better solution is to try to learn the basis for A's decisions, and use it to predict future decisions in a wide range of situations.

Our approach is based on the assumption that A is a rational decision maker, using decision theory to pick a course of action. According to decision theory, rational decision making amounts to the maximization of the expected utility of a sequence of decisions (Section 1.1). Thus, to predict A's actions, we need to know the probabilistic model of the domain which he uses and his utility function. In some situations, we can assume that the domain model constitutes common knowledge. In some others, a history of previous interactions with A will allow us to acquire the model he uses. However, the utility function remains a problem. In competitive situations, the knowledge of the opponent's utilities may constitute a strategic advantage. Thus, in such situations we cannot directly ask our opponent about his preferences.

Example: On-line Bookseller

As an example, consider a problem where we are trying to learn the utility function of some customer A of an online bookseller B. The customer A visits B's website from expects to enjoy. (The customer A is never completely sure himself whether he will enjoy a particular book. However, he can estimate how likely he is to enjoy it based on a review, experience with other books by the same author, etc.) By watching A's behavior, B also gains insight about A's utility function. A's overall utility typically depends on a combination of utility attributes, corresponding to such aspects as: the price he paid, his enjoyment of the book, etc.

The bookseller B can acquire information about the customer's utility function in one decision problem, and then apply it in another. However, it is rarely necessary (or possible) to learn A's exact utility function. The observer B typically needs to deal with A in the context of a particular interaction, and learning A's utility function is only useful inasmuch as it helps predict A's actions in that context.

Consider a single interaction between a bookseller and his online customer, illustrated in Figure 1.3. Here, the bookseller B is considering whether to offer a frequent customer A a discount on a newly published title. B would prefer to sell the book at the full price, but selling at a discount is better than not making a sale at all. B can furthermore alert his frequent customers by e-mail and notify them of the book's availability. However, this is only possible if the customer A previously signed up for this e-mail service. The customer's decision with respect to the e-mail service may influence B's decision about the discount offer.

We assume that both players can predict from A's previous purchases the chances that he will enjoy a particular title. This assumption, which may be unrealistic in some domains, is commonly made in game theory. Enjoyment of a book is taken here to be a decision made by nature. A cannot be sure of it before he actually reads the book. On the other hand, B may be collecting book ratings from his customers, which will allow him to predict A's enjoyment with high accuracy.

Clearly, B's utility depends on the actions taken by A, which in turn depend on A's unknown utility function.

The utility attributes we model for this problem are:

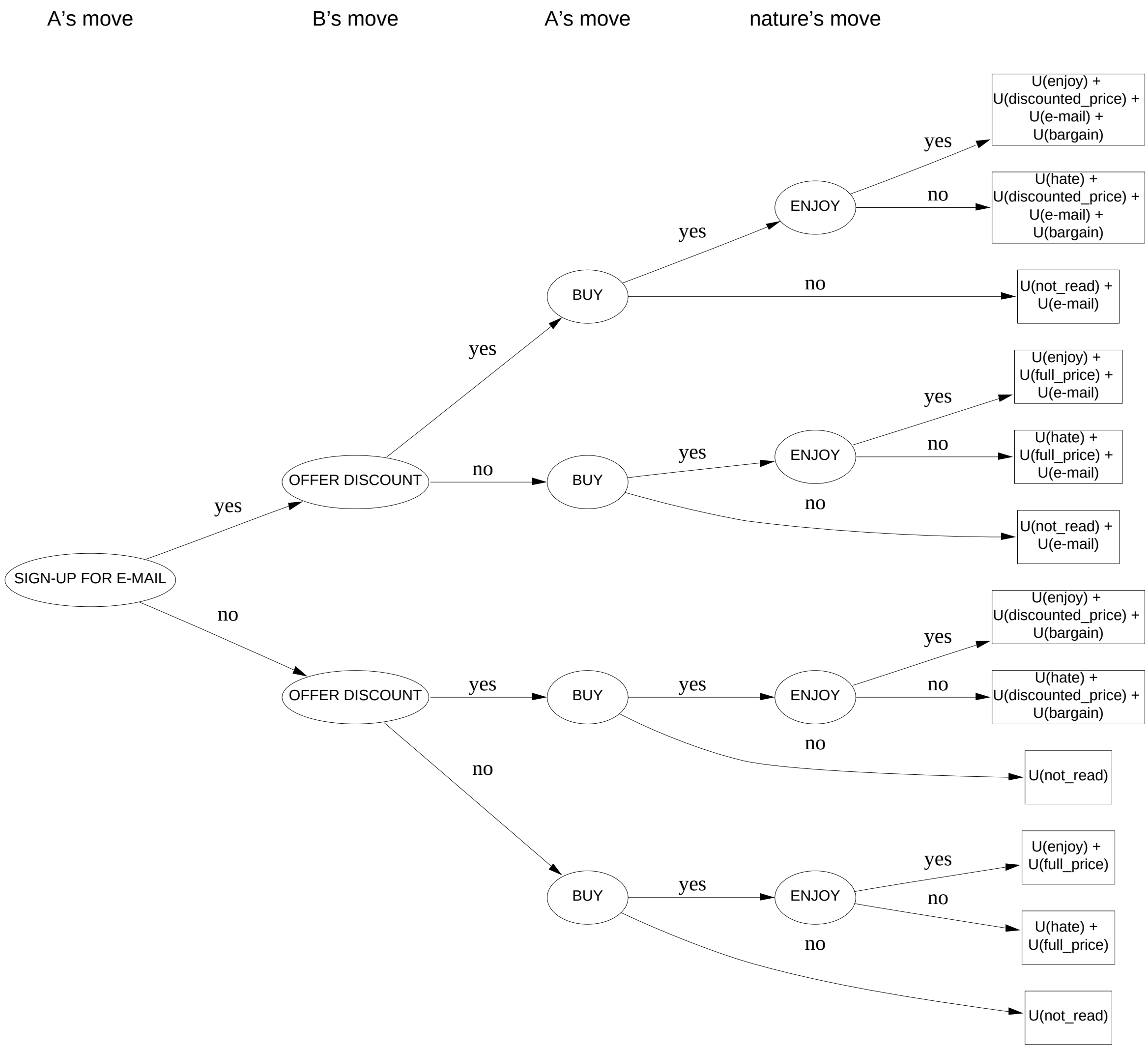


Figure 1.3: Bookseller example

- dislike of junk mail,
- satisfaction from obtaining a bargain.

Clearly, the more the bookseller B knows about the customer's utility function, the easier it is for him to predict A's future actions and maximize his gains by making optimal decisions.

1.6 Dissertation Overview

In Chapter 2 and Chapter 3, we review some background from decision theory and utility theory.

Chapter 4 develops the idea of utilities as random variables. It presents the probabilistic framework we use to encode our uncertainty over utility information and explains how we can exploit the independence properties among the utility attributes to simplify the model. It describes how we can use this new formulation to reason under utility uncertainty in a variety of settings.

In Chapter 5, we describe the process of learning a utility model. Our learning framework allows us to estimate the density from a database of utility functions collected from a population of users and discover the structure among utility variables at the same time. Chapter 6 shows how to adapt the distribution over utility functions in the population to a specific user by conditioning the prior density on new information whether collected in the process of utility elicitation or by observing the agent's behavior.

Chapter 7 presents an algorithm for adaptive utility elicitation. In this algorithm, we use the tools developed in Chapters 4 and 6 in the context of a medical decision support system. The utility elicitation process is customized for each patient. We choose elicitation questions according to their informational value and terminate which we have to reason about another agent's utility function. We cannot elicit any utility information, but we incorporate the information gathered by observing the other agent's behavior. Having a distribution over our opponent's utility function, we can compute the optimal sequence of actions to take.

We conclude in chapter 9 with discussion of related work and future directions.

Chapter 2

Decision Models

Decision theory is a framework for analyzing human decision making. On the one hand, it attempts to describe how people arrive at the decisions they make. On the other, it deals with choosing optimal decisions given a decision maker's preferences and his knowledge of the world. Thus, it has both descriptive and normative aspects which it tries to satisfy at the same time.

In this work, we are concerned principally with the normative side of decision theory. However, the two aspects are deeply interrelated. As we will see in Chapter 3, we cannot conduct a process of utility elicitation, which forces the user to choose actions in artificial decision scenarios, without acknowledging the fact that actual human decision making sometimes departs from the mathematical model.

2.1 One-Person Decision Problems

A *decision problem* for a single agent consists of several decisions, taken in sequence, often with some information revealed between one decision and the next. There are many possible models for this type of situation, including *influence diagrams* (Howard and Matheson 1984), *Markov decision processes* (MDPs) (Puterman 1994), and *decision trees* (Raiffa and Schlaifer 1961). For finite horizon situations, decision trees are a perhaps the most general representation from a semantic perspective. Any influence diagram can be expanded into a decision tree (Pearl 1988), and any MDP of the semantics of the decision task. Hence, we define decision problems in terms of decision trees.

2.1.1 Decision Trees

A decision tree is a rooted tree consisting of a finite set T of nodes and a binary relation $\lhd$ on T. We interpret $t_i \lhd t_j$ as meaning that the node t_i precedes the node t_j. We require that $\lhd$ be transitive and acyclic. For any node $t_i \in T$, the set $A(t_i) = \{t_j \in T : t_j \lhd t_i\}$ of predecessors of t_i is completely ordered by $\lhd$.

The root node t_1 is the node with no predecessors: $A(t_1) = \emptyset$. Similarly, the set L of *leaves* of the tree contains all the nodes with no successors $L = \{t_i \in T : S(t_i) = \emptyset\}$, where $S(t_i) = \{t_j \in T : t_i \lhd t_j\}$. $T - L$ is the set of *interior* nodes.

There are two types of interior nodes: decision nodes d and chance nodes y. A decision node indicates a choice to be made by the agent; its values are actions the agent can perform at the given point of the game. Each of the decision nodes d can correspond to a different set of actions; the tree arcs going out of d correspond to these actions. We represent the new information revealed to the agent in different tree branches and at different stages of the problem using a set of chance nodes. A chance node indicates a stochastic event. Each of the chance nodes y has a domain of two or more values. When the game reaches a chance node, its value is determined (by "nature") according to some distribution $P(y)$ and revealed to the agent. The outgoing arcs correspond to the node's values.

Note that the decision tree is not necessarily symmetric: in some branches of the tree the agent may be required to make different decisions than in others and may have different information at his disposal. The leaf nodes of the tree represent the *outcomes* of the decision problem. The *utility function*, U, assigns values to all possible outcomes. These values represent the decision maker's preferences (see Chapter 3).

The decision process corresponds to a top-down traversal of the tree. If the root selects an action from the set of actions associated with d. Even if two decision nodes have identical value sets, the agent can make different choices in each of them, thereby allowing his decision to depend on the values of their predecessors. The tree continues in this way, until at the very end, the final utility is determined according to the value function defined over the outcomes.

A *strategy* or *policy* is a sequence of decision rules, $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, one for each of the decision nodes. Each π_i is a mapping from a given decision node d to the set of values of that node, or possible decisions. The optimal strategy, π^*, maximizes the *value* of the decision problem, $\mathrm{EV}(\pi)$, which is taken to be the expected value of U under the strategy π.

2.1.2 Expectimax Algorithm

The value of the decision problem, and the optimal strategy, can be found by following the *expectimax* algorithm. The algorithm assigns the expected value EV_t to each node t in the decision tree. It proceeds from the leaves up, propagating expected values.

- At leaves, the values simply equal utility values for the outcomes:

$$\mathrm{EV}_t = U(t).$$

- At chance nodes, we take expectation over the values of the children $Ch(t)$ of the given node t:

$$\mathrm{EV}_t = \sum_{t' \in Ch(t)} P(t')\mathrm{EV}_{t'}.$$

- At decision nodes d, we determine the optimal action $\pi^*(d)$ by optimizing the expected value:

$$\pi^*(d) = \operatorname*{argmax}_{t' \in Ch(d)} \mathrm{EV}_{t'}.$$

The expected value of the root node EV_{t_1} is the expected value of the decision problem.

2.1.3 Symmetric Decision Problems

Some decision problems exhibit a high degree of regularity: all of the nodes at each stage of the game are of the same type (decision nodes or chance nodes) and have the same set of values. This results in a symmetric tree. When we convert a decision problem represented as an influence diagram to the decision tree representation, the resulting tree is always symmetric. The sets of identical nodes at every stage of the decision tree correspond to decision variables and chance variables in the influence diagram.

We can order the decision variables $D_1, \ldots, D_k$ according to their temporal structure. Similarly, the chance variables can be partitioned into mutually exclusive and exhaustive sets $\mathbf{Y}_1, \ldots, \mathbf{Y}_{k+1}$, where $\mathbf{Y}_i$ are the variables that are revealed to the decision maker prior to D_i. The value of a symmetric decision problem can be described using the following *sum-max-sum* rule (Jensen, Jensen, and Dittmer 1994):

$$\begin{aligned}\mathrm{EV}(\pi^*) = {} & \sum_{\mathbf{Y}_1} P(\mathbf{Y}_1) \max_{D_1} \sum_{\mathbf{Y}_2} P(\mathbf{Y}_2 \mid \mathbf{Y}_1, D_1) \ldots \\ & \cdot \sum_{\mathbf{Y}_k} P(\mathbf{Y}_k \mid \mathbf{Y}_1, D_1, \ldots, \mathbf{Y}_{k-1}, D_{k-1}) \\ & \cdot \max_{D_k} \sum_{\mathbf{Y}_{k+1}} P(\mathbf{Y}_{k+1} \mid \mathbf{Y}_1, D_1, \ldots, \mathbf{Y}_k, D_k) \cdot U(\mathbf{Y}_1, \ldots, \mathbf{Y}_{k+1}, D_1, \ldots, D_{k+1})\end{aligned}$$

The value of a symmetric decision problem can be computed by using the expectimax algorithm described above. More efficient algorithms exist for the influence diagram representation (Jensen, Jensen, and Dittmer 1994), where we can take advantage of the regularity of the decision problem.

Game theory aims to model situations in which two or more rational decision makers interact. The essentials of the theory were developed by von Neumann and Morgenstern (1947) in their famous book *The Theory of Games and Economic Behavior*, which also initiated modern utility theory (see Section 3.1.1).

The basic framework of game theory assumes an interaction between several decision makers. Each decision maker participating in this interaction (called a *player* in game theory literature) has a set of decisions available to him in different circumstances. Decisions may be made by all players simultaneously or in sequence, usually alternating between players. The set (in case of simultaneous moves) or the sequence of decisions determines the final state or outcome. Each player has his own preferences defined over the set of outcomes which is usually represented as a *payoff function* which assigns numbers to outcomes in some way corresponding to the preference ordering. In games with an element of chance, the payoff function is simply the utility function of the player (see Chapter 3).

In the basic formulation of a game, it is assumed that each player is fully aware of the rules of the game and his own as well as others' payoff functions. Similarly to single player decision problems, each player's objective is to the maximize his payoff function. However, this goal, straightforward in the single-player case, is complicated here by the players' anticipation of each others' moves.

A complete review of the field of game theory is beyond the scope of this dissertation. It can be found, e.g., in (Fudenberg and Tirole 1991). In the remainder of this section, we briefly review one of the classic representations of a game — the extensive form — which generalizes the decision tree representation we used for single player decision problems in Section 2.1.1. We also mention games with incomplete information, a class of games in which some information (e.g., about another player's payoff function) is unavailable. A special case of a game with incomplete information will be analyzed in Chapter 8.

Section 2.1.1.

As before, the interior nodes of the tree represent either decision nodes or chance nodes. Decision nodes in a game tree are annotated with the name or number of the player who makes a decision at the given node, $d_1^{p_1}, \ldots, d_n^{p_n}$, where p_i denotes the player making a decision at node d_i. As in single-agent decision trees, some new information can be revealed to some or all the players at different stages of the problem using a set of chance nodes. Chance nodes are often presented as decision nodes of a special player, nature. Nature does not have its own utility function and chooses its actions according to some pre-specified probability distribution known to some or all players. The leaves of the tree, $l \in L$, are annotated with the payoff values for all players, $U^p(l)$.

A player may or may not be aware of the moves made by other players or nature earlier in the game. This situation is captured using the concept of an *information set*. Two decision nodes d_i and d_j belong to the same information set I if the player who is supposed to make a decision in either of them cannot distinguish between the two. All nodes in the information set must belong to the same player and the sets of actions available to him at each of these nodes must be identical. We will refer to the set of actions available to player p at each of the nodes in the information set I^p as $a(I^p)$.

A game is said to have *perfect information* if each information set consists of a single decision node. Chess is an example of a perfect information game. Most card games (games in which the players do not reveal their cards to their opponents) are games with imperfect information.

We restrict our attention to games with *perfect recall*, in which each player retains all the information acquired at all the stages of the game and remembers his own prior moves as well as the moves of other players he was able to observe.

Similarly to decision trees, the act of playing a game corresponds to a top-down traversal of the tree. At each chance node y nature makes the choice about the value of other players and of nature that they were able to observe at earlier stages of the game. One path through the tree is called a *play* or a *game trajectory*.

A *pure strategy* for player p is a sequence of decision rules, $\pi^p = (\pi_1^p, \pi_2^p, \ldots, \pi_{n_p}^p)$, where n_p is the number of information sets I^p for player p and each π_i^p is a mapping from I_i^p to $a(I_i^p)$. A notion of a player's strategy can be extended to include *mixed strategies*: stochastic choices between pure strategies according to some fixed distribution.

2.2.2 Nash Equilibria

How can we compute the best strategy for a given player? If we knew in advance the strategies adopted by all the other players, it would be simple: we would be able to use our expectimax algorithm from Section 2.1.2. However, we usually do not have any information about other players' plans. Thus, we need to guess which actions they are going to take and adjust our strategy accordingly. If every player guesses right and behaves optimally with respect to his guess, the strategies will be at an equilibrium — no player will have any reason to change his behavior.

A solution to a game is such an equilibrium, called after its inventor a *Nash equilibrium*.

Definition 2.2.1: A *Nash equilibrium* for an m-player game is a set of strategies $\{\pi^1, \ldots, \pi^m\}$ such that no player i can achieve a higher expected payoff by using any strategy other than π^i given that every other player j adheres to π^j. ∎

A famous theorem, proven by Nash and bearing his name (Nash 1951), states that every game has a Nash equilibrium. However, it is not necessarily a pure-strategy equilibrium. Not every game has a pure strategy Nash equilibrium; if it exists it is not necessarily unique.

Nash equilibrium is not necessarily the most desirable state of the game. In a famous 2-player game called the Prisoner's Dilemma, the unique Nash equilibrium is an

Extensive form games with perfect information can be solved by a process of *backward induction*, which is a generalization of the expectimax algorithm described in Section 2.1.2. Recall that in perfect information games every information set is a singleton, so every player knows at every stage of the game which decision node he finds himself at.

The algorithm assigns the expected payoff value for every player p to every node t in the tree, EV_t^p. It proceeds from the leaves up, propagating expected payoff values.

- At leaves, the values simply equal payoff values for the outcomes:

$$\mathrm{EV}_t^p = U^p(t).$$

- At nature's nodes, we take expectation over the values of the children $Ch(t)$ of the given node t, separately for each player:

$$\mathrm{EV}_t^p = \sum_{t' \in Ch(t)} P(t')\mathrm{EV}_{t'}^p.$$

- At decision nodes d^i for player i, the algorithm determines the optimal action $\pi^*(d^i)$ by optimizing the expected payoff for player i:

$$\pi^*(d^i) = \arg\max_{t' \in Ch(d^i)} \mathrm{EV}_{t'}^i.$$

  The expected payoff values for $t = d^i$ are then set:

$$\mathrm{EV}_t^p = \mathrm{EV}_{\pi^*(d^i)}^p$$

  for all players p.

The set of optimal actions computed by backward induction defines a pure strategy

2.2.4 Games with Imperfect or Incomplete Information

Backward induction cannot be applied to games with imperfect information — the optimal action at one node in the information set is not necessarily the same as the optimal action at another node in the same set. We do not have any principled way to choose between them. Recall that games with imperfect information may not have a pure-strategy Nash equilibrium. They are guaranteed, however, to have a mixed-strategy one. Such equilibria can be computed by algorithms more complex than backward induction; see a comprehensive survey by McKelvey and McLennan (1996).

When some players do not know the payoff of the others, the game is said to have *incomplete information*. When there is a small number of possible payoff functions and we can assume that the distribution over these possibilities is commonly known, we can convert the incomplete information game to an imperfect information game in which the first move is made by nature and concerns the payoff functions.

Chapter 3

Utility Theory

Utility theory originated from the study of people's economic behavior. The terminology used reflects this origin: we talk about people's preferences over "goods" and use examples in which we gain or lose "assets" or simply money. Utility analysis, however, is not limited to monetary or economic resources and can be applied equally well to decision problems whose consequences are non-monetary, such as one's health state or enjoyment. We will use the term *outcome* for all types of decision consequences.

3.1 Basic Framework

3.1.1 Preferences and Lotteries

Modern utility theory was developed by von Neumann and Morgenstern (1947). The axioms of the theory are formulated in terms of preference and indifference relations defined on a set of outcomes **O**.

$o_i \succ o_j$ — outcome o_i is preferred to outcome o_j

$o_i \sim o_j$ — the agent is indifferent between o_i and o_j

$o_i \succsim o_j$ — the agent prefers o_i to o_j or is indifferent between them

The set of outcomes **O** can be enriched to include *lotteries*, or probability mixtures. A lottery ticket represents a chance mechanism which yields the prizes $o_1, o_2, \ldots, o_n$

and Morgenstern's original formulation (for a discussion and comparison see, e.g., (Luce and Raiffa 1957) or (Fishburn 1982)). We present the formulation used in (Coombs, Dawes, and Tversky 1970), which closely follows the original one.

Given the primitives $\succsim$ and $\mathbf{O}$, the following axioms are assumed to hold for all outcomes o_i, o_j, o_k in $\mathbf{O}$ and for all probabilities p, q that are different from 0 and 1.

A1 Closure: $[o_i, p, o_j]$ is in $\mathbf{O}$.

A2 $\succsim$ is a weak ordering of $\mathbf{O}$. That is, the following conditions are satisfied:

> **A2.1** Reflexivity: $o_i \succsim o_i$
>
> **A2.2** Connectivity: $(o_i \succsim o_j)$ or $(o_j \succsim o_i)$ or both
>
> **A2.3** Transitivity: $(o_i \succsim o_j) \wedge (o_j \succsim o_k) \Rightarrow (o_i \succsim o_k)$

A3 Reducibility: $[[o_i, p, o_j], q, o_j] \sim [o_i, pq, o_j]$

A4 Substitutability: $o_i \sim o_j \Rightarrow [o_i, p, o_k] \sim [o_j, p, o_k]$

A5 Monotonicity: $o_i \succ o_j \Rightarrow o_i \succ [o_i, p, o_j] \succ o_j$

A6 Continuity: $o_i \succ o_j \succ o_k \Rightarrow \exists p\, [o_i, p, o_k] \sim o_j$

The first axiom asserts that if o_i and o_j are available alternatives, so are all lotteries of the form $[o_i, p, o_j]$ that can be formed with o_i and o_j as outcomes. It is assumed implicitly that $[o_i, p, o_j] = [o_j, 1-p, o_i]$.

The second axiom is very important. The requirements of reflexivity and connectivity are very intuitive: It is easy to accept that any lottery or outcome is equivalent to itself and that the decision maker should know what he wants. Transitivity is a little more difficult. Although very compelling on normative grounds, it is the most frequently violated axiom in practice. Some researchers suspect that these "mistakes" is forced to make choices between inherently incomparable alternatives. The idea is that each pairwise comparison invokes a preference response on a different "attribute" scale and that, although each scale itself may be transitive, their combination need not be. Here, we will assume that the transitive preference ordering is at least a close approximation to reality even if it does not perfectly represent it.

Axiom A3 states that the preferences depend only on final outcomes, not the process in which they are obtained. It implies that a person does not derive any additional pleasure (or displeasure) from suspense or participation in the game.

Axiom A4 simply states that two equally preferred outcomes can be substituted for one another in any lottery. It is reminiscent of the assumption of the independence of irrelevant alternatives (Luce and Raiffa 1957).

The fifth axiom asserts that if o_i is preferred to o_j, then it must be preferred to any probability mixture of o_i and o_j, which, in turn, must be preferred to o_j. It does not seem objectionable. It was challenged, however, by citing the examples of a risky behavior such as Russian roulette or mountain climbing. People who choose to engage in such behaviors seem to prefer a probability mixture of "life" and "death" to "life", even though they prefer "life" to "death." The dispute is often resolved by revising the outcome descriptions. The outcome "life" can mean "life without climbing" or "life plus the thrill of a climb," which allows us to model the situation without violating the axioms.

The last axiom excludes the possibility that one alternative is "infinitely better" than another one, in the sense that any probability mixture involving the former is preferable to the latter. It captures the relationship between probabilities and preferences and the form in which they compensate for each other. It turns out to be an important justification for one of the methods of utility elicitation (see Section 3.4).

Theorem 3.1.1: *(von Neumann and Morgenstern 1947) If an agent's preferences obey axioms A1–A6, then there exists a real-valued function* U *defined on* $\mathbf{O}$ *such that*

$$o_i \succsim o_j \Leftrightarrow U(o_i) \geq U(o_j) \tag{3.1}$$

$$U([o_i, p, o_j]) = pU(o_i) + (1-p)U(o_j) \tag{3.2}$$

Furthermore, if U' *is any other function satisfying 3.1 and 3.2, then there exist numbers* $a > 0$ *and* b *such that* $U'(o_i) = aU(o_i) + b$*.*

The theorem guarantees that if axioms A1–A6 hold, there exists a utility function that preserves the preference order of the agent. The utility scale is uniquely determined except for the origin and a unit of measurement.

Equation (3.2) is sometimes referred to as the *expected utility principle* since it asserts that the utility of a lottery is equal to the expected utility of its component outcomes.

Given that an agent's preferences can be represented by a utility function, when the agent makes decisions according to his preferences, he behaves as if he were maximizing his expected utility. (The agent may or may not be aware of making choices in this manner.) The principle of maximizing expected utility has become synonymous with making *rational* decisions.

The fact that a utility function is defined only up to a positive linear (affine) transformation leads to problems for interpersonal comparisons of utility functions. The simplest solution is to establish a fixed range by assigning specific values to the least and most preferred outcomes in the decision problem. That creates its own problems, however: the least preferred outcome is not necessarily equally valued by everyone. A better solution is to find two *anchor* outcomes, $o_\perp$ and $o_\top$, which represent states whose values are supposedly independent of one's life situation. The most commonly used anchor states are the death of the decision maker for $o_\perp$ and perfect happiness (perfect health in medical domains) for $o_\top$. These states are assigned values

3.1.3 From Utilities to Preferences

Decision theory traditionally takes the preference relation as primitive and the utility function as implied. However, in practice we also reverse this relationship.

A utility function U induces a *preference ordering* $\succsim_U$ on lotteries (probability distributions) over $\mathbf{O}$ as follows:

$$p_1 \succsim_U p_2 \quad \text{iff} \quad \sum_{o \in \mathbf{O}} p_1(o)U(o) \geq \sum_{o \in \mathbf{O}} p_2(o)U(o),$$

where p_1 and p_2 are two distributions over $\mathbf{O}$. In other words, we prefer lotteries with higher expected utility.

3.2 Multi-attribute Utility Theory

Multi-attribute utility theory deals with utility functions defined over outcomes represented as assignments of values to utility attributes. The main focus of the theory is to identify regularities in an agent's preferences. Such regularities result from *independence properties* among utility attributes. The existence of a structure in an agent's preferences allows us to decompose his utility function into a sum or a product of subutility functions, each defined over a subset of utility attributes. The theorems linking various independence properties to different forms of utility function decomposition are called *representation theorems*.

We will review the most important results of multi-attribute theory here; for a thorough survey of the field the reader is referred to (Keeney and Raiffa 1976).

3.2.1 Utility Attributes

The naive representation of a utility function is a vector of real numbers, ascribing a utility to each possible outcome. This representation is quite reasonable in domains

Let $\mathbf{V} = \{V_1, V_2, \ldots, V_m\}$ be a fixed set of m attributes. Each attribute V_i has a domain Dom_{V_i} of two or more elements. The set of outcomes (states), $\mathbf{O}$, consists of the set of points in the product space $\prod_{i=1}^{m} \mathrm{Dom}_{V_i}$. Each $o \in \mathbf{O}$ is thus a vector of m values, one value for every attribute: $\langle v_1, v_2, \ldots, v_m \rangle$. Clearly, the size of $\mathbf{O}$ is exponential in m.

Example 3.2.1: Our prenatal diagnosis domain (Section 1.4) is a good example of the vector representation. We have 5 utility attributes:

- pregnancy loss (domain: {no loss, miscarriage, elective termination}),
- fetus' Down's status (domain: {normal, Down's}),
- mother's knowledge (domain: {none, accurate, inaccurate}),
- future pregnancy (domain: {yes, no}), and
- type of test (domain: {none, CVS, amnio}).

An outcome is an assignment of values to all the attributes. For example, ⟨no loss, normal, none, yes, none⟩ is one of the possible outcomes. It represents the situation in which the fetus is not affected by Down's syndrome, the patient decides not to take any tests (as a consequence, she is unaware of the Down status of the fetus until the end of the pregnancy), the pregnancy results in normal birth and there is a future pregnancy. Another outcome, ⟨miscarriage, normal, accurate, no, CVS⟩ represents the situation of a patient deciding to undergo the chorionic villus sampling test. The test result correctly asserts that fetus is not affected by the Down's syndrome. However, a miscarriage occurs as a side effect of the procedure and there is no future pregnancy.

Since three of the attributes are ternary and two binary, the total number of outcomes is $3 \times 2 \times 3 \times 2 \times 3 = 108$. As can be easily seen, not all outcomes have a positive probability. For example, in the absence of testing, knowledge about the

If the decision problem is represented as a symmetric decision tree (Section 2.1.3), we often use decision and chance variables as utility attributes.

3.2.2 Subutility Functions

If $\mathbf{X} \subseteq \mathbf{V}$ then $f(\mathbf{X})$ stands for some real valued function all of whose arguments are in $\mathbf{X}$, i.e.,

$$f(\mathbf{X}) : \prod_{V_i \in \mathbf{X}} \mathrm{Dom}_{V_i} \to I\!R.$$

We will call such functions defined over subsets of $\mathbf{V}$ *subutility functions*.

Subutility functions are different from utility functions, since they are not defined over complete outcomes. They do not always have an intuitive meaning. Unlike probabilities, utilities cannot be marginalized. It is hard to talk about one's utility function over a subset of attributes without making some assumptions about the values of other attributes. For example, consider the attribute "mother's knowledge" in our prenatal diagnosis domain. It is easy to develop an intuition behind marginal probability over "mother's knowledge". There is not, however, an analogous "marginal value" concept we could invoke. The value of knowledge could be different in different contexts (i.e., with the remaining attributes being set to different values) and it is not clear how we could aggregate these different functions or what the aggregate would mean.

Note that subutility functions are different from *conditional utility functions* which are defined over some subset of attributes $\mathbf{X}$ for fixed values $\mathbf{y}$ of the remaining attributes $\mathbf{Y}$ and denoted $U(\mathbf{X}, \mathbf{y})$ or $U_{\mathbf{y}}(\mathbf{X})$.

---

[1] We can reduce the number of outcomes further by restricting the space of strategies allowed by our model. For example, 18 outcomes involve an elective termination of the pregnancy following a negative (i.e., indicating the absence of Down's syndrome) test result. If we decide not to model this unlikely scenario, we are left with only 66 outcomes.

The set of attributes $\mathbf{X}$ is *preferentially independent* of its complement $\mathbf{Y}$ if the agent's conditional preferences over $\mathbf{X}$ given $\mathbf{y}$ (some value of $\mathbf{Y}$) do not depend on the particular value $\mathbf{y}$. In other words, if $\mathbf{x}$ is preferred to $\mathbf{x}'$ in the presence of some $\mathbf{y}$, it must be also preferred to $\mathbf{x}'$ in the presence of any other value of $\mathbf{Y}$. More formally,

Definition 3.2.2: (adapted from (Keeney and Raiffa 1976), page 109) The set of attributes $\mathbf{X}$ is *preferentially independent* of $\mathbf{Y} = \mathbf{V} - \mathbf{X}$ if and only if for all $\mathbf{y}, \mathbf{y}' \in \prod_{V_i \in \mathbf{Y}} \mathrm{Dom}_{V_i}$,

$$\langle \mathbf{x}, \mathbf{y} \rangle \succsim \langle \mathbf{x}', \mathbf{y} \rangle \Rightarrow \langle \mathbf{x}, \mathbf{y}' \rangle \succsim \langle \mathbf{x}', \mathbf{y}' \rangle$$

for all $\mathbf{x}, \mathbf{x}' \in \prod_{V_i \in \mathbf{X}} \mathrm{Dom}_{V_i}$. ∎

Preferential independence is not a symmetric relation — even if a set of attributes $\mathbf{X}$ is preferentially independent of its complement $\mathbf{Y}$, $\mathbf{Y}$ may not be preferentially independent of $\mathbf{X}$.

3.2.4 Utility Independence

Preference independence is only defined for decision making with certainty. The corresponding property for the uncertainty case is *utility independence.*

To define utility independence, we need to extend our definition of preference orderings to lotteries over a restricted set of outcomes, i.e., to cases where the values of some of the attributes are fixed:

Definition 3.2.3: Let $\mathbf{X} \subset \mathbf{V}$, $\mathbf{Y} = \mathbf{V} - \mathbf{X}$. Let $\mathbf{y}$ be any particular element of $\prod_{V_i \in \mathbf{Y}} \mathrm{Dom}_{V_i}$. Every probability distribution p over $\prod_{V_i \in \mathbf{X}} \mathrm{Dom}_{V_i}$ corresponds to a distribution p^* on $\mathbf{O}$ such that $p^*(\mathbf{y}) = 1$ and $p^*(\mathbf{X}|\mathbf{y}) = p(\mathbf{X})$. We define the *conditional preference over* $\mathbf{X}$ *given* $\mathbf{y}$, $\succsim_{\mathbf{y}}$ to be the preference ordering such that

$$p \succsim_{\mathbf{y}} q \quad \text{iff} \quad p^* \succsim q^*,$$

lotteries on $\mathbf{X}$ do not depend on the particular value given to $\mathbf{V} - \mathbf{X}$. That is,

$$(\forall \mathbf{y}, \mathbf{y}' \in \prod_{V_i \in \mathbf{V}-\mathbf{X}} \mathrm{Dom}_{V_i}) \quad p \succsim_{\mathbf{y}} q \quad \text{iff} \quad p \succsim_{\mathbf{y}'} q,$$

where p and q are any two distributions over $\prod_{V_i \in \mathbf{X}} \mathrm{Dom}_{V_i}$. In this case, we write $\mathrm{UI}(\mathbf{X}, \mathbf{V} - \mathbf{X})$. ∎

Utility independence essentially asserts that the preferences on lotteries involving outcomes with different values of a given attribute (or set of attributes) do not depend on the values to which the remaining attributes are set.

Utility independence, like preference independence, is not a symmetric relation. In a utility function with two attributes, all cases are possible: neither attribute is independent of the other, one of the attributes is independent of the other but not vice versa, or each attribute is independent of the other.

Note also that utility independence is only defined for a set of attributes and its complement. That is, if a utility function is defined over a product space of three attributes, we cannot say that one attribute is independent of another without mentioning the third. Unlike probabilities, utility functions cannot be marginalized.

In our prenatal diagnosis example (Section 1.4), the attribute "Testing" is (for most people) utility independent of the rest. One typically dislikes the invasive testing regardless of one's preferences over pregnancy outcomes. On the other hand, the attribute "Future Pregnancy" is clearly utility dependent on the values of the remaining attributes: one's preferences towards the future pregnancy may well be influenced by the outcome of the current one.

Utility independence gives us several decomposition theorems. First, consider the simplest case when a subset of attributes is utility independent of the rest:

Proposition 3.2.5: *(adapted from (Keeney and Raiffa 1976), page 226)* $\mathbf{X}$ *is utility independent of its complement in a preference structure* $\succsim$ *if and only if* $\succsim$ *corresponds* *where g is positive.*

Note that each of the functions f, g and h is defined over a subset of attributes. Thus, despite the fact that we have to assess three functions now instead of one, this may require fewer parameters to specify.

Can we decompose the utility function further if every attribute is utility independent of the rest? The answer is yes: each function in the resulting decomposition is a function of one attribute. However, the full specification of the utility function in such a case still requires the assessment of potentially exponentially many constants.

Proposition 3.2.6: *(adapted from (Keeney and Raiffa 1976), page 293) If every variable is utility independent of the rest there is a function $f_i(V_i)$ for each variable, such that $U_{\succsim}(\mathbf{V})$ is a multilinear combination of the f_i's. That is,*

$$U_{\succsim}(\mathbf{V}) = \sum_{\mathbf{X} \subset \mathbf{V}} (k_{\mathbf{X}} \prod_{V_i \in \mathbf{X}} f_i(V_i))$$

for some constants k and functions f_i.

An interesting case to consider is the situation when each subset of attributes is preferentially independent of its complement. We say in such cases that a set of attributes exhibits *mutual utility independence* (MUI). MUI utility functions have a very simple decomposition:

Proposition 3.2.7: *(adapted from (Keeney and Raiffa 1976), page 289) The attributes in $\mathbf{V}$ are MUI (i.e., every subset of attributes is independent of its complement in $\succsim$) if and only if there exists m functions $f_i(V_i)$ (i.e., each f_i depends on a single variable), such that either*

$$U_{\succsim}(\mathbf{V}) = \prod_{i=1}^{m} f_i(V_i) + c$$

3.2.5 Additive Independence

A much stronger form of independence is *additive independence.* Intuitively, additive independence not only requires that our preferences over values of some attribute are independent of the remaining attributes, it also asserts that the "strength" of these preferences remains unchanged for all possible values of other attributes.

Definition 3.2.8: (adapted from (Keeney and Raiffa 1976), page 295) Let $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ be a partition of $\mathbf{V}$. $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ are *additively independent* (for $\succsim$) if, for any probability distributions p_1 and p_2 that have the same marginals on $\mathbf{Z}_i$ for all i, p_1 and p_2 are indifferent under $\succsim$, i.e., $p_1 \sim p_2$. In this case, we write $\forall_i\ \mathrm{AI}(\mathbf{Z}_i, \mathbf{V} - \mathbf{Z}_i)$. ∎

Consider the following example, due to Bacchus and Grove (1995):

Example 3.2.9: We have two utility attributes: *Health*, with values h_1 and h_2(healthy and ill) and *Wealth*, with values w_1 and w_2 (rich and poor). The agent's utility function is defined as: $U(h_1, w_1) = 5$, $U(h_1, w_2) = 2$, $U(h_2, w_1) = 1$, and $U(h_2, w_2) = 0$. It is easy to see that these two attributes are utility independent of each other: no matter whether the agent is rich or poor he always prefers the lotteries which yield h_1 with higher probability. Similarly, whether he is healthy or ill, he will choose the lottery which gives him a better chance to get rich.

Are these two attributes also additively independent? Consider two probability distributions p_1 and p_2:

$$
\begin{array}{ll}
p_1(h_1, w_1) = 1/4 & p_2(h_1, w_1) = 1/2 \\
p_1(h_1, w_2) = 1/4 & p_2(h_1, w_2) = 0 \\
p_1(h_2, w_1) = 1/4 & p_2(h_2, w_1) = 0 \\
p_1(h_2, w_2) = 1/4 & p_2(h_2, w_2) = 1/2
\end{array}
$$

Intuitively, the increase in agent's utility corresponding to moving from illness to health when he is poor is not the same as the increase in the case when he is rich. In other words, his preference for health is "stronger" when is rich. More technically, he displays a preference for probability distributions in which health and wealth are positively correlated. ∎

Additive independence is a very strong assumption, too strong for many domains. However, in many cases, we can fix a subset of the attributes to a specific value and discover that the remaining (sets of) attributes are additively independent in the resulting *conditional utility function.* The conditional version of additive independence is defined in a straightforward manner:

Definition 3.2.10: (adapted from (Keeney and Raiffa 1976), page 336) $\mathbf{X}$ and $\mathbf{Y}$ are *conditionally additively independent* (CA-independent) given $\mathbf{Z}$ ($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ disjoint, $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} = \mathbf{V}$) if, for any fixed value $\mathbf{z}$ of $\mathbf{Z}$, $\mathbf{X}$ and $\mathbf{Y}$ are additively independent in the conditional preference ordering over $\mathbf{X} \cup \mathbf{Y}$ given $\mathbf{z}$. In this case, we write $\mathrm{CAI}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$. ∎

Conditional additive independence is a very useful notion in practice. Weaker than additive independence, it is more likely to occur in real-life domains.

The notion of additive independence can be also applied to overlapping sets of attributes:

Definition 3.2.11: (Bacchus and Grove 1995) Let $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ be sets of variables *not necessarily disjoint* such that $\mathbf{V} = \bigcup_i \mathbf{Z}_i$. $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ is *generalized additively independent* (for $\succsim$) if, for any probability distributions p_1 and p_2 that have the same marginals on $\mathbf{Z}_i$ for all i, p_1 and p_2 are indifferent under $\succsim$. In this case, we write $\mathrm{GAI}(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_k)$. ∎

All forms of additive independence result in the additive decomposition of the utility function. The simplest one corresponds to pure additive independence:

$$U_{\succsim}(\mathbf{V}) = \sum_{i=1} f_i(\mathbf{Z}_i)$$

for some functions f_i*.*

Conditional additive independence and generalized additive independence cause the utility function to decompose in a similar manner:

Proposition 3.2.13: *(adapted from (Keeney and Raiffa 1976), page 338)* $\mathbf{X}$ *and* $\mathbf{Y}$ *are conditionally additively independent given* $\mathbf{Z}$ *iff* $U_{\succsim}$ *can be written in the form*

$$U_{\succsim}(\mathbf{V}) = f_1(\mathbf{X}, \mathbf{Z}) + f_2(\mathbf{Y}, \mathbf{Z})$$

for some functions f_1*,* f_2*.*

Proposition 3.2.14: *(Bacchus and Grove 1995) Let* $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ *be sets of variables not necessarily disjoint such that* $\mathbf{V} = \bigcup_i \mathbf{Z}_i$*.* $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ *are generalized additively independent for* $\succsim$ *iff* $U_{\succsim}$ *can be written as*

$$U_{\succsim}(\mathbf{V}) = \sum_{i=1}^{k} f_i(\mathbf{Z}_i)$$

for some functions f_i*.*

Additive independence and corresponding to it additive decomposition of the utility function have received a great deal of attention due to their simplicity.

3.2.6 Utility Function Decomposition

The utility function structure plays an important role when we attempt to use decision theory in real-life applications. Our prenatal diagnosis domain from Section 1.4

$\forall_i$ UI$(V_i, \mathbf{V} - V_i)$	multilinear	$\sum_i \lvert \mathrm{Dom}_{V_i} \rvert + 2^{\lvert \mathbf{V} \rvert}$
MUI$(\mathbf{V})$	additive or multiplicative	$\sum_i \lvert \mathrm{Dom}_{V_i} \rvert + 1$
Additive Independence		
AI$(\mathbf{X} \subset \mathbf{V}, \mathbf{Y} = \mathbf{V} - \mathbf{X})$	additive	$\lvert \mathrm{Dom}_{\mathbf{X}} \rvert + \lvert \mathrm{Dom}_{\mathbf{Y}} \rvert$
$\forall_i$ AI$(V_i, \mathbf{V} - V_i)$	additive	$\sum_i \lvert \mathrm{Dom}_{V_i} \rvert$
CAI$(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})$, $\mathbf{V} = \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$	additive	$\lvert \mathrm{Dom}_{\mathbf{X}} \rvert \times \lvert \mathrm{Dom}_{\mathbf{Z}} \rvert + \lvert \mathrm{Dom}_{\mathbf{Y}} \rvert \times \lvert \mathrm{Dom}_{\mathbf{Z}} \rvert$
GAI$(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_k)$, $\mathbf{V} = \bigcup_i \mathbf{Z}_i$	additive	$\sum_i \lvert \mathrm{Dom}_{\mathbf{Z}_i} \rvert$

Table 3.1: Utility function decomposition — summary

is not unusual in its complexity — many medical decision problems have utility functions that depend on five or more attributes. To specify such a function completely (assuming no structure), we need to assess more than 100 parameters.

Table 3.1 summarizes the decomposition results for different types of independence assumptions. As can be easily seen, each independence assumption reduces the number of parameters needed to represent the utility function.

Assuming some independencies between utility function attributes not only allows us to use a more compact representation, but also reduces the complexity of inference (such as expected utility computation used in determining the best action to take)[2] and knowledge acquisition. As we describe in detail in Section 3.4, assessing people's utility functions in real-world domains presents considerable problems. The number of parameters in the utility function is often critical for the success of the utility elicitation process.

Table 3.2 presents the decomposition results for some specific independence assumptions in the prenatal diagnosis domain (Section 1.4). We use as examples assumptions people often hold, e.g., that the (dis)utility of invasive testing is (utility or additively) independent of all the other parameters.

[2]in the influence diagram representation (Jensen, Jensen, and Dittmer 1994)

UI(DKL, TF)	$18 + 2 \times 6 = 30$
$\forall_i$ UI($V_i, \mathbf{V} - V_i$)	$3 + 2 + 3 + 2 + 3 + 2^5 = 13 + 32 = 45$
MUI($\mathbf{V}$)	$3 + 2 + 3 + 2 + 3 + 1 = 13 + 1 = 14$
Additive Independence	
AI($T, DKLF$)	$3 + 36 = 39$
AI(DKL, TF)	$18 + 6 = 24$
$\forall_i$ AI($V_i, \mathbf{V} - V_i$)	$3 + 2 + 3 + 2 + 3 = 13$
CAI($TF, K \mid DL$),	$6 \times 6 + 3 \times 6 = 54$
CAI($TF, DK \mid L$),	$6 \times 3 + 6 \times 3 = 36$
CAI($TF, LK \mid D$),	$6 \times 2 + 9 \times 2 = 30$
GAI(TL, FL, LDK)	$9 + 6 + 18 = 33$
GAI(T, FL, LD, DK)	$3 + 6 + 6 + 6 = 21$

Table 3.2: Utility function decomposition in the prenatal diagnosis domain. Attributes: testing (T), Down's status (D), pregnancy loss (L), knowledge (K), and future pregnancy (F)

3.3 Psychological Aspects of Decision Making

3.3.1 Attitudes Towards Risk

People's risk attitudes have been studied primarily in the context of the utility of money. It is commonly assumed that people exhibit a monotonic preference for money. That is, all other things being equal, they prefer to have more money rather than less. However, our utility function does not have to be a linear function of money. Note that this discussion applies equally to any monotonically increasing utility function defined over a real-valued attribute. The example of money is commonly chosen since it is considered to be the most intuitive.

Consider a lottery $L = [o_i, p, o_j]$, where o_i and o_j are monetary payoffs. Associated with this lottery are two expectations: its expected monetary value,

$$\mathrm{EV}(L) = po_i + (1 - p)o_j,$$

These two numbers are not necessarily equal. Let $o_i = \$1,000$, $o_j = \$0$ and $p = 1/2$. The expected value of the lottery is $\mathrm{EV}(L) = \$500$. What is its expected utility? Most people, when faced with a choice between a lottery and a guaranteed payoff of the expected value of that lottery, would choose the expected value. Thus,

$$U([\$1,000, 1/2, \$0]) < U(\$500).$$

According to the utility theory, every person will be indifferent between our lottery and a guaranteed payoff o_k, where the monetary value of o_k is between 0 and 1,000. o_k is called the *certainty equivalent* of the lottery. The difference between the expected value of the lottery and its certainty equivalent is the *risk premium* or the *insurance premium*. If $o_k = \$400$, the risk premium is \$100.

Why is the certainty equivalent of the lottery lower (for most people) than its expected value? It is explained by the concave shape of the utility function: we reportedly value our first $\$1,000$ more than the additional ones. The concavity of the utility function (and implied by it the positive sign of the risk premium) is referred to as the agent's *risk aversion*. If the utility function is convex (as it often is for negative payoffs), the risk premium is negative and leads to *risk seeking* behavior. Linear utility functions imply *risk neutrality*. People are generally risk averse when it comes to monetary gains and risk seeking with respect to monetary losses.

3.3.2 Uncertainty Perception and Prospect Theory

People make decisions taking into account their preferences over outcomes and their beliefs concerning the likelihood of these outcomes. Two famous psychologists, Tversky and Kahneman (1986a) have shown that human ability to estimate these likelihoods is affected by many biases. People commonly use heuristics in their probability assessments, which simplify the problem, but often lead to serious errors.

Specifically, they often pay disproportionate attention to low probability events and treat high probability events as though they were less likely than they actually are (Tversky and Kahneman 1986b).

To explain the apparent inconsistencies in human behavior, Kahneman and Tversky proposed *prospect theory* (Kahneman and Tversky 1979) as a more accurate descriptive model of decision making under uncertainty. One part of the theory introduces the notion of a *decision weight* π. Decision weight is a monotonic function of probability, but it is not itself a probability measure. $\pi(0) = 0$ and $\pi(1) = 1$, but the function is not well-behaved near the endpoints. For low probabilities, $\pi(p) > p$, but $\pi(p) + \pi(1-p) \leq 1$ — a phenomenon called subcertainty.[3]

The theory suggests that people transform probabilities into decision weights when faced with a decision problem. This can happen because of psychological misconceptions of numerical probabilities, but it can also be a conscious and deliberate choice of the decision maker. For example, one can decide to pay relatively more attention to worse outcomes than to better ones. Note that the decision weights are supposed to measure not only the perceived likelihood of different events, but also "the impact of events on the desirability of prospects" (Kahneman and Tversky 1979), thus blurring the distinction between probability and utility.

According to Kahneman and Tversky, the intuitive notion of risk is not adequately captured by the assumed concavity of the utility function. On the other hand, the use of decision weights accounts much better for people's dislike of lotteries. Prospect theory can explain many famous paradoxes of decision theory, such as Allais' paradox described in Section 3.3.3. Note, however, that the prospect theory is purely descriptive. It attempts to explain people's behavior under uncertainty without making any normative claims.

[3] Another part of prospect theory deals with the definition of the value function, an alternative to the utility function. The value function is defined on deviations from the reference point (usually the status quo), it is generally concave for gains and convex for losses and steeper for losses than for gains.

early days of decision theory, people have shown that they are sometimes violated. The most famous of these "paradoxes" is due to Allais (1953).

Consider a choice between obtaining \$1 million for sure and a lottery, in which you have a 10% chance of winning \$5 million, 89% chance of winning \$1 million and 1% chance of winning nothing. Most people, when faced with this choice, prefer \$1 million for sure. That preference implies the following inequality:

$$U(1m) > 0.10 \cdot U(5m) + 0.89 \cdot U(1m) + 0.01 \cdot U(0).$$

Next, consider a choice between two lotteries. In the first lottery, you have a 10% chance of winning \$5 million and a 90% chance of getting nothing. In the second, you have an 11% chance of winning \$1 million and an 89% chance of getting nothing. Most people prefer the first lottery. That preference implies

$$0.10 \cdot U(5m) + 0.90 \cdot U(0) > 0.11 \cdot U(1m) + 0.89 \cdot U(0).$$

As can be easily seen, these two inequalities are contradictory. Some researchers have suggested that the inconsistency is caused by "mental shortcuts", such as dismissing the difference between 10% and 11% as negligible. There are people, however, who stick to their preferences even after seeing the expected utility analysis.

Their behavior can be explained using the notion of *regret*. A decision maker who chooses a lottery over the sure \$1 million may feel devastated if the result of the lottery is the unlikely event of getting nothing. In the second choice, there is no option with a sure payoff, so regret does not play a role.

Regret was first suggested as a decision criterion by Savage (1951). It is often called the *minimax risk criterion*. According to this criterion, we should associate with each outcome not only some utility value, but also a regret value. To compute the regret value, we group outcomes into sets, placing in every set outcomes differing only by the decision maker's choice and identical in their attributes related to the

The suggestion of regret as the sole decision criterion was criticized on several grounds (see (Luce and Raiffa 1957) for discussion and examples). First, it is not clear that the differences in utilities of different states are an adequate measure of regret. In other words, the regret of going from a state of utility 5 to a state of utility 3 may be very different than the regret of going from a state of utility 105 to a state of utility 103. Second, the regret criterion is not independent of irrelevant alternatives. That is, adding a new, even sub-optimal, course of action to the decision problem may alter our assessment of which of the two formerly present courses of action is preferable. The reason for this unintuitive behavior is simple: adding a new course of action requires that we also add one new outcome (corresponding to this new course of action) to every set. This change may alter all regret values and thus our decision choice.

A more refined approach to accounting for the role of regret in decision making was proposed by Bell (1982). He suggested incorporating regret as an additional attribute in a multi-attribute utility function. (A utility function should be a decreasing function of regret.) This formulation allows us to stay within the traditional framework of maximizing expected utility. The subutility function over the regret attribute can be estimated in the process of utility elicitation just like the other components of the overall utility function. Bell shows how his regret theory can be used to explain such apparently irrational behaviors as gambling on negative expected value lotteries or buying costly insurance.

3.4 Utility Elicitation

People have very different values and preferences. In a given decision problem, every person's utilities may be different. In building a decision support system, we assess the probabilities in our model only once. (The probabilities of chance events do not change across a population.) In contrast, utilities must be elicited many times —

primary applications were in the field of economics, and most research concentrated on estimating people's utility for money.

In the last twenty years, however, as decision theory has found uses in other domains, more attention has been given to eliciting utilities for a large number of discrete outcomes. We will concentrate on that aspect of utility elicitation here. Since this work is motivated in large part by the need to create effective decision support tools for medical domains, we pay particular attention to utility elicitation practices in health care.

3.4.1 Standard Gamble

The *standard gamble* is the classical method of measuring preferences. First presented by von Neumann and Morgenstern (1947), it is based on the axioms of utility theory. Recall the continuity axiom:

$$o_i \succ o_j \succ o_k \Rightarrow \exists p \; [o_i, p, o_k] \sim o_j$$

According to this axiom, to assess the utility of outcome o_j, we need to find two other outcomes: outcome o_i, which is preferred to o_j, and outcome o_k, which is less valuable than o_j. This is easy: we usually have two anchor outcomes (see Section 3.1.2) which are the most and the least valued outcomes in $\mathbf{O}$ and they can be used for that purpose.

The next step is to construct the following decision problem: imagine the choice between obtaining the outcome o_j for sure and the lottery that offers o_i with probability p and o_k with probability $1-p$ ($[o_i, p, o_k]$) for some p. The value p^* for which the user is indifferent between the lottery and the outcome o_j for sure is called the *indifference point*.

How do we find the indifference point? It is usually assumed that we cannot ask the user to assess the value of p^* directly; it is too difficult to estimate. Therefore, involving the two anchor outcomes where $p = 1 - a$. If the user prefers the lottery, p is decremented by a, thus making o_i less likely and o_k more likely, and the user is asked to choose again. This process is repeated until the indifference point is found. Other methods include *ping-pong*, which alternates between high and low values of p ($1 - a$, a, $1 - 2a$, $2a$, etc.), and binary search.

Once the indifference point p^* is found, we use the basic theorem of utility theory (Equation (3.2)) to compute the utility of outcome o_j:

$$U(o_j) = U([o_i, p^*, o_k]) = p^* U(o_i) + (1 - p^*) U(o_k)$$

If we are using standard anchor outcomes with $U(o_i) = 1$ and $U(o_k) = 0$, it follows that $U(o_j) = p^*$.

A typical standard gamble question in a medical domain is formulated in this way:

> "Imagine that you have a certain health condition which limits your activities in a specific way [the detailed description follows]. Imagine that there is a new experimental drug which only needs to be taken once. If taken, it will cure the condition p percent of the time, and $1 - p$ percent of the time it will cause a painless death. Would you take the pill?"

Unfortunately, many independent studies have established that the final values obtained in the process of standard gamble elicitation are sensitive to the choice of anchors (Llewellyn-Thomas, Sutherland, Tibshirani, Ciampi, Till, and Boyd 1982) and to the choice of the search procedure (Lenert, Cher, Goldstein, Bergen, and Garber 1998). In addition, people are better at making gross distinctions than fine-grained ones. As a result, a user's answers are the most reliable far from the actual indifference point and most error-prone immediately around it.

outcome and a magic pill which may result in their death. An alternative method, *time trade-off*, was developed by (Torrance, Thomas, and Sackett 1972) specifically for use in health care research. It attempts to maintain the spirit of the standard gamble procedure while restricting the choices to the certain outcomes rather than lotteries.

Time trade-off estimates what percentage of the remaining years of life in the current health condition the patient would be willing to give up in order to spend the rest of his life in a condition he considers superior. The two choices in a time trade-off elicitation procedure are: (1) t years (where t is equal to the patient's life expectancy) in the current state o_j; or (2) t' years (where $t' < t$) in perfect health ($o_\top$). As in standard gamble, we vary t' until the indifference point t^* is reached. The utility of the patient's current state of health is taken to be $t^*/t \cdot U(o_\top)$.

Time trade-off was designed to produce similar results to the standard gamble method at smaller cost and with less cognitive burden on the respondent.

A potential difficulty with the time trade-off method is the role of discounting. Some researchers have suggested that people do not value each year of their lives equally, giving less weight to the ones in the distant future (Lipscomb 1989; Stiggelbout, Kiebert, Kievit, Habbema, and Haes 1995).

A recent study of cancer patients in the Netherlands (Stiggelbout, Kiebert, Kievit, Habbema, and Haes 1995) reported many interesting problems with administering this elicitation method. Several patients refused to entertain the idea of trading some portion of their life on religious grounds. Some felt that the period offered exceeded the time they would be able to endure in their current health state, which made the task impossible. The majority of the patients were not willing to trade off a fraction of their life in any situation. The ones that were often gave inconsistent answers: their indifference points for different values of t did not scale proportionally to t.

with clearly defined endpoints representing the utilities of the anchor outcomes. The user is asked to specify the utility for the given outcome o_j by placing a mark on the line between the two anchors. The placement of the mark (its distance from the endpoints) should indicate the relative desirability of the outcomes. A commonly used variation of this method is to discretize the scale into 10 or more intervals and ask the respondents to sort the outcomes into appropriate categories.

According to (Fromberg and Kane 1989b), visual analog scale is the most frequently used method for measuring health state preferences. However, it has its own share of problems. Some studies have shown (Read, Quinn, Berwick, Fineberg, and Weinstein 1984) that the utilities elicited using the visual analog scale technique suffer from the "distribution effect": people tend to distribute the marks fairly evenly along the scale, since crowding them at one end is less visually appealing.

3.4.4 Choosing the Elicitation Method

The standard gamble is the oldest method. It is theoretically appealing, since it is built on the fundamental axioms of utility theory. It is the only one which forces the respondent to make choices under uncertainty (even if the choice is made with respect to an artificial scenario). Consequently, the standard gamble is very sensitive to risk aversion, which has been considered to be one of its strengths by some researchers and a major weakness by others. The main drawback of the standard gamble method is the frequently demonstrated difficulty in explaining it to users.

On the other end of the spectrum, the visual analog scale method is the easiest one to use and the most difficult to justify theoretically. The idea is easy to grasp, so there is no need to train the interviewers. The users are comfortable answering the questions. However, it completely ignores the uncertainty inherent in many decision problems. It is not clear whether it is an appropriate assessment method in domains involving risky choices.

to the two outcomes involved. On the other hand, it does not require evaluating the utility of a lottery. For this reason, it is significantly easier for the users to understand than standard gamble. At the same time, however, just as in case of the visual analog scale, the utility values assessed by time trade-off are not necessarily a good basis for decision making in the presence of uncertainty.

The standard gamble and the time trade-off methods are typically more expensive, especially for large population studies. They both require asking many questions per outcome to determine the indifference points, thus resulting in lengthy interviews. They also require well-trained interviewers. The visual analog scale method is faster, cheaper and cognitively easier for the users.

3.4.5 Validity of Utility Assessment

One may ask if there exists such a thing as a person's "true" utility value for a given outcome. If it does exist, is the person aware of it? Can this awareness be brought about by the process of utility elicitation? How well can we estimate the "true" utility values using existing assessment methods? We don't have answers to all these questions. However, we can develop some intuitions based on the results of studies comparing people's utilities that have been assessed at different points in time and using different methods.

In general, people's utilities tend to remain fairly stable over time. The correlations between the utility values elicited up to 6 weeks apart for the same outcomes and using the same method are quite high (Fromberg and Kane 1989b), typically around 0.8. The correlations for assessments taken 1 year apart are lower (0.5 – 0.6), but that may reflect a genuine preference change rather than a measurement error.

In many domains a preference change can be a credible explanation. Studies show that people's utilities sometimes depend on life experiences. In our prenatal diagnosis domain, women who reported knowing a person with a Down's syndrome consistently reported by (Schkade and Kahneman 1998; Lenert, Treadwell, and Schwartz 1999; Jansen, Stiggelbout, Wakker, Nooy, Noordijk, and Kievit 2000). In many cases, utilities for a highly undesirable outcome (such as a serious health condition or an unpleasant procedure) are higher among people who have previously experienced it than among those for whom the outcome is purely hypothetical. On the other hand, predicted utilities for desirable outcomes (such as winning a lottery) tend to be overestimated. Schkade and Kahneman (1998) attribute this discrepancy to the phenomenon they call a *focusing illusion*: when the attention of the person making a utility judgment is focused on one aspect of the outcome, this aspect is likely to be given an unduly large weight.

When we examine the consistency of the utility values elicited from the same people for the same outcomes using different assessment methods, the results are much less encouraging. Numerous studies ((Torrance 1976; Read, Quinn, Berwick, Fineberg, and Weinstein 1984; O'Leary, Fairclough, Jankowski, and Weeks 1995; Stiggelbout, Eijkemans, Kiebert, Kievit, Leer, and Haes 1996) among many others) have demonstrated surprisingly low correlations between these utilities. Some researchers have suggested specific forms of functional dependencies between results of different assessment methods, but none of these findings have been consistent across different studies. The only result demonstrated in multiple studies and uniformly accepted is the fact that the utilities elicited using the standard gamble method are higher than the utilities for the same outcomes elicited by other techniques. The commonly given explanations for this phenomenon are the risk aversion and imperfect probability perception (see Section 3.3.1 and Section 3.3.2) which affect the evaluation of lotteries in standard gamble.

Interestingly, it has been shown (Read, Quinn, Berwick, Fineberg, and Weinstein 1984) that the dependence between the utility attributes may also vary depending on the assessment method.

The stability of utility values across assessments separated by short periods of time by the limitations of our imagination or the lack of adequate descriptions used in utility elicitation of hypothetical outcomes. It is clear, however, that the discrepancies between results of utility assessments performed with different techniques cast serious doubts on our ability to approximate a person's utility function well enough to avoid serious errors in decision making.

3.4.6 Eliciting Decomposed Utility Functions

Decomposed utility functions typically require many fewer parameters than utility functions with no structure. This property should make eliciting such functions much easier. However, eliciting decomposed models raises its own problems.

Determining the Structure

First, how do we know which independence properties hold? Every person's utility function may be different. This is true not only of the numerical values assigned to the outcomes, but also of the independencies present. For example, in our prenatal diagnosis domain, some pregnant women may have a fixed preference for (or against) a future pregnancy regardless of everything else, including the outcome of the current one. In such cases the attribute "future pregnancy" is utility independent of the set of remaining attributes. Some others, however, may change their preference depending on whether or not the current pregnancy ends in a normal birth, resulting in a utility model where that independence is absent.

Of course, we can always determine the appropriate decomposition given elicited utility values for an adequate number of complete outcomes. Keeney and Raiffa (1976) describe the following procedures designed to verify various independence properties.

Consider a utility function over two subsets of attributes, $\mathbf{V} = \mathbf{X} \cup \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ disjoint. If the utility function is additively independent, the following condition for all $\mathbf{x}$, $\mathbf{y}$ given specific $\mathbf{x}'$, $\mathbf{y}'$. Note that the marginals on $\mathbf{X}$ and $\mathbf{Y}$ are the same in both lotteries, so this property follows from Proposition (3.2.8).

This property immediately suggests the procedure to verify additive independence: choose some fixed $\mathbf{x}'$ and $\mathbf{y}'$ and instantiate the lotteries in Equation (3.3) for several different values of $\mathbf{x}$ and $\mathbf{y}$. The number of different values of $\mathbf{x}$ and $\mathbf{y}$ should be high enough to adequately cover the space.

Similarly, we can verify whether $\mathbf{X}$ is utility independent of $\mathbf{Y}$ by finding the certainty equivalents for a number of lotteries $[\langle \mathbf{x}_\perp, \mathbf{y} \rangle, 0.5, \langle \mathbf{x}_\top, \mathbf{y} \rangle]$ with varying values $\mathbf{y}$. If all the certainty equivalents are the same, regardless of the $\mathbf{y}$ value, the independence property holds.

Another technique for inferring the utility function decomposition from data was proposed by Anderson (1974a, 1974b) as one of the applications of his *functional measurement* approach to the measurement of subjective constructs. The theory describes perception as integration of stimulus information. It considers individual percepts to be the results of a combination of responses to stimuli of different kinds. One of the motivating examples is depth perception, which involves integration of our responses to cues like interposition, stereopsis and texture.

Anderson assumes that a response to a stimulus is based on a separate subjective scale and has a subjective weight, both potentially different for every person. He proposes several algebraic models to account for different combinations of responses: additive, multiplicative and multilinear among them. One of Anderson's contributions is the introduction of the explicit error term indicating the presence of noise in the measurement process.

Anderson's integration theory can be applied in a straightforward way to perception of abstract concepts, social judgments, and, in particular, to decision making in situations involving uncertainty (Anderson 1976; Anderson and Shanteau 1970). In this case, individual utility attributes play the role of stimuli and the utility values

An important contribution of Anderson's theory is a goodness-of-fit measure which he uses to evaluate different hypotheses. He uses statistical analysis of variance to check how well the data matches a given model. However, he does not consider responses defined for subsets of stimuli or conditional forms of independence properties, thus reducing the space of possible models drastically. Therefore, exhaustive search over possible models becomes a viable option.

A serious drawback of both Keeney and Raiffa's and Anderson's approaches is the necessity of eliciting a significant number of utility values for full outcomes from every person whose utility function we want to assess. Since the purpose of decomposition testing is usually to avoid a long and difficult interview, this necessity makes their techniques infeasible in large domains.

In practice, most attempts to infer the utility function decomposition from data have been much more straightforward. Typically, researchers assume a fixed, simple utility model (almost always the additive model) of the utility function. Then, they elicit the utilities of a small number of complete outcomes. The parameters of the model are estimated using simple least-squares regression (Fischer 1979; Fromberg and Kane 1989a).

It is not clear, however, how to validate a model obtained in this way. All simple methods for checking for the presence of additive decomposition assume that the agent's utility assessments are free from random response error. It is hard to determine how large a deviation from linearity to tolerate before rejecting the given model.

Estimating Model Parameters

Even if the underlying structure of the model is known, it is difficult to elicit the subutility functions directly. Utilities cannot be marginalized like probabilities — it is hard to assign a utility to a specific value of one attribute of an outcome without making (often subconsciously) some assumptions about the values of all the other attributes. It is particularly difficult if the attributes are not additively independent.

pregnancy.

Therefore, in order to obtain a reliable assessment, we need to make the assumptions about the remaining attributes explicit and elicit conditional utilities (see Section 3.2.2). Let $\mathbf{V} = \mathbf{X} \cup \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ disjoint, where $\mathbf{X}$ is a subset of attributes with no independence properties holding among the members of the set. For every value $\mathbf{y}$ of $\mathbf{Y}$, we elicit the conditional utility function $U(\mathbf{X}, \mathbf{y})$ (sometimes denoted $U_{\mathbf{y}}(\mathbf{X})$ in the literature).

Since every conditional utility function is assessed using its own scale, in addition to these functions, we need to assess the scaling constants. The number of scaling constants can range from linear in the number of attributes (for an additive model) to exponential (for a multilinear model) (see Section 3.2). To assess r scaling constants, Keeney and Raiffa (1976) suggest obtaining r equations with r unknowns using any combination of certainty equivalence and other properties.

Finally, it is advisable to elicit utilities for a few more outcomes and check their consistency with the proposed model. Such consistency checks often lead to partial reevaluation of the model.

3.4.7 Cognitive Difficulties

Regardless of the method chosen, the process of utility elicitation is cognitively difficult.

The outcomes whose utility the respondent is supposed to assess are often hypothetical. If the respondent has not experienced a given outcome, a great deal depends on providing an accurate and detailed description. Many studies have shown that utilities are sensitive to the type of the description (in terms of losses or gains) (Tversky and Kahneman 1986b) and even the gender and race of the hypothetical patient represented in the description (Lenert, Ziegler, Lee, Unfred, and Mahmoud 2000). There is an obvious trade-off between the desire to represent all the relevant domains, it is hard to find an outcome which would be rated highest (lowest) by everyone. In such cases, we typically resort to outcomes not related to the given decision problem. The top anchor is usually taken to represent "perfect health" and the bottom anchor the death of the decision maker. The hope is that these two outcomes are equally (un)desirable for everybody. That, however, is not necessarily always the case. Moreover, it is very difficult for people to estimate the utility of lotteries involving their own deaths. Some refuse even to entertain the idea. Not surprisingly, people are typically extremely risk averse towards that outcome. Studies have shown that people's utilities assessed using the standard gamble method differ considerably depending on the choice of the bottom anchor (Llewellyn-Thomas, Sutherland, Tibshirani, Ciampi, Till, and Boyd 1982).

Finally, utility elicitation requires long interviews, which cause fatigue that can influence a user's assessments. For every outcome, one first needs to allow the respondent to study the description carefully. Then, for standard gamble and time trade-off, a sequence of questions must be answered before we find the indifference point. According to some practitioners, one can assess between 5 and 10 outcomes in one hour. To check for consistency, we sometimes ask preference questions, i.e., ones designed to establish the ordinal ranking. We can use the result of these pairwise comparisons to check if the utilities assigned in the process of utility elicitation satisfy Equation (3.1). Unfortunately, as fatigue grows, the answers often start becoming inconsistent. The respondents, when faced with these inconsistencies, often revise their answers. It is not clear, however, to what extent we can rely on these "corrected" utilities.

In general, the process of utility elicitation is long, tedious and very noisy. Cognitive difficulties are prevalent in all elicitation approaches, and there is no clear solution to this problem. The numbers we obtain are only very rough approximations to the "true" utility values.

Chapter 4

Utilities as Random Variables

Practical experience shows that it is virtually impossible to elicit a person's exact utility function. In domains with large outcome spaces it is impossible to ask about every possible outcome. A few utility values we can elicit are unlikely to represent the person's true preferences — they may be imprecise due to a bias in the elicitation method, fatigue, inability to imagine the outcome based on the description and many other factors.

In this thesis, we propose to take another approach. Rather than aiming at a completely specified utility function for a given person, we maintain a *probability distribution* representing our beliefs about that person's utility function.

4.1 Distributions over Utilities

Treating utilities as random variables allows us to deal in a principled way with the uncertainty inherent in utility assessments.

Using a probability distribution over utility parameters, we can represent a complete lack of knowledge about a person's preferences as well as a slight uncertainty over one utility parameter. We can deal with ignorance (absence of information about a utility parameter) as well as lack of precision (e.g., due to noise and utility elicitation bias). We can use any prior knowledge we may have. We can update our beliefs based on new information.

erally, utility parameters) $o \in \mathbf{O}$ needed to specify the utility function U completely.

We view the different quantities $\{U_o\}_{o \in \mathbf{O}}$ as a set of continuous-valued random variables (in the interval $[0, 1]$), with joint probability density function (PDF) $p(\mathbf{U})$ ($\mathbf{U} = \{U_{o_1}, \ldots, U_{o_n}\}$), representing our beliefs about the agent's utilities.

This type of PDF can be represented in many ways. Our approach applies to any representation that allows random samples to be generated from the PDF, thereby allowing moments of the distributions and expectations over it to be estimated numerically. However, inference with such a utility model can be made more efficient in cases where the PDF allows some computations to be done in closed form. A very convenient representation is one where PDFs are jointly Gaussians or a mixture of Gaussians (cut off to fit in the $[0, 1]$ hypercube). Recall that an n-dimensional Gaussian distribution is defined as

$$P(X) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right),$$

where μ is the mean and Σ is the covariance matrix. A Gaussian can represent dependencies between a person's utilities for different outcomes. In general, any PDF can be approximated arbitrarily well with a mixture of Gaussians (see e.g., (Bishop 1995)). Furthermore, there are efficient algorithms for estimating these PDFs from data and conditioning them on new information.

As we discussed in Section 3.2, we can sometimes find utility independence or additive independence properties holding among utility attributes. For example, in our prenatal diagnosis domain (see Section 1.4), we can hypothesize that the attribute "Testing" may be additively independent of the other attributes. If this hypothesis turns out to be true, the number of parameters required to specify our utility function is reduced from 108 to 39.[1]

[1]Since "Testing" is ternary, the number of parameters required to specify the subutility function over the remaining attributes is $108/3 = 36$ and the number of parameters needed to specify the subutility function over "Testing" is 3.

a function over a subset of utility attributes, see Section 3.2.2). As the utility variables are linear in these parameters, a mixture of Gaussians over the subutility parameters induces a mixture of Gaussians over the utility function $\mathbf{U}$ (see Section 4.3).

4.2 Representation for Factored Utility Functions

We are most interested in exploiting the structure corresponding to various additive independence properties among utility attributes (see Section 3.2.5). Recall that we defined each outcome as an assignment to a set of attribute variables $\mathbf{V} = \{V_1, \ldots, V_n\}$. Each variable V_i has a domain Dom_{V_i} of two or more elements.

Definition 4.2.1: Let $\mathcal{C}$ be a set of clusters of variables $\mathbf{C}_1, \ldots, \mathbf{C}_r$. We say that a utility function is *factored according to* $\mathcal{C}$ if there exist functions $u_i : \mathrm{Dom}_{\mathbf{C}_i} \mapsto I\!R$ $(i = 1, \ldots, r)$ such that $u(\mathbf{v}) = \sum_i u_i(\mathbf{c}_i)$ where $\mathbf{c}_i$ is the assignment to the variables in $\mathbf{C}_i$ in $\mathbf{v}$. We call the functions u_i *subutility functions.* ∎

Note that this definition encompasses purely additive, conditionally additive and generalized additive utility functions.

The decomposition of the utility function is equivalent to the following property about the preference ordering (see Proposition (3.2.12), Proposition (3.2.13) and Proposition (3.2.14)).

Proposition 4.2.2: (Bacchus and Grove 1995) *A utility function u is factored according to $\mathcal{C}$ if and only if for any probability distributions p_1 and p_2 that have the same marginals on $\mathbf{C}_i$ for all i, it is the case that p_1 and p_2 are indifferent under u.*

Factored utilities admit a representation in terms of subutility functions over a much smaller domain. They can therefore be specified using a much smaller set of parameters. However, there are many slightly different ways to parameterize a

Definition 4.2.3: We say that two functions h, h' over some domain Ω are *orthogonal* if $\sum_{\omega\in\Omega} h(\omega)\cdot h'(\omega) = 0$. ▮

Our goal will be to construct a fixed *basis* $h_{\mathcal{C}}$ of orthogonal functions, and represent a factored utility function u over $\mathcal{C}$ as a linear combination of the functions in this basis. The coefficients $\mathbf{w}$ of the different basis functions would be the parameters specifying u. The orthogonality property will allow us to perform the computation described in the subsequent sections more efficiently.

The atomic units in the construction of our basis are the basis functions that depend only on a single variable. For each variable V with values $v_1, \ldots, v_k$, we define a set of k basis functions $h_1^V, \ldots, h_k^V : \mathrm{Dom}_V \mapsto I\!\!R$. Our construction is such that:

- $h_1^V \equiv 1$, i.e., $h_1^V(v_i) = 1$ for all i;
- the h_i^V functions are pairwise orthogonal.

For a binary-valued attribute B, we simply define:

$$\begin{aligned} h_2^B(v_1) &= 1 \\ h_2^B(v_2) &= -1 \end{aligned}$$

For a three-valued attribute C, we define:

$$\begin{array}{rcr|rcr} h_2^C(v_1) & = & 1 & h_3^C(v_1) & = & 1 \\ h_2^C(v_2) & = & 0 & h_3^C(v_2) & = & -2 \\ h_2^C(v_3) & = & -1 & h_3^C(v_3) & = & 1 \end{array}$$

In general, we can define a set $\mathcal{H}[V]$ of k orthogonal basis functions for any k-ary variable V. Note that, as the functions are orthogonal, they span the space of all tions over the entire set of outcomes. With a slight abuse of notation, we will view a function h_i^V as a function over $\mathrm{Dom}_{\mathbf{V}}$. Let o be an outcome; recall that o defines a value $V[o]$ for each variable $V \in \mathbf{V}$. We simply define $h_i^V(o) = h_i^V(V[o])$. Similarly, we can view a function h_i^V as a function over $\mathrm{Dom}_{\mathbf{C}}$ for some cluster of variables $\mathbf{C}$. Each assignment $\mathbf{c}$ defines a value $V[\mathbf{c}]$ for each variable $V \in \mathbf{C}$ and $h_i^V(\mathbf{c}) = h_i^V(V[\mathbf{c}])$.

We can now define a basis for a cluster of variables $\mathbf{C}$ as the set of all functions that are composed as products of basis functions for the individual variables in $\mathbf{C}$:

$$\mathcal{H}[\mathbf{C}] = \{ \prod_{V \in \mathbf{C}} h^V \;:\; h^V \in \mathcal{H}[V] \}.$$

Example 4.2.4: Consider a cluster $\mathbf{C}$ with three attributes: A, B and C. A and B are binary and C is ternary. In this case, $k = 2 \times 2 \times 3 = 12$. The basis functions for individual attributes in this cluster are:[2]

$$\begin{array}{rcr|rcr|rcr|rcr}
h_2^A(\mathbf{c}_1) & = & 1 & h_2^B(\mathbf{c}_1) & = & 1 & h_2^C(\mathbf{c}_1) & = & 1 & h_3^C(\mathbf{c}_1) & = & 1 \\
h_2^A(\mathbf{c}_2) & = & -1 & h_2^B(\mathbf{c}_2) & = & 1 & h_2^C(\mathbf{c}_2) & = & 1 & h_3^C(\mathbf{c}_2) & = & 1 \\
h_2^A(\mathbf{c}_3) & = & 1 & h_2^B(\mathbf{c}_3) & = & -1 & h_2^C(\mathbf{c}_3) & = & 1 & h_3^C(\mathbf{c}_3) & = & 1 \\
h_2^A(\mathbf{c}_4) & = & -1 & h_2^B(\mathbf{c}_4) & = & -1 & h_2^C(\mathbf{c}_4) & = & 1 & h_3^C(\mathbf{c}_4) & = & 1 \\
h_2^A(\mathbf{c}_5) & = & 1 & h_2^B(\mathbf{c}_5) & = & 1 & h_2^C(\mathbf{c}_5) & = & 0 & h_3^C(\mathbf{c}_5) & = & -2 \\
h_2^A(\mathbf{c}_6) & = & -1 & h_2^B(\mathbf{c}_6) & = & 1 & h_2^C(\mathbf{c}_6) & = & 0 & h_3^C(\mathbf{c}_6) & = & -2 \\
h_2^A(\mathbf{c}_7) & = & 1 & h_2^B(\mathbf{c}_7) & = & -1 & h_2^C(\mathbf{c}_7) & = & 0 & h_3^C(\mathbf{c}_7) & = & -2 \\
h_2^A(\mathbf{c}_8) & = & -1 & h_2^B(\mathbf{c}_8) & = & -1 & h_2^C(\mathbf{c}_8) & = & 0 & h_3^C(\mathbf{c}_8) & = & -2 \\
h_2^A(\mathbf{c}_9) & = & 1 & h_2^B(\mathbf{c}_9) & = & 1 & h_2^C(\mathbf{c}_9) & = & -1 & h_3^C(\mathbf{c}_9) & = & 1 \\
h_2^A(\mathbf{c}_{10}) & = & -1 & h_2^B(\mathbf{c}_{10}) & = & 1 & h_2^C(\mathbf{c}_{10}) & = & -1 & h_3^C(\mathbf{c}_{10}) & = & 1 \\
h_2^A(\mathbf{c}_{11}) & = & 1 & h_2^B(\mathbf{c}_{11}) & = & -1 & h_2^C(\mathbf{c}_{11}) & = & -1 & h_3^C(\mathbf{c}_{11}) & = & 1 \\
h_2^A(\mathbf{c}_{12}) & = & -1 & h_2^B(\mathbf{c}_{12}) & = & -1 & h_2^C(\mathbf{c}_{12}) & = & -1 & h_3^C(\mathbf{c}_{12}) & = & 1
\end{array}$$

[2]We omit the first basis function for each variable, since $h_1^V(\mathbf{c}_i) = 1$ for all V and all i.

$$\begin{bmatrix} 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 0 & -2 & 1 & 0 & -2 & 0 & -2 & 0 & -2 \\ 1 & -1 & 1 & 0 & -2 & -1 & 0 & 2 & 0 & -2 & 0 & 2 \\ 1 & 1 & -1 & 0 & -2 & -1 & 0 & -2 & 0 & 2 & 0 & -2 \\ 1 & -1 & -1 & 0 & -2 & 1 & 0 & 2 & 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

∎

Proposition 4.2.5: *The functions in $\mathcal{H}[\mathbf{C}]$ are pairwise orthogonal, and the set $\mathcal{H}[\mathbf{C}]$ exactly spans the set of all possible functions over $\mathbf{C}$.*

Proof: First, we are going to show that the basis functions in $\mathcal{H}[\mathbf{C}]$ are pairwise orthogonal, that is, for any two vectors in $\mathcal{H}[\mathbf{C}]$, h and h', $\sum_{\mathbf{c}\in\mathrm{Dom}_{\mathbf{C}}} h(\mathbf{c})h'(\mathbf{c}) = 0$.

Each basis vector in $\mathcal{H}[\mathbf{C}]$ is a product of basis functions for the the individual variables in $\mathbf{C}$. For two such vectors to be orthogonal, we must have

$$\sum_{\mathbf{c}\in\mathrm{Dom}_{\mathbf{C}}} h(\mathbf{c})h'(\mathbf{c}) = \sum_{\mathbf{c}\in\mathrm{Dom}_{\mathbf{C}}} \prod_{V_i\in\mathbf{C}} h^{V_i}(\mathbf{c}) \prod_{V_i\in\mathbf{C}} h'^{V_i}(\mathbf{c}) = 0.$$

A value assignment $\mathbf{c}$ to a cluster $\mathbf{C}$ is defined as an assignment of values to all variables $V_1, V_2, \ldots, V_l$ belonging to $\mathbf{C}$. Recall that we defined a basis function of an individual variable V over $\mathrm{Dom}_{\mathbf{C}}$ as $h_i^V(\mathbf{c}) = h_i^V(V[\mathbf{c}])$. Therefore, we can rewrite the expression above as

$$\sum_{v^1\in\mathrm{Dom}_{V_1}} \sum_{v^2\in\mathrm{Dom}_{V_2}} \cdots \sum_{v^l\in\mathrm{Dom}_{V_l}} \prod_{V_i\in\mathbf{C}} h^{V_i}(\mathbf{c}) \prod_{V_i\in\mathbf{C}} h'^{V_i}(\mathbf{c})$$

The two basis functions h and h' must differ in at least one of the individual variable basis functions used in creating them. That is, there exists at least one i, such that $h^{V_i} \neq h'^{V_i}$. Since we constructed our individual variable vectors to be pairwise orthogonal, it follows that $\sum_{v^i \in \mathrm{Dom}_{V_i}} h^{V_i}(v^i) h'^{V_i}(v^i) = 0$. We can assume without loss of generality that $i = l$ (the expression above can always be rearranged to have the terms depending on i in the innermost sum). This makes the entire expression equal to zero. Therefore, the basis functions in $\mathcal{H}[\mathbf{C}]$ are pairwise orthogonal.

Pairwise orthogonality together with the fact that none of the vectors in $\mathcal{H}[\mathbf{C}]$ is a zero vector imply their linear independence. Finally, linear independence implies that $\mathcal{H}[\mathbf{C}]$ spans $I\!R^k$ where $k = |\prod_{V \in \mathbf{C}} \mathrm{Dom}_V|$ (see e.g., (Strang 1980)). ∎

By taking the union of the bases for the appropriate clusters, we can span any set of factored utility functions.

Corollary 4.2.6: *Let $\mathcal{C}$ be a set of clusters. The set of functions $\mathcal{H}[\mathcal{C}] = \cup_{\mathbf{C} \in \mathcal{C}} \mathcal{H}[\mathbf{C}]$ spans the set of all factored utility functions over $\mathcal{C}$.*

Proof: Follows immediately from Proposition (4.2.5) and Definition (4.2.1). ∎

We can therefore parameterize any factored utility function over $\mathcal{C}$ using a set of coefficients w_i, one for every function in $\mathcal{H}[\mathcal{C}]$. How many parameters are required? For each cluster $\mathbf{C}$, we have $|\mathrm{Dom}_{\mathbf{C}}|$ functions in $\mathcal{H}[\mathbf{C}]$. However, the bases for the different clusters are not disjoint.

Example 4.2.7: Assume that our clusters are $\{A\}$, $\{B, C\}$, and $\{C, D\}$, and that all of our variables are ternary. We have 3 functions in $\mathcal{H}[A]$, and 9 in each of $\mathcal{H}[\{B, C\}]$ and $\mathcal{H}[\{C, D\}]$. However, the h_1 (all 1) function is common to all clusters, and the three h^C functions are common to the two clusters that contain C. Of course, we must be careful not to undercount by double-counting the overlap: h_1 is also among

In general, we can compute the total number of distinct functions in our basis by a simple inclusion-exclusion formula, keeping in mind that the overlap between the bases for two clusters $\mathbf{C}$ and $\mathbf{C}'$ is precisely the basis for $\mathbf{C} \cap \mathbf{C}'$ (taking $\mathcal{H}[\emptyset]$ to be the single vector h_1):

$$\begin{aligned} |\mathcal{H}[\mathcal{C}]| &= \sum_i |\mathcal{H}[\mathbf{C}_i]| - \sum_{i_1 \neq i_2} |\mathcal{H}[\mathbf{C}_{i_1} \cup \mathbf{C}_{i_2}]| \\ &\quad + \sum_{i_1 \neq i_2 \neq i_3} |\mathcal{H}[\mathbf{C}_{i_1} \cup \mathbf{C}_{i_2} \cup \mathbf{C}_{i_3}]| - \cdots \end{aligned}$$

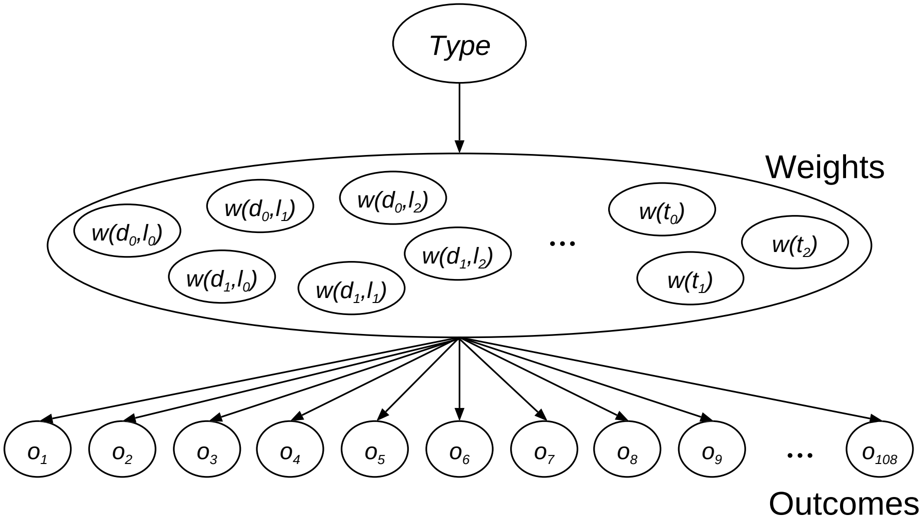
Thus, the total number of basis functions, and thereby of parameters required, grows (at most) linearly with the number of clusters and exponentially with the size of each one.

4.3 The Generative Model

Our approach relies on a few basic assumptions about the population of users whose utility we are trying to model. The first assumption is that the population is composed of several disjoint subpopulations, or types (which we model using a random variable T), where the utility functions of the individuals of each type are statistically similar. Is it justified to postulate the existence of such types? Every person's utility function may be different, both in terms of the numerical utility values assigned to different outcomes and the in terms of the independence properties among the utility attributes.

However, while in principle any utility function is possible, some are much more likely to occur in practice than others. In our utility database for the prenatal diagnosis domain, we can observe some interesting patterns. For example, a few of the patients reported very high utility values for all outcomes, resulting in utility functions which are close to constant. Another group of patients rated all outcomes involving miscarriage significantly lower than the rest. These observations lead us

(a)



(b)

Figure 4.1: (a) A possible decomposition of the utility function in the prenatal diagnosis domain; (b) the corresponding probabilistic model

to believe that a population of patients is likely to be composed of several distinct subpopulations or types. The utility functions in each subpopulation may decompose in a different way. More formally, each subpopulation (each type t) may utilize a different factorization $\mathcal{C}_t$ of the utility function. Every individual is associated with a vector $\mathbf{w}_t$ of dimension $m_t = |\mathcal{H}[\mathcal{C}_t]|$, where each w_j is the coefficient of the jth basis function $h_j \in \mathcal{H}[\mathcal{C}_t]$. The vector $\mathbf{w}_t[j]$ represents the user's subutility functions.

We represent a probabilistic model over utilities by defining a vector random variable $\mathbf{W}_t$. For each value t of T, we represent $P(\mathbf{W}_t \mid t)$ as a multivariate Gaussian with mean vector $\boldsymbol{\mu}_t$ and covariance matrix Σ_t. We assume that individuals in the subpopulation are IID samples from the $P(\{\mathbf{W}_t\}_t \mid T)$ distribution.

An individual's subutility vector $\mathbf{w}_t$ defines a complete utility function, which ought to be $\mathbf{u}^* = A_t\mathbf{w}_t$. However, the utility elicitation process can be quite noisy. We accommodate for that problem by assuming that the user's actual utility vector $\mathbf{u}$ is modified by some white noise, i.e., for each o, we have that u_o is u_o^* plus some random white noise ϵ_t sampled from a zero-mean Gaussian distribution with some variance σ_t^2. More formally, we have a vector random variable $\mathbf{U}$ of dimension n, which is a linear Gaussian whose mean is $A_t\mathbf{W}_t$ and whose variance is $\sigma_t^2 I$ where I is the unit matrix.

Note that, for each type t, the distribution over $(\mathbf{W}_t, \mathbf{U})$ is a simple multivariate Gaussian, defined using a Gaussian distribution over $\mathbf{W}_t$ and a conditional linear Gaussian for $\mathbf{U}$ given $\mathbf{W}_t$. However, the distribution as a whole is not exactly a mixture of linear Gaussians, as the dimension of the vector $\mathbf{w}_t$ can vary for the different types.

4.4 Estimating a Factored Utility Function

A model such as this can be used for several purposes. The most basic use is to compute the most probable factored utility function for a given user. More precisely, assume we are given a vector $\mathbf{u}$ representing the full utility function elicited from a certain user. Our goal is to compute the type t and vector $\mathbf{w}_t$ such that the probability $P(\mathbf{w}_t \mid \mathbf{u}, t)$ is maximized. We perform a separate computation for each t.

From the definition of our generative model, we have that:

$$P(\mathbf{w}_t \mid \mathbf{u}, t) = \frac{P(\mathbf{u} \mid \mathbf{w}_t)P(\mathbf{w}_t \mid t)}{P(\mathbf{u} \mid t)}.$$

The denominator is a constant, so it does not affect the choice of maximum.

$$(\prod_o P(u_o \mid \mathbf{W}_t)) \cdot P(\mathbf{W}_t \mid t).$$

Maximizing this function is equivalent to minimizing an error function corresponding to its negative logarithm (see e.g. (Bishop 1995)):

$$-\sum_o \ln P(u_o \mid \mathbf{w}_t) - \ln P(\mathbf{w}_t \mid t).$$

Recall that the distribution of each variable U_o is given as:

$$P(U_o \mid \mathbf{w}_t) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{((A_t)_o\mathbf{w}_t) - U_o)^2}{2\sigma_t^2}\right)$$

where $(A_t)_o$ is the row of the matrix A_t that corresponds to the outcome o. Thus, the first term in our error function (for the given vector $\mathbf{u}$) can be simplified to

$$-\frac{1}{2\sigma_t^2}\sum_o((A_t)_o\mathbf{w}_t - u_o)^2 + n\ln\sigma_t + \frac{n}{2}\ln(2\pi) \qquad (4.1)$$

The conditional distribution of the $\mathbf{W}_t$ variables is expressed as

$$P(\mathbf{W}_t \mid t) = \frac{1}{(2\pi)^{m_t/2}|\Sigma_t|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{W}_t - \boldsymbol{\mu}_t)^T\Sigma_t^{-1}(\mathbf{W}_t - \boldsymbol{\mu}_t)\right),$$

and its negative logarithm is

$$\frac{m_t}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma_t| + \frac{1}{2}(\mathbf{W}_t - \boldsymbol{\mu}_t)^T\Sigma_t^{-1}(\mathbf{W}_t - \boldsymbol{\mu}_t). \qquad (4.2)$$

If we put together (4.1) and (4.2), and eliminate terms that do not depend on $\mathbf{w}_t$ (and therefore do not affect the choice of minimizing value), we get as our final error function:

$$E(\mathbf{w}_t) \quad = \quad \frac{1}{2\sigma_t^2}\sum_o(A_t(o)\mathbf{w}_t - u_o)^2 + \frac{1}{2}(\mathbf{w}_t - \boldsymbol{\mu}_t)^T\Sigma_t^{-1}(\mathbf{w}_t - \boldsymbol{\mu}_t)$$

$$= \frac{1}{2\sigma_t^2}\|A_t\mathbf{w}_t - \mathbf{u}\|^2 + \frac{1}{2}\|B_t\mathbf{w}_t - B_t\boldsymbol{\mu}_t\|^2$$

where $B_t^T B_t = \Sigma_t^{-1}$. (We are guaranteed that such a decomposition exists because the covariance matrix of a Gaussian is guaranteed to be positive definite.)

Thus, maximizing the posterior probability of the vector $\mathbf{w}_t$ is equivalent to minimizing a squared-error function. Let D_t be the $(n + m_t) \times m_t$ matrix obtained by concatenating the matrices $\frac{1}{\sigma_t}A_t$ and B_t. We also define a vector $\mathbf{u}'$ of length $n + m_t$ defined by concatenating $\frac{1}{\sigma_t}\mathbf{u}$ and $B_t\boldsymbol{\mu}_t$.

Our construction was designed especially to make this least-squares computation efficient. Recall that, by construction, the columns of A_t are orthogonal. The columns of D_t, while no longer necessarily orthogonal, remain linearly independent. Thus, we can compute the optimal solution to the least-squares problem by projection (Strang 1980):

$$\begin{aligned} \hat{\mathbf{w}}_t &= (D_t^T D_t)^{-1} D_t^T \mathbf{u}' \\ &= (\frac{1}{\sigma_t^2} A_t^T A_t + B_t^T B_t) D_t^T \mathbf{u}' \\ &= (\frac{1}{\sigma_t^2}\Lambda + \Sigma_t^{-1})^{-1} D_t^T \mathbf{u}' \end{aligned}$$

where Λ is a diagonal matrix whose ith diagonal element is $\|h_i\|^2 = \sum_o h_i(o)^2$. Note that the matrix $(\frac{1}{\sigma_t^2}\Lambda + \Sigma_t^{-1})^{-1} D_t^T$ does not depend on $\mathbf{u}$, and can therefore be computed once and reused for every individual for whom we want to estimate $\mathbf{w}_t$. If Σ_t is a diagonal matrix, our choice of basis makes the computation very efficient.

This computation gives us, for each type t, the most likely vector $\mathbf{w}_t$ for the user given that he is in class t. We can now easily compute the most likely pair $(t, \mathbf{w}_t)$ for this user.

This model can also be used to give us more information. Recall that the conditional distribution on $\mathbf{W}_t, \mathbf{U}$ is a multivariate Gaussian distribution. At the cost of the vector $\hat{\mathbf{w}}_t$ computed above. The covariance matrix of the distribution could be used as an indicator for how confident we are in our estimate $\hat{\mathbf{w}}_t$. Clearly, there are situations where this information can be quite important, but it is not clear that it is always worth the computational overhead. On the other hand, unlike projection, this technique can be used even if some of the values in the original utility vector are missing.

4.5 Benefits of Probabilistic Modeling

4.5.1 Using Population Models

Our probabilistic model of utility functions in the population allows us to represent uncertainty over numerical utility values and uncertainty over the function structure at the same time. We can use such a model in a variety of ways. It can give us insight into a given population's preferences, the number and character of distinct types represented in that population and the range of utilities we are likely to encounter.

We can also use the model to develop behavioral guidelines. For example, in medical domains, there is a need for general clinical practice guidelines to help doctors schedule screening tests, choose among treatment options, etc. An example of such a guideline is a rule that pregnant women over 35 years of age should be given an opportunity to attend genetic counseling sessions and offered tests to diagnose chromosomal abnormalities in the fetus, such as Down's syndrome. (Women younger than 35 are considered to be at too low a risk for chromosomal abnormalities to warrant the expense.) Some recent work (Sanders 1998) considered developing such guidelines automatically from probabilistic models of the domain. It has been pointed out, however (e.g., in the context of the 35 year old cutoff guideline described above (Kuppermann, Goldberg, Nease Jr., and Washington 1999)), that developing such guidelines without a thorough study of the utilities for the problem outcomes in the evaluation metric is the average difference between the expected utility resulting from following the optimal strategy and the expected utility resulting from the strategy considered to be optimal according to the guideline. Our population model allows us to estimate the expectation of this difference (see Section 7.3).

We could also use the model directly in developing the guidelines. Instead of computing the optimal strategy for an arbitrary utility function, we could compute optimal strategies for the means of each of our subpopulations. We could also compute the probability of encountering a "strange" utility function — one far away from the means of our mixture components — for which a general guideline is likely to be suboptimal.

4.5.2 Adapting the Model to a Particular User

In some domains, the risks involved are too great to use a general guideline — we need a utility model specific to a particular user. Even in such cases, our population model can be very useful.

When we encounter a new user, we can use the population model as our prior over the new user's utility function. Whenever we acquire some new information about the user, we can incorporate it into the model by conditioning on it to obtain a more informed posterior distribution. As our knowledge about the new user grows, our belief about his or her utility function should exhibit smaller variance and favor one of the types more and more. At every step, we can compute a point estimate of the user's utility function and our confidence in that estimate.

The process of adapting the population model to a particular user is described in detail in Chapter 6. In that chapter we show how to condition the model on various types of information, from constraints on utilities of individual outcomes obtained in the process of utility elicitation to observations of the user's behavior.

many ways to use the distribution over utility functions to facilitate utility elicitation and improve the quality of the results.

The most obvious is simply to use the model as a guide to the range of utility functions within the population. In particular, our model incorporates a built-in measure of confidence. When we assess a new user's utility function, we can quickly discover if he or she is an "outlier" — a person with an atypical utility function. We can ask such a person additional questions to make sure that there was no error in the process.

A somewhat deeper use of the model, along the same lines, is for smoothing the results of the utility elicitation process for a particular individual based on trends in the population as a whole. Given the amount of noise in the utility elicitation process, the mean of our posterior distribution (population prior conditioned on the user's answers to the utility elicitation questions) may be actually closer to the user's true utility function than the utility function obtained in the elicitation process. Smoothing of this type is likely to be very useful in getting robust utility estimates.

We can also use the model in a much more fundamental way to change the entire utility elicitation process in cases where the utility function can be assumed to exhibit some form of additive independence. For (conditionally) additive decompositions, Keeney and Raiffa (1976) describe a utility elicitation procedure which exploits the structure to reduce the number of questions asked. A separate scale is established for every utility function component and the user is asked a series of questions about its parameters. At the end, a new set of assessments must be made to discover the scaling constants (see Section 3.4.6). This procedure has become a gold standard in many applications.

This method cannot take advantage of the more generalized factorizations (e.g., generalized additive independence). An alternative procedure, general enough to handle all factorizations, would ask questions about full outcomes. When we assess the utility function of a new user, we only need to ask as many questions as the number of all the subutility functions are known, we can compute the utility values for the remaining outcomes. It would be good practice to double check that the chosen decomposition really matches the new user's utility function structure by asking a few more "redundant" questions and comparing the answers with those predicted by the function we had computed.

This procedure can also be modified to utilize the model in a more principled way. We can view the utilities elicited for different outcomes as evidence in the distribution defined by the model. We can then use standard probabilistic inference to compute the distribution over the user's subutility functions. The more utilities we elicit, the more evidence we have, the more certain we are about the actual value of the user's subutility functions. We can apply techniques such as conditional mutual information or variance reduction to decide, at each point in time, which utility elicitation question is likely to be the most informative about the subutility variables. We can also make principled decisions on when to stop the elicitation process by considering our uncertainty about these variables.

Finally, we can use probabilistic models of the utility function as the basis for a more targeted process of utility elicitation. In a given decision making task, the utilities of different outcomes typically influence the decision, and the resulting expected utility, to radically different extents. Most simply, some outcomes may have very low probability in the current setting, so their utility is largely irrelevant. Having a distribution over the utility functions in the population, we can compute the value of information of every elicitation question; we can then focus our efforts on those that have the highest impact on our actual decision (see Chapter 7).

4.5.4 Choosing Optimal Decisions given Utility Uncertainty

Even with a lot of information about the user's utility function, some uncertainty is likely to remain. Therefore, an important question we must address is how to make

$$\mathrm{EU}_\pi(\mathbf{u}) = \sum_{o \in \mathbf{O}} P(o \mid \pi) u_o.$$

The expected utility under $p(\mathbf{U})$ can easily be shown to be

$$\begin{aligned}
\mathrm{EU}_\pi(p) &= \int p(\mathbf{u}) \mathrm{EU}_\pi(\mathbf{u}) d\mathbf{u} \\
&= \sum_{o \in \mathbf{O}} \int P(o \mid \pi) p(u_o) u_o du_o \\
&= \sum_o P(o \mid \pi) \int p(u_o) u_o du_o \\
&= \sum_o P(o \mid \pi) \mathcal{E}_p[u_o].
\end{aligned}$$

That is, the expected utility of any strategy π, with uncertainty $p(\mathbf{U})$ over the user's utility function, is equal to its expected utility using the mean of $\mathbf{U}$ under p.

Hence, we can find the best strategy given $p(\mathbf{U})$ by running the expectimax algorithm (see Section 2.1.2) on our decision tree using the mean of U_o under p as the utility value for outcome o. In general, we can compute the mean of U_o under p by Monte Carlo sampling. However, under the assumption that p is a mixture of Gaussians, we can compute it much more efficiently (see Section 6.4).

This ability to determine the optimal decision given our current beliefs about a user's utility function will be useful both in the context of a decision support system (Chapter 7) and in the context of a game (Chapter 8).

4.5.5 Predicting Another Agent's Actions

The population model of utilities can also be used to predict future actions of one's opponent in a competitive setting, such as described in our bookseller example in Section 1.5. Such a model, conditioned on any information we may have gathered about our specific opponent (e.g., his past decisions) allows us to compute not only his expected course of action in a future situation, but also a distribution over possible

Chapter 5

Learning the Distribution

In Chapter 4, we defined a statistical model of utilities in a population of users, and showed how it can be used to compute a factorization of an elicited utility function. We also discussed various benefits of having such a model at our disposal (Section 4.5). A utility model can be learned from data in the same way we can learn probabilistic models involving state variables. In this chapter, we define a learning framework for acquiring such a statistical model.

We assume that we have a database of fully or partially elicited utility functions at our disposal. These utility functions should come from persons randomly selected from our target population. Our goal is to estimate the density function over utilities and discover the structural properties of the utility functions in the population at the same time.

Even if the utility function is factored, the utility elicitation process is typically done in terms of utilities of full outcomes. This is certainly the case if, as we assumed, the factorization of the utility function is unknown in advance. Thus, we assume that the training data we are given is a set of utility vectors $\mathbf{u}[j]$, one for each individual. We allow for cases where some of the components of the utility vectors may be missing. The type variable T and the corresponding decomposed utility vector $\mathbf{W}_t$ are unobserved in the training data.

cannot use full Bayesian estimation in the presence of partially observable data, it will nevertheless be useful to view the model parameters as having a prior and a posterior. This perspective will be useful both for smoothing our numerical estimates and to provide the appropriate bias for the structure selection task.

Suppose that, for every value t of the variable T, we have an m_t dimensional multivariate Gaussian with an unknown mean vector $\boldsymbol{\mu}_t$ and an unknown covariance matrix Σ_t. We need to find an appropriate *conjugate prior* over $\boldsymbol{\mu}_t$ and Σ_t.

Definition 5.1.1: Consider a parametric model $P(\mathbf{X} \mid \psi)$ which defines a distribution for data instances given some set of parameters ψ. Let $P(\psi \mid \alpha)$ be a parametric form for a prior over ψ, parameterized by *hyperparameters* α. We say that $P(\mathbf{X} \mid \psi)$ and $P(\psi \mid \alpha)$ are a *conjugate family* if the *posterior parameter distribution* $P(\psi \mid \alpha, \mathbf{x}) \propto P(\mathbf{x} \mid \psi)P(\psi \mid \alpha)$ has the same parametric form as our parameter prior $P(\psi \mid \alpha)$, albeit with different hyperparameters. We also say that $P(\psi \mid \alpha)$ is the *conjugate prior* for $P(\mathbf{X} \mid \psi)$. ∎

An appropriate conjugate prior over $\boldsymbol{\mu}_t$ and Σ_t is the *Normal-Wishart* (DeGroot 1970). We use a Normal-Wishart prior for the parameters of each of the type-specific Gaussian distributions over $\mathbf{W}_t$ (one for each type t) and for the parameters of the conditional Gaussian over the U_o given $U^*(o) = A_t(o)\mathbf{W}_t$. We assume that the parameters θ_t representing the prior probability $P(T = t)$ are distributed with a Dirichlet distribution.

The main problem is that our data is only partially observable, rendering full Bayesian estimation infeasible. We therefore resort to finding the MAP parameter estimate using the *expectation-maximization* (EM) algorithm (Dempster, Laird, and Rubin 1977). The algorithm begins by choosing some (possibly random) initial configuration of the parameters. It then proceeds to iterate the following two steps:

- **E step:** We use our parameter prior to define a Gaussian prior distribution over $\mathbf{W}_t, \mathbf{U}$. For each instance j and each type t, we condition this distribution on $\mathbf{u}[j]$, and obtain a Gaussian posterior over the hidden variables its possible values, each of which gets a fraction of the weight of the sample. We use these Gaussian posterior distributions to compute *expected sufficient statistics*: the expected empirical means and expected empirical covariances.

- **M step:** The expected sufficient statistics are used to update the Wishart priors, which then generate a new Gaussian prior distribution over $\mathbf{W}_t, \mathbf{U}$. A similar update is done to the Dirichlet distribution over the types.

The process iterates until convergence. EM is guaranteed to converge to a point where the gradient of the log-likelihood function is zero. (Recall that the likelihood function is the probability of the data given the model.) In theory, this can occur at local maxima, local minima, or saddlepoints. In practice, convergence only happens at local maxima.

The procedure for the maximum likelihood variant of EM is identical, except that the prior is not used. We provide a precise description of the EM computation below.

5.1.1 Parameter Prior

When applying EM to our model, the parameters to be estimated are $\theta_t, \boldsymbol{\mu}_t, \Sigma_t$ and σ_t^2 for every t. The hidden variables are T and $\mathbf{W}_t$.

In our setting, we assume that the parameters $\boldsymbol{\mu}_t, \Sigma_t$ of $P(\mathbf{W}_t \mid t)$ are distributed Normal-Wishart with parameters $(R_t^0, \beta_t^0, \lambda_t^0, \nu_t^0)$. We also assume that the variance σ_t^2 associated with all of the variables U_o is distributed one-dimensional Wishart with parameters ρ_t^0, γ_t^0 and η_t^0. ρ_t, γ_t and η_t correspond to R_t, β_t and ν_t in the distribution over $\mathbf{W}_t$.

A Normal-Wishart prior defines a distribution over the mean $\boldsymbol{\mu}_t$ and covariance matrix Σ_t of a Normal distribution. It is parameterized by: a *precision matrix* R_t; a number $\beta_t > m_t - 1$; a mean vector λ_t; and a number $\nu_t > 0$. Essentially, R_t and β_t define a Wishart distribution $w(R_t, \beta_t)$ over $m_t \times m_t$ matrices Q_t. The conditional distribution of $\boldsymbol{\mu}_t$ given Q_t is a Gaussian with mean λ_t and covariance $\nu_t Q_t^{-1}$. The other words, if we have a Normal-Wishart prior $(R_t^0, \beta_t^0, \lambda_t^0, \nu_t^0)$, and we observe vectors $\mathbf{y}[1], \ldots, \mathbf{y}[\ell]$ sampled from the associated Gaussian $N(\boldsymbol{\mu}_t, \nu_t Q_t^{-1})$, then the posterior distribution over the parameters is also Normal-Wishart, with the following update rule:

$$\begin{aligned}
\bar{\mathbf{y}} &= \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{y}[j] && (5.1)\\
\lambda_t &= \frac{\nu_t^0\lambda_t^0 + \ell\bar{\mathbf{y}}}{\nu_t^0 + \ell} && (5.2)\\
\nu_t &= \nu_t^0 + \ell && (5.3)\\
S_t &= \sum_{j=1}^{\ell}(\mathbf{y}[j] - \bar{\mathbf{y}})(\mathbf{y}[j] - \bar{\mathbf{y}})^T && (5.4)\\
R_t &= R_t^0 + S_t + \frac{\nu_t^0\ell}{\nu_t^0 + \ell}(\lambda_t^0 - \bar{\mathbf{y}})(\lambda_t^0 - \bar{\mathbf{y}})^T && (5.5)\\
\beta_t &= \beta_t^0 + \ell && (5.6)
\end{aligned}$$

The update rules for ρ_t, γ_t and η_t correspond to those for R_t(5.5), β_t (5.6) and ν_t (5.3), respectively.

5.1.2 Data Completion

In order to complete the data, we must compute $P(T[j], \mathbf{W}_t[j] \mid \mathbf{u}[j], \mathit{params})$. First we need to marginalize the parameter prior and obtain a distribution over the domain variables only. Given a Normal-Wishart parameter distribution $(R_t, \beta_t, \lambda_t, \nu_t)$, the distribution over $\mathbf{W}_t$ given t is an n dimensional t *distribution*, which can be approximated using a multivariate Gaussian (DeGroot 1970). For the type-specific distributions, we get:

$$\begin{aligned}
\boldsymbol{\mu}_t &= \boldsymbol{\lambda}_t\\
\Sigma_t &= \frac{\nu_t + 1}{\nu_t \cdot (\beta_t - m_t - 1)} R_t
\end{aligned}$$

The marginalization for a Dirichlet distribution over the type, with hyperparameters α_t, is the standard one:

$$\theta_t = \frac{\alpha_t}{\sum_{t'} \alpha_{t'}}.$$

The result is a Gaussian distribution $P(\mathbf{W}_t, \mathbf{U} \mid t)$. For each t, we compute $P(\mathbf{W}_t \mid t, \mathbf{u}[j])$ and the marginal $P(\mathbf{u}[j] \mid t)$. We also compute the posterior probability of the different types as $P(t \mid \mathbf{u}[j]) \propto P(t) \cdot P(\mathbf{u}[j] \mid t)$.

5.1.3 Expected Sufficient Statistics

Using these probabilities, we can easily compute the (expected) sufficient statistics required for the update of our various parameter priors. For the Dirichlet, we merely need the expected count $\bar{N}(t) = \sum_j P(t \mid \mathbf{u}[j])$. For the various type specific Gaussians, we must compute the expected value of λ_t and S_t. Instead of finding the most likely values for our hidden variables, we take into account the entire posterior distribution. Each possible value gets a fraction of the weight of the sample. Intuitively, it amounts to computing the expectation over uncountably many "completed" data cases — a continuum of possible completions for each j. Fortunately, this turns out to be easy. The key is that the posterior distribution over $\mathbf{W}_t[j]$ given t and $\mathbf{u}[j]$ is a multivariate Gaussian with mean $\boldsymbol{\mu}_t[j]$ and covariance $\Sigma_t[j]$. ($P(\mathbf{W}_t[j] \mid t, \mathbf{u}[j])$ is computed by standard probabilistic inference.) Let $\pi_t[j]$ denote $P(t \mid \mathbf{u}[j])$; intuitively $\pi_t[j]$ is the extent to which the jth sample belongs to type t, and therefore the extent to which it influences the estimate of its parameters. It is straightforward to verify that

$$\begin{aligned}
\bar{\ell}_t &= \sum_{j=1}^{\ell} \pi_t[j] \\
\bar{\mathbf{y}}_t &= \frac{1}{\bar{\ell}_t} \sum_{j=1}^{\ell} \pi_t[j] \boldsymbol{\mu}_t[j]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{\ell} \pi_t[j] \int (\mathbf{w}_t[j] - \bar{\mathbf{y}}_t)(\mathbf{w}_t[j] - \bar{\mathbf{y}}_t)^T P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \\
&= \sum_{j=1}^{\ell} \pi_t[j] \int ((\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j]) + (\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t))((\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j]) + (\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t))^T \\
&\qquad\qquad P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \\
&= \sum_{j=1}^{\ell} \pi_t[j] \Big(\int (\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)(\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)^T P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \\
&\qquad\qquad + \int (\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])(\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])^T P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \\
&\qquad\qquad + \int (\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)(\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])^T P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \\
&\qquad\qquad + \int (\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])(\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)^T P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] \Big) \\
&= \sum_{j=1}^{\ell} \pi_t[j] \Big((\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)(\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)^T \int P(\mathbf{w}_t[j] \mid t, \mathbf{u}[j]) d\mathbf{w}_t[j] + \Sigma_t[j] \\
&\qquad\qquad + (\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)\mathcal{E}\left[(\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])^T\right] + \mathcal{E}\left[(\mathbf{w}_t[j] - \boldsymbol{\mu}_t[j])\right](\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)^T \Big) \\
&= \sum_{j=1}^{\ell} \pi_t[j] \Big((\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)(\boldsymbol{\mu}_t[j] - \bar{\mathbf{y}}_t)^T + \Sigma_t[j] \Big)
\end{aligned}
$$

Finally, we must compute the expected empirical variance $\bar{s}_t$ needed to update ρ_t and in turn σ_t^2. Simple linear algebra shows that, if $\mathbf{W}_t$ is distributed Gaussian with mean $\boldsymbol{\mu}_t[j]$ and variance $\Sigma_t[j]$, then $U^* = A_t\mathbf{W}_t$ is distributed Gaussian with mean $A_t\boldsymbol{\mu}_t[j]$ and variance $\Upsilon_t[j] = A_t\Sigma_t[j]A_t^T$. Thus, we get that

$$
\bar{s}_t = \sum_{j=1}^{\ell} \pi_t[j] \sum_o (\Upsilon_t(o,o)[j] + ((A_t\boldsymbol{\mu}_t[j])_o - u_o)^2)
$$

and

$$
\rho_t = \rho_t^0 + \bar{s}_t + \frac{\eta_t^0 n\ell}{\eta_t^0 + n\ell} \cdot \sum_o \sum_{j=1}^{\ell} \pi_t[j]((A_t\boldsymbol{\mu}_t[j])_o - \bar{u}_o)^2.
$$

the mean of U_o^* and the observed utility for outcome o, and a contribution for the inherent variance of U_o^*.

We can now use these expected sufficient statistics in place of the exact sufficient statistics in Equations (5.2), (5.3), (5.5) and (5.6). This gives us new estimates of the posterior over the parameters relative to the completed data. We then marginalize the posterior to induce a new Gaussian prior distribution over $\mathbf{W}_t, \mathbf{U}$, and continue.

5.2 Model Selection

Now, we consider the problem of finding a good structure. We focus on the problem of discovering the structure of the subutility functions within the clusters, and assume the number of clusters is given.[1] We apply Bayesian model selection to this task. More precisely, we define a discrete variable S whose states s correspond to possible models, i.e., possible decompositions of the subutilities in the different clusters; we encode our uncertainty about S with the probability distribution $P(s)$. For each model s, we define a continuous vector-valued variable Ψ_s, whose instantiations ψ_s correspond to the possible parameters of the model. We encode our uncertainty about Ψ_s with a probability density function $P(\psi_s \mid s)$, as described above.

5.2.1 Model Score

We score the candidate models by evaluating the *marginal likelihood* of the data set D given the model s (Heckerman 1996). That is, we want to compute the

$$P(D \mid s) = \int P(D \mid \psi_s, s) P(\psi_s \mid s) P(s) d\psi_s.$$

[1] Our techniques easily extend to the more standard problem of discovering the number of clusters (Cheeseman and Stutz 1996). Assuming that the number of clusters k can range from 1 to some fixed K, we simply run a separate search procedure for each k. Then, we compare the best models in each run and pick the best one overall.

puted efficiently for complete data. If D_c is any completion of the data set D, we have

$$P(D \mid s) = P(D_c \mid s) \frac{\int P(D, \psi_s \mid s) d\psi_s}{\int P(D_c, \psi_s \mid s) d\psi_s}.$$

Letting $\tilde{\psi}_s$ be either an MAP or an ML estimate for ψ_s, we can apply the BIC/MDL approximation[2] to the numerator and denominator, and get

$$\log P(D \mid s) \approx \log P(D_c \mid s) + \log P(D \mid \tilde{\psi}_s, s) - \log P(D_c \mid \tilde{\psi}_s, s).$$

(In our case, the model s and its parameterization $\tilde{\psi}_s$ are the same for the complete data and the actual data, so the model complexity term cancels out.) We can estimate the last two terms in this expression by running our EM algorithm from the previous section. Chickering and Heckerman (1996) showed that this approximation is surprisingly accurate, much more so than a direct use of BIC/MDL.

The first term, $P(D_c \mid s)$, is the probability of a complete data set, where the distribution of the continuous variables in the network, conditioned on each instantiation of the discrete variable *Type*, is a multivariate normal distribution. Geiger and Heckerman (1994) show that, in the case of complete data, the marginal likelihood has a closed form that decomposes (as usual) as a product over separate families in the model (the superscript denotes the part of the sample corresponding to the appropriate variable(s)):

$$P(D_c|s) \quad = \quad P(D_c^T|s) \cdot \prod_t P(D_c^{\mathbf{W}_t}|t, s) P(D_c^{\mathbf{U}}|t, \mathbf{W}_t, s)$$

---

[2]BIC, or *Bayesian information criterion* (Schwarz 1978) is defined as

$$\log P(D \mid s) \approx \log P(D \mid \tilde{\psi}_s, s) - \frac{d}{2} \log \ell$$

where d is the number of parameters in $\tilde{\psi}_s$ and ℓ is the number of samples in D. The BIC approximation is exactly minus the *minimum description length* (MDL) criterion described by (Rissanen 1987).

$$\prod_t \qquad (\nu_t^0 + \ell) \qquad c(m_t, \beta_t^0 + \ell)$$

$$\prod_t (2\pi)^{-\frac{n\ell}{2}} \left(\frac{\eta_t^0}{\eta_t^0 + n\ell}\right)^{\frac{1}{2}} \cdot \frac{c(1, \gamma_t^0)}{c(1, \gamma_t^0 + n\ell)} |\rho_t^0|^{\frac{\gamma_t^0}{2}} |\rho_t|^{-\frac{\gamma_t^0 + n\ell}{2}}$$

where $c(i, \alpha) = \left(2^{\alpha i/2} \pi^{i(i-1)/4} \prod_{j=1}^{i} \Gamma\left(\frac{\alpha+1-j}{2}\right)\right)^{-1}$.

5.2.2 Search over Models

Given a scoring function, we can apply standard techniques for finding a high-scoring structure. We use a greedy hill-climbing search with random restarts. Our search space operators modify the subutility structure of each type separately. An operator can add a variable to an existing subutility function, delete a variable from a function, or introduce a new subutility function with a single variable. We evaluate each candidate successor structure by running EM on it, and then scoring it using the Cheeseman-Stutz approximation to the Bayesian score.

Unfortunately, our algorithm has a high computational cost. In each search step, we evaluate all successor models to our current model s. Given a set of utility attributes $\mathbf{V}$, there are $O(|\mathbf{V}|^2)$ such models. For each successor model s', we run EM to find an appropriate parameterization $\tilde{\psi}_{s'}$. Note that each iteration of EM requires that we run full Bayesian network inference on each data case.

This problem is common to all Bayesian network learning with partially observable data. An alternative approach called *structural EM* (Friedman 1997) can be used in some cases to speed up the search, but we cannot apply it here. In structural EM, the completion of the data computed in the context of the current model s is used to parameterize and score successor models. However, our successor models contain different subutility variables than the current network s, so we cannot use the completion computed in the context of s to parameterize them.

The algorithm is tractable only for very small domains. Fortunately, the size of

fully connected	463	200	500
2 clusters			
additive/additive	463	350	500
additive/conditionally additive	500	500	500
conditionally additive/conditionally additive	550	500	750
conditionally additive/fully connected	3750	2500	5000
fully connected/fully connected	4000	3000	5000

Table 5.1: Number of samples needed to recover the structure

the attribute set for a utility function (unlike the size of the outcome space) cannot be very large. In our prenatal diagnosis domain (Section 1.4), we have 5 utility attributes. This domain is considered to be unusually complex. It is rare to encounter a domain with a much larger number of attributes.

5.3 Experiments

We tested our approach on both real and synthetically generated data.

5.3.1 Synthetic Data Results

In our artificial domain, we had 3 utility attributes, one ternary and two binary, and 12 outcomes. We had three basic structures: fully additive; structured, in which $u(o) = u_1(V_1, V_2) + u_2(V_2, V_3)$; and fully connected (no independencies). We generated 10–20 distributions for each structure, using different parameters.

In one cluster tests, we were always able to recover the structure of the original distribution. For the additive model, the correct structure was chosen after seeing at most 2 data points. (This result was to be expected given the well-known bias towards simpler structures in Bayesian learning.) For the structured model, the number of samples needed ranged from 100 to 750. For the fully connected model, we needed

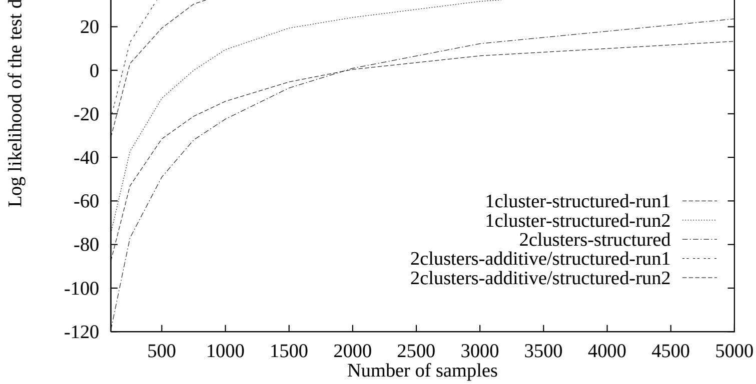


Figure 5.1: Learning curves for several models

200-500 samples.

In two-cluster tests, small amounts of data (10–100 samples) always resulted in a model with one fully connected and one fully additive structure, regardless of the underlying distribution. Given more data (350–5000), we were able to learn either the correct structure or one differing by only one variable's presence or absence in a subutility function. We obtained these results for models with the same as well as with differing decompositions in the different clusters. Table 5.1 summarizes these results.

We also tested our algorithm as a density estimator. For these tests, we used a domain with 4 attributes, one ternary and three binary. We had two structures: one fully additive and one structured in which $u(o) = u_1(V_1, V_2) + u_2(V_2, V_3) + u_3(V_2, V_4)$. We created several 1- and 2-cluster models, with the same decomposition in different clusters in some models and different decompositions in other models. The learning curve tests are presented in Figure 5.1. As the number of samples grows, the learned parameters generally seem to converge to the generating distribution.

Finally, we tested the smoothing effect of using parameter priors in our algorithm. We randomly selected parameters for several 1- and 2-cluster structured (conditionally additive) models with 4 utility attributes. From each model, we sampled a large

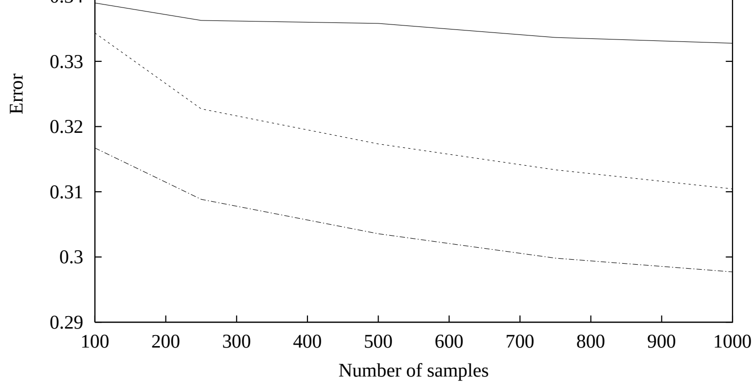


Figure 5.2: Least-squares projection vs. MAP projection

number of the utility functions. We divided our data into a training set and a test set. We varied the size of the training set to examine the dependence of the error on the number of samples. Instead of searching over possible models, we used the correct number of clusters and the correct utility function structure (i.e., the structure our data was generated from) in our learning procedure. After learning the parameters of the model based on the training set, we computed the values of the weight vector **w** using least-squares projection and MAP projection (as described in Section 4.4) for the samples in our test set. We compared these values to the true weights **w** used to generate these samples. Figure 5.2 shows the results on 1- and 2-cluster models. The upper curve in both cases corresponds to the least-squares projection, the lower to MAP projection. The error for MAP projection is not only lower, it also decreases more rapidly.

5.3.2 Utility Data for the Prenatal Diagnosis Domain

Our dataset consists of utility functions elicited in a prenatal diagnosis study performed by (Kuppermann, Shiboski, Feeny, Elkin, and Washington 1997). All study

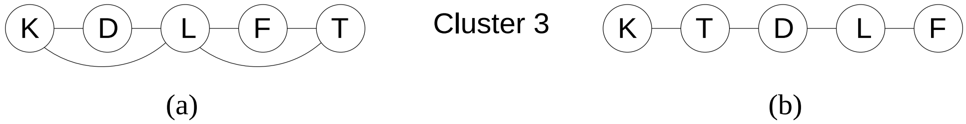


Figure 5.3: Best decomposition for (a) Visual Analog Scale and (b) Standard Gamble

subjects were recruited at the University of California at San Francisco (UCSF) Prenatal Diagnosis Center. Study subjects were recruited from a counseling session for women who had not yet decided which prenatal diagnostic test to undergo, or, in some cases, whether to undergo prenatal diagnosis at all. The decision problem for prenatal diagnosis is described in Section 1.4.

Out of 70 subjects, we selected 51 who completed the entire interview, which involved assessing utilities for 22 outcomes using two elicitation methods: standard gamble (SG) and visual analog scale (VAS). These two methods are known to produce very different utility values (see Section 3.4), so we treated the two sets of utilities as two distinct databases. We treated the values of all the outcomes the women were not asked about as missing.

5.3.3 Real Data Results

We searched the space of 1-, 2- and 3-cluster models. The best models we learned for our two databases were in both cases 3-cluster models. They are presented in Figure 5.3. The nodes correspond to utility attributes in our domain: testing (T), Down's status (D), pregnancy loss (L), knowledge (K) and future pregnancy (F). Additive and conditional additive independence correspond to vertex separation. While the size of the database does not allow us to treat our models as representing the true structure of the utility functions in the population, some of the correlations found are very interesting. For example, the correlation between the utilities for pregnancy loss and utilities for Down's status and future pregnancy are highly intuitive. Losing than losing a healthy one — many women decide to abort a planned pregnancy upon learning about the presence of chromosomal abnormalities in the fetus.

We note that, in both cases, structures having multiple clusters received substantially higher scores than structures having a single cluster. At the first glance, this may suggest the danger of overfitting. However, our utility models were highly structured. The best model for the SG database had 3 clusters with the sizes of the weight vectors 11, 13 and 19 respectively. The best model for the VAS database also had 3 clusters albeit with slightly different decompositions. The sizes of the weight vectors were 9, 11 and 31. Note that the total number of parameters needed in both cases is smaller than the number of parameters required for a 1-cluster fully connected model. Most likely, the danger of overfitting is mitigated by the use of the Bayesian score.

Finally, structures where the different clusters had different decompositions scored more highly than structures where all clusters used the same decomposition. This supports our hypothesis that different subpopulations exist, and have different decompositions.

Chapter 6

Conditioning on New Information

A distribution over utility functions in the population provides us with insight into the utility function structure and can help in the analysis of the decision problem for which the utilities were elicited. There are many contexts, however, in which it is insufficient. For example, consider the situation in which we want to make decision recommendations. In that case, we need to have a model representing our beliefs about an individual user's preferences.

In some cases, we can use the population model as a starting point. It is justified if we can regard the individual user as a randomly selected instance from the same population from which we acquired the training data used to learn the population model. In other words, the population model can be an appropriate *prior* distribution. We can use it to reason about users about whose preferences we have no additional information. As soon as such information becomes available, we have to adjust the model by *conditioning* it on the new information to obtain a more informed *posterior* distribution. In this chapter we consider the problem of computing the posterior distribution over an individual user's utilities given various kinds of information. Unfortunately, this distribution often does not have a simple parametric form. We discuss two major approaches to estimating the posterior distribution: Gaussian approximation (Section 6.4) and Markov Chain Monte Carlo techniques (Section 6.5).

mation to incorporate into our utility model. Unfortunately, most utility elicitation methods, such as standard gamble (Section 3.4.1) or time trade-off (Section 3.4.2) do not provide the utility values directly. They work by incrementally constraining each utility parameter until the remaining range is negligibly small. Consider a standard gamble question of the form: "Which is more preferable: a lottery $[o_\top, s, o_\perp]$ or outcome o for sure?" The information contained in the answer to such a question translates to evaluating the truth value of a constraint $U_o < s$ (Section 3.4.1). Thus, each step of the elicitation process using either standard gamble or time trade-off provides us with a constraint involving the utility of one outcome.

The only elicitation method that provides point values for utility parameters directly, visual analog scale (Section 3.4.3), suffers from a number of problems ranging from the lack of theoretical justification and its known cognitive biases to the lack of precision. Thus, we can never completely trust point values we obtain in this manner. We can treat them as approximations with some margin of error (in other words, as data obtained through a noisy sensor), which translates to a set of constraints of the same form as above.

There are other types of information we can obtain about a given user's utilities. *Preference questions* are questions of the form: "Is outcome o_i preferred to outcome o_j?" These questions establish the preference structure (ordinal ranking) among outcomes. Obviously, preference questions are much easier for the users to answer. There are no lotteries, which cause so many cognitive difficulties. We don't need to worry about the choice of anchor outcomes; both outcomes in any preference question refer to the same decision problem. The knowledge of such ranking is sufficient for rational decision making without uncertainty. A straightforward application of decision-theoretic principles cannot, however, use this type of information in the uncertainty case. In contrast, we can easily incorporate such information into our probabilistic framework. Each preference question translates to a constraint of the form $U_{o_i} > U_{o_j}$. We can condition our utility model on such constraints.

Thus, all new information we obtain in the course of our interactions with a given user comes in the form of constraints involving one or more utility parameters. In some cases, we can condition our model directly on such information. In others, we have to approximate the posterior distribution. The most general solution to this problem, which uses Monte Carlo sampling, is presented in Section 6.5. It applies to the case of arbitrary constraints over utility parameters. However, if the constraints involve only one outcome (as is typical for standard gamble and time trade-off methods), we can take advantage of the properties of the distribution and use more efficient solutions.

6.2 Utility Space

Our probability density function (PDF) over utility functions in the population $p(\mathbf{u})$ is defined over the $[0,1]^n$ hypercube, where n is the number of outcomes, or utility variables. This hypercube defines the space $\mathcal{U}$ of possible utility functions. Initially, the PDF assigns a non-zero probability to every point within the hypercube. The utility space is constrained only by the normalization assumption.

As we collect new constraints in the course of our interactions with a given user, parts of the utility space are eliminated. Let $\mathcal{C}$ be the set of constraints we have over the utility space at a given time. The *feasible region* $\mathcal{U}_\mathcal{C}$ is the part of the utility space consistent with these constraints. Our posterior distribution $q(\mathbf{u})$ ($p(\mathbf{u})$ conditioned on $\mathcal{C}$) should assign non-zero probability only to utility functions in $\mathcal{U}_\mathcal{C}$.

Information about the utility of a single outcome introduces axis-parallel constraints. If we restrict ourselves to this type of information, our feasible region remains a hypercube. Preference ordering among outcomes translates to diagonal constraints. Some behavior observations correspond to arbitrary linear constraints in the utility space. These two latter kinds of information reduce the feasible region to an arbitrary convex polytope.[1]

[1] A subset P of $I\!R^d$ is called a convex polyhedron if it is the set of solutions to a finite system of

through linear relaxation of these constraints.

6.3 Framework

Recall from Section 4.3 that we chose to represent the distribution over utility functions in the population as a mixture of multivariate Gaussian distributions, one for each subpopulation t. This population model will serve as our prior distribution $p(\mathbf{u})$.

When we collect constraints over the utility space during the interaction with a specific user, we need to condition our prior distribution $p(\mathbf{u})$ on the constraint set in order to compute a more informed posterior distribution $q(\mathbf{u})$. It would be desirable for $q(\mathbf{u})$ to have the same parametric form as $p(\mathbf{u})$. However, conditioning a multivariate Gaussian on utility constraints does not, in general, result in a new multivariate Gaussian.

In case of simple constraints of the form $U_o < s$, we can efficiently approximate the posterior distribution over $\mathbf{U}$ given the constraint set as a Gaussian distribution of the same form as $p(\mathbf{u})$. We discuss Gaussian approximation in Section 6.4.

If our utility information is of a more complex form, such as constraints $\sum_i \alpha_i U_{o_i} > 0$ for some coefficients α_i, we have to resort to Markov Chain Monte Carlo techniques to create a set of samples from the posterior distribution $q(\mathbf{u})$. MCMC methods are discussed in detail in Section 6.5.

6.4 Constraints Involving One Outcome

The computation is simplest in the case when we can assume that the different utility variables U_o are independent in p. We present this case first. A more realistic case, for an arbitrary distribution p, follows.

---

linear inequalities. P is called a convex polytope if it is a convex polyhedron and bounded. When a convex polyhedron (or polytope) has dimension k, it is called a k-polyhedron (k-polytope).

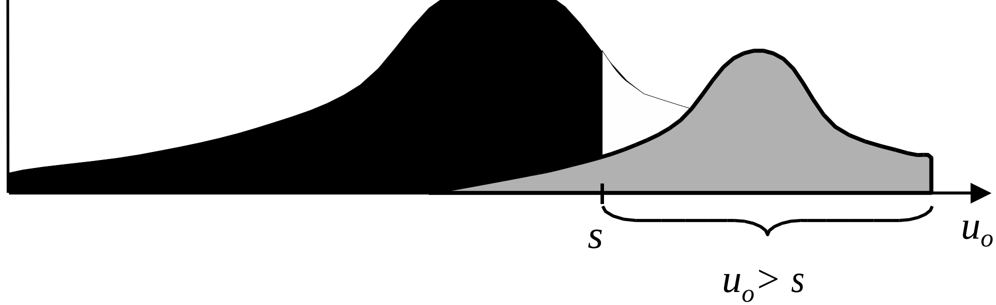


Figure 6.1: Conditioning the distribution over one outcome

6.4.1 Independent Outcomes

If we can assume that utility variables U_o are independent in p, we can deal with the distribution over each one of them separately. Thus, for every outcome U_o, we have a univariate Gaussian distribution restricted to the $[0, 1]$ range. Clearly, this restriction means that the distribution is not truly Gaussian. Nevertheless, a Gaussian can be a reasonable approximation since the probability mass that lies outside of the normalized utility range will generally be negligibly small. Therefore, we assume that our distribution over U_o is

$$p(u_o) = -\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(u_o - \mu)^2}{2\sigma^2}\right).$$

However, once we hear the user's answer to a utility elicitation question, asserting $U_o < s$ (or its negation), we can no longer assume that the probability mass outside the further restricted range for U_o is small enough to ignore. Thus, we need to reestimate the moments of the distribution.

The expectation for U_o over the restricted range $[l, h]$ is

$$\mathcal{E}[U_o] \quad = \quad \int_l^h u_o p(u_o) du_o$$

The second term in the expression can be found easily in the normal distribution tables.

The variance of the posterior distribution restricted to $[l,h]$ range is of course $\mathcal{E}^2[U_o] - \mathcal{E}[(U_o)^2]$ where

$$\begin{aligned}
\mathcal{E}[(U_o)^2] &= \int_l^h (u_o)^2 p(u_o) du_o \\
&= \frac{\sigma}{\sqrt{2\pi}} \left[(-u_o - \mu) \exp\left(-\frac{(u_o-\mu)^2}{2\sigma^2} \right) \right]_l^h + (\mu^2 + \sigma^2) \int_l^h p(u_o) du_o.
\end{aligned}$$

Thus, we can easily compute the moments of the posterior distribution by using a combination of closed form integration and the normal distribution tables.

6.4.2 Correlated Outcomes

In the previous section, we assumed that the different utility variables U_o are independent in $p(\mathbf{u})$. Unfortunately, this assumption is too strong in many cases. Taking as an example our prenatal diagnosis domain from Section 1.4, we can easily see that it is quite likely that a woman's utility for one outcome involving a Down's baby will be correlated with her utility for another outcome involving the same event. In this section, we consider the more general case of an arbitrary distribution p, which will allow us to model the framework described in Section 4.3, where a distribution over subutility functions induces a distribution over utilities.

We begin by assuming that our prior $p(\mathbf{u})$ is a multivariate Gaussian with an arbitrary covariance matrix, constrained to lie within the $[0,1]$ hypercube. Again, our utility function distribution cannot be truly Gaussian because of the restriction on the parameters' range. Nevertheless, we assume, just as in the independent outcomes case, that we can ignore the probability mass that lies outside of the normalized utility range.

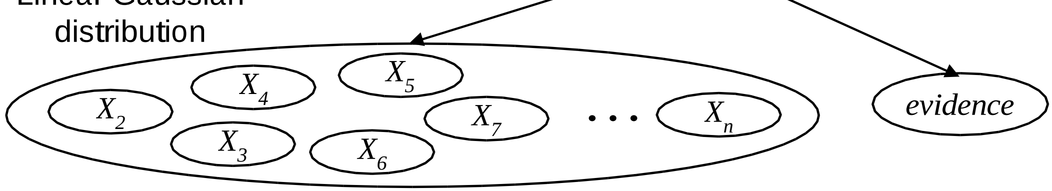


Figure 6.2: Decomposition of a multivariate Gaussian

Any multivariate Gaussian $p(X_1, X_2, \ldots, X_n)$ can be decomposed as a univariate Gaussian over X_1 and a *linear Gaussian (LG)* $p(X_2, \ldots, X_n \mid X_1)$ that defines a multivariate Gaussian over $X_2, \ldots, X_n$ with mean $\boldsymbol{\mu}(x)$, which is a vector linear function of x_1, and a fixed covariance matrix, which does not depend on x_1 (Shachter and Kenley 1989). If we condition X_1 on some evidence, the parameterization of the LG $p(X_2, \ldots, X_n \mid X_1)$ does not change.

The process of decomposing and reassembling the multivariate Gaussian is particularly simple in our case, since we are conditioning on a single variable (see (Shachter and Kenley 1989) for the general case).

- **From a conditional distribution to a joint distribution.**

  We start with a univariate Gaussian $X_1 = \mu^{X_1} + V$ (where $V \sim N(0, \sigma^2)$) and an LG $p(X_2, \ldots, X_n \mid X_1)$ parameterized by $\boldsymbol{\mu}^{LG}$, Σ^{LG} and a weight vector $\mathbf{b}$. We want to find the mean $\boldsymbol{\mu}$ and the covariance matrix Σ for the multivariate Gaussian $p(X_1, X_2, \ldots, X_n)$. It is easy to verify that

$$\begin{aligned} \boldsymbol{\mu}_1 &= \mu^{X_1} \\ \Sigma_{1,1} &= \sigma^2 \end{aligned}$$

For all $i = 2, \ldots, n$ we have

$$\begin{aligned} \boldsymbol{\mu}_i &= \boldsymbol{\mu}^{LG}(X_i) + \mathbf{b}(X_i)\mu^{X_1} \\ \Sigma_{1,i} &= \Sigma_{i,1} = \mathbf{b}(X_i)\sigma^2 \end{aligned}$$

- **From a joint distribution to a conditional distribution.**

  We are given a multivariate Gaussian $p(X_1, X_2, \ldots, X_n)$ with a mean vector $\boldsymbol{\mu}$ and a covariance matrix Σ. We want to decompose it into a univariate Gaussian $X_i = \mu^i + V$ and an LG $p(X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n \mid X_i)$. Let us assume, for the simplicity of exposition, that $i = 1$ (we can always rearrange the mean and the covariance matrix to put ith variable in the first place). We can immediately find the parameters of the univariate Gaussian over X_1:

  $$\begin{aligned} \mu^{X_1} &= \boldsymbol{\mu}_1 \\ \sigma^2 &= \Sigma_{1,1} \end{aligned}$$

  We can find the weight vector $\mathbf{b}$ by solving $n-1$ equations with $n-1$ unknowns:

  $$\Sigma_{1,i} = \mathbf{b}(X_i)\sigma^2,$$

  where $i = 2, \ldots, n$. Knowing $\mathbf{b}$, we can easily find $\boldsymbol{\mu}^{LG}$:

  $$\boldsymbol{\mu}^{LG}(X_i) = \boldsymbol{\mu}_i - \mathbf{b}(X_i)\mu^{X_1}.$$

  Finally,

  $$\Sigma^{LG}(X_i, X_j) = \Sigma^{LG}(X_j, X_i) = \Sigma_{i,j} - \mathbf{b}(X_i)\mathbf{b}(X_j)\sigma^2$$

  for all $i = 2, \ldots, n$ and $j < i$.

This suggests that we could decompose our multivariate Gaussian over $U_{o_1}, \ldots, U_{o_n}$ as a univariate Gaussian over $U_{o'}$, the utility parameter about which we have just acquired some information, and an LG over remaining variables given $U_{o'}$. After conditioning the univariate Gaussian on new information, we could reassemble the multivariate Gaussian by multiplying the posterior over $U_{o'}$ and the LG.

rather a “strip” of one. It can be verified that the conditional mean for this distribution is not a linear function. We address this difficulty using a simple approximation. Let q be the desired distribution, conditioned on all of the relevant information. We approximate q using a distribution $\hat{q}$ which is a multivariate Gaussian.

An approximation to the joint PDF as a multivariate Gaussian, given evidence on $U_{o'}$, can be obtained by finding the best approximation to $q(U_{o'})$ as a univariate Gaussian $\hat{q}(U_o)$ (just as in Section 6.4) and leaving the conditional linear Gaussian over the remaining utility variables given $U_{o'}$ unchanged. We can then regenerate a new approximate multivariate Gaussian $\hat{q}$ by multiplying $\hat{q}(U_{o'})$ and the LG.

We can extend this approach to the case of a mixture of Gaussians. First, we use our information about $U_{o'}$ (such as, e.g., $U_{o'} > s$) to update the mixture weights. This can be done by a standard application of Bayes rule:

$$p(t \mid U_{o'} > s) \propto p(U_{o'} > s \mid t)p(t),$$

for all mixture elements t. Next, we update every element of the mixture separately, by approximating the distribution as a multivariate Gaussian in the same way as described above.

6.5 Constraints Involving Multiple Outcomes

The constraints generated by preference questions (Section 6.1) and behavior observations (see Section 8.2) involve more than one outcome. If all of these constraints are linear, the feasible utility region they specify is a convex polytope.

New information, such as answers to preference questions and our observations of the agent’s actions, are evidence regarding $\mathbf{u}$. Certain utility functions in the original utility space $\mathcal{U}$ are consistent with the agent’s actions and answers, whereas others are not. We would like to condition our prior $p(\mathbf{u})$ on this evidence, to derive a more

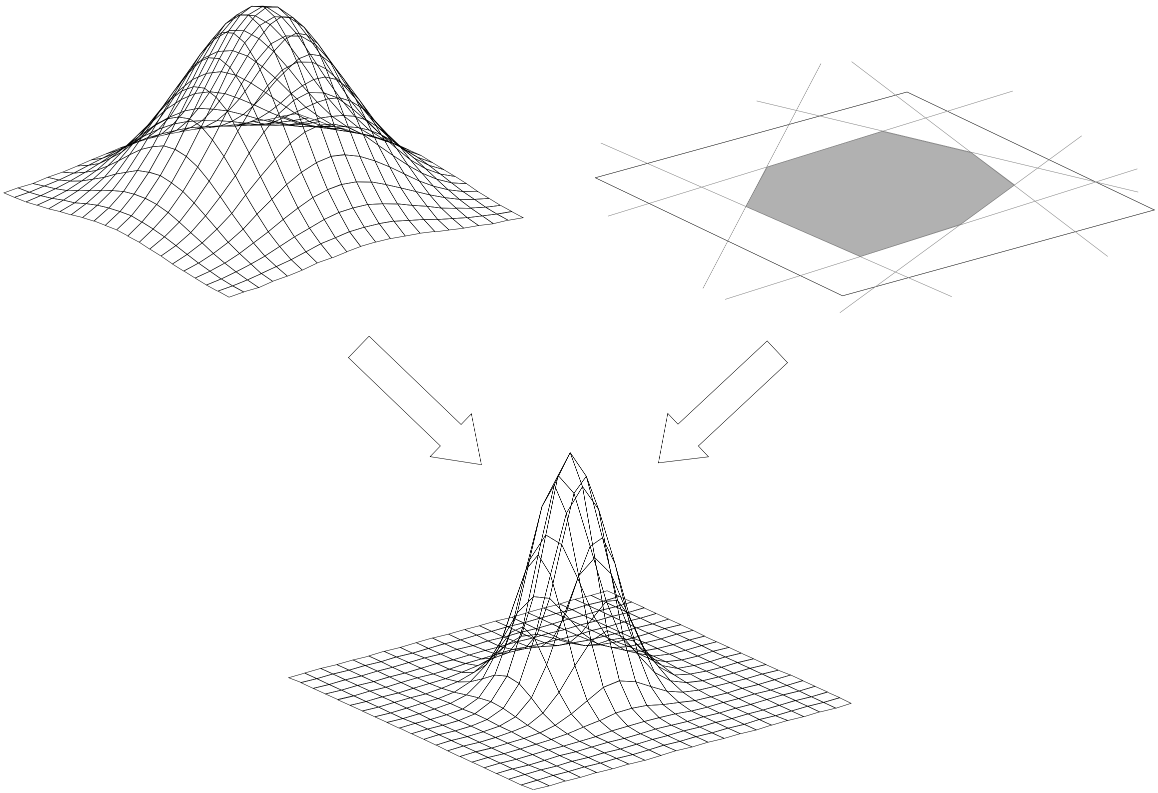


Figure 6.3: Conditioning the distribution on linear constraints in 2-dimensional utility space

ular polytopes (even convex ones) are computationally difficult to deal with. For example, even estimating the volume of a polytope (i.e., its probability under the uniform distribution) is a hard computational problem (Bárány and Füredy 1986). Fortunately, we can use Markov Chain Monte Carlo (MCMC) techniques to generate a set of samples from the posterior distribution $q(\mathbf{u})$ in an efficient fashion.

Our approach is based on the MCMC algorithms of (Applegate and Kannan 1991) for estimating the volume of a polytope on the one hand, and for generating samples from a log-concave density function on the other. Applegate and Kannan show that both of these algorithms are rapidly mixing, so that the number of sampling steps required for convergence to a stationary distribution is polynomial in the dimension of the polytope. It seems likely that a similar analysis can be applied to our algorithm, which simply combines the two algorithms into one.

Let $\mathcal{C}$ be the set of constraints on $\mathbf{U}$ and $\mathcal{U}_{\mathcal{C}}$ the feasible region, that is, the region where all utility functions are consistent with these constraints. We use a Metropolis-Hastings algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953; Hastings 1970), which traverses the convex set $\mathcal{U}_{\mathcal{C}}$ based on the distribution p. We first construct a regular grid in the n-dimensional hypercube $[0,1]^n$ with side-length β. This grid defines a partition of $[0,1]^n$ into cubes. The set $\mathbf{X}$ of cube centers in the feasible region $\mathcal{U}_{\mathcal{C}}$ constitutes the state space for our Markov chain $\mathcal{M}$. Then, starting from some arbitrary starting point $\mathbf{x}^{(0)}$ in $\mathcal{U}_{\mathcal{C}}$, we carry out a sequence of MCMC steps:

1. Starting from the current grid location $\mathbf{x}^{(t-1)}$, choose a candidate successor state $\mathbf{y}$ as follows. With probability 1/2, $\mathbf{y} = \mathbf{x}^{(t-1)}$. With probability 1/2, $\mathbf{y}$ is chosen uniformly from among $\mathbf{x}^{(t-1)}$'s $2n$ neighbors, so that for each possible neighbor $\mathbf{z}$, the probability that $\mathbf{y} = \mathbf{z}$ is $\frac{1}{4n}$.

2. If $\mathbf{y}$ is located outside $\mathcal{U}_{\mathcal{C}}$, stay at the current position, i.e., set $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$. Otherwise, accept the new location with probability $\min\left\{1, \frac{p(\mathbf{y})}{p(\mathbf{x}^{(t-1)})}\right\}$.

- $T(\mathbf{x},\mathbf{y}) = \frac{1}{4n} \cdot \min\left\{1, \frac{p(\mathbf{y})}{p(\mathbf{x})}\right\}$ if $\mathbf{y}$ is a neighbor of $\mathbf{x}$ and 0 otherwise ($\mathbf{x} \neq \mathbf{y}$);
- $T(\mathbf{x},\mathbf{x}) = 1 - \sum_{\mathbf{y}\in N(\mathbf{x})} T(\mathbf{x},\mathbf{y})$ where $N(\mathbf{x})$ is the set of neighbors of $\mathbf{x}$ in $\mathcal{U}_C$.

Let us examine the properties of this Markov chain. Note that $\mathcal{M}$ is *homogeneous*, that is, the transition probabilities do not depend on the time step.

We would like to show that the chain $\mathcal{M}$ converges to the desired distribution, namely our posterior $q(\mathbf{u})$. First, we need to define the concept of a *stationary* distribution. A stationary (or *invariant*) distribution over the states of a Markov chain is one that persists forever once it is reached. Note that a Markov chain may have more than one invariant distribution.

Definition 6.5.1: The distribution given by probabilities $r(\mathbf{x})$ is invariant with respect to the Markov chain with transition probabilities $T(\mathbf{x},\mathbf{y})$ if

$$r(\mathbf{x}) = \sum_{\mathbf{y}} r(\mathbf{y})T(\mathbf{y},\mathbf{x}).$$

∎

Proposition 6.5.2: *The distribution* $\{\hat{q}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ *such that, for all* $\mathbf{x} \in \mathbf{X}$,

$$\hat{q}(\mathbf{x}) = \frac{p(\mathbf{x})}{\sum_{\mathbf{y}\in\mathbf{X}} p(\mathbf{y})}$$

is invariant with respect to $\mathcal{M}$.

Proof: We need to show that

$$\hat{q}(\mathbf{x}) = \sum_{\mathbf{y}} \hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}).$$

Note that the only states with non-zero transitions into $\mathbf{x}$ are $\mathbf{x}$ itself and its neighbors.

$$
\begin{aligned}
&= \sum_{\mathbf{y}\in N(\mathbf{x})} \hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}) + \hat{q}(\mathbf{x})T(\mathbf{x},\mathbf{x}) \\
&= \sum_{\mathbf{y}\in N(\mathbf{x})} \hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}) + \hat{q}(\mathbf{x})\left(1 - \sum_{\mathbf{y}\in N(\mathbf{x})} T(\mathbf{x},\mathbf{y})\right) \\
&= \hat{q}(\mathbf{x}) + \sum_{\mathbf{y}\in N(\mathbf{x})} \left(\hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}) - \hat{q}(\mathbf{x})T(\mathbf{x},\mathbf{y})\right)
\end{aligned}
$$

The summation over the neighbors in the expression above is clearly equal to 0 if, for each $\mathbf{y} \in N(\mathbf{x})$,

$$\hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}) - \hat{q}(\mathbf{x})T(\mathbf{x},\mathbf{y}) = 0.$$

This follows from the fact that $\mathcal{M}$ is *reversible.* ∎

Lemma 6.5.3: *The Markov chain $\mathcal{M}$ is reversible. That is, it satisfies the condition of detailed balance*

$$\hat{q}(\mathbf{x})T(\mathbf{x},\mathbf{y}) = \hat{q}(\mathbf{y})T(\mathbf{y},\mathbf{x}).$$

Proof: If $\mathbf{x}$ and $\mathbf{y}$ are not neighbors, $T(\mathbf{x},\mathbf{y}) = T(\mathbf{y},\mathbf{x}) = 0$ and the condition is trivially satisfied. Otherwise, we need to show that

$$\hat{q}(\mathbf{x})\frac{1}{4n} \cdot \min\left\{1, \frac{p(\mathbf{y})}{p(\mathbf{x})}\right\} = \hat{q}(\mathbf{y})\frac{1}{4n} \cdot \min\left\{1, \frac{p(\mathbf{x})}{p(\mathbf{y})}\right\}.$$

There are three cases to consider:

1. $p(\mathbf{x}) = p(\mathbf{y})$

   In this case, $\hat{q}(\mathbf{x}) = \hat{q}(\mathbf{y})$ and $\frac{p(\mathbf{x})}{p(\mathbf{y})} = \frac{p(\mathbf{x})}{p(\mathbf{y})} = 1.$

2. $p(\mathbf{x}) < p(\mathbf{y})$

Since $\hat{q}(\mathbf{y}) = cp(\mathbf{y})$ and $\hat{q}(\mathbf{x}) = cp(\mathbf{x})$ (where c is a normalizing constant), we get

$$cp(\mathbf{x})\frac{1}{4n} = cp(\mathbf{y})\frac{1}{4n}\frac{p(\mathbf{x})}{p(\mathbf{y})} = cp(\mathbf{x})\frac{1}{4n}.$$

3. $p(\mathbf{x}) > p(\mathbf{y})$

   The argument is analogous to the previous case.

∎

Proposition 6.5.4: *The Markov chain $\mathcal{M}$ is ergodic. That is, the probability distribution over the state space converges asymptotically to the stationary distribution $\hat{q}(\mathbf{x})$ regardless of the initial state.*

Proof: First note that $\mathcal{M}$ is irreducible. That is, for any two states $\mathbf{x}, \mathbf{y} \in \mathcal{U}_\mathcal{C}$, $\mathbf{y}$ is reachable from $\mathbf{x}$ in a finite number of steps ($T^k(\mathbf{x}, \mathbf{y}) > 0$ for some k). All self-loop probabilities are non-zero. (In fact, for all $\mathbf{x}$, $T(\mathbf{x}, \mathbf{x}) \geq 1/2$.) Hence, it is ergodic (Neal 1993). ∎

After an initial "mixing phase," we store samples $\mathbf{u}^{(t)}$ from the Markov chain at regular intervals. We can also store all of the samples after the mixing phase, but the computational cost of using all of them is often too large. We choose $\mathbf{u}^{(t)}$ uniformly from the cube corresponding to $\mathbf{x}^{(t)}$.[2] Since the stationary distribution of our Markov chain $\mathcal{M}$ is $p(\mathbf{u})$ constrained to $\mathcal{U}_\mathcal{C}$, which is exactly the (discretized) posterior $\hat{q}(\mathbf{u})$, after some number of steps, we will start collecting samples from the desired distribution.

[2] We can discretize this choice by defining a second subgrid for each subcube with side-length $\gamma \ll \beta$.

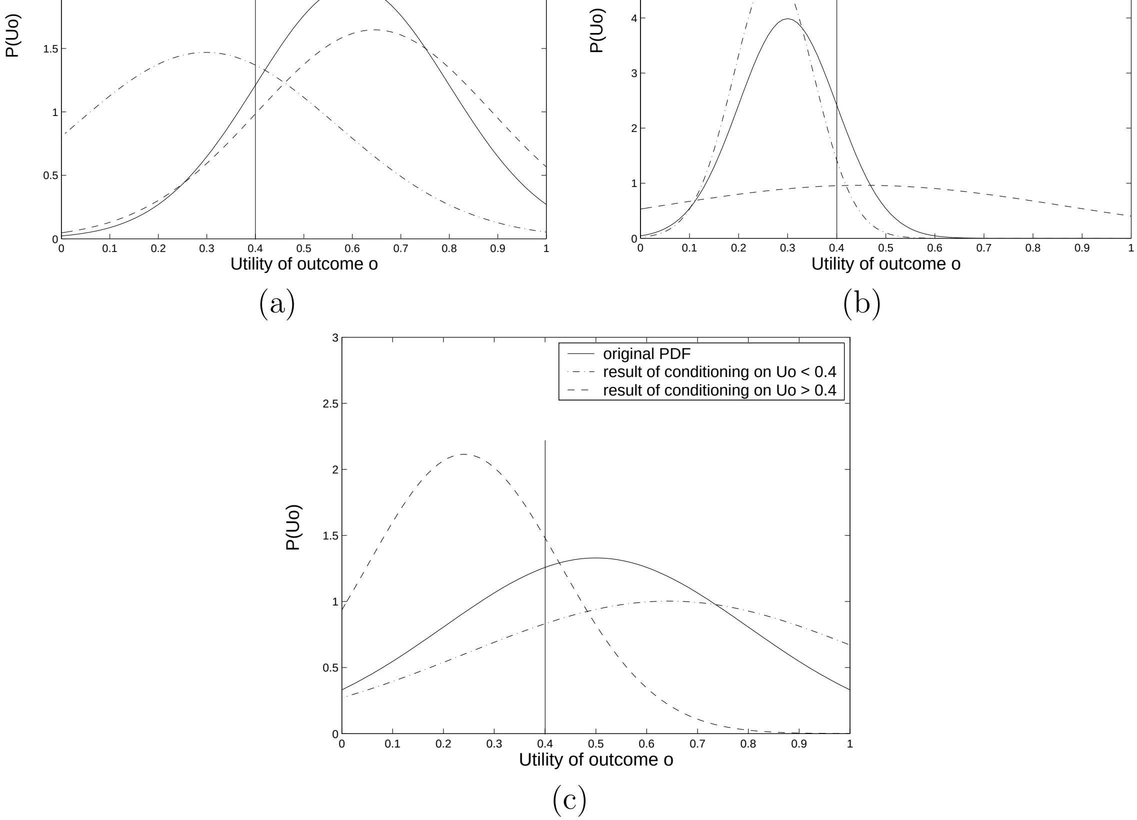


Figure 6.4: Conditioning a Gaussian on a single constraint. The parameters of the original PDF are: (a) mean 0.6, st. dev. 0.2; (b) mean 0.3, st. dev. 0.1; (c) mean 0.5, st. dev. 0.3

6.6 Experimental Results

First, we present experiments directly related to the behavior of our conditioning algorithms. Figure 6.4 shows the results of the Gaussian approximation in the independent outcomes case. As can easily be seen, the approximation produces very good results in some cases. In others, especially where the probability mass in the feasible interval (as indicated by the new constraint) is small, the resulting Gaussian has a large variance.

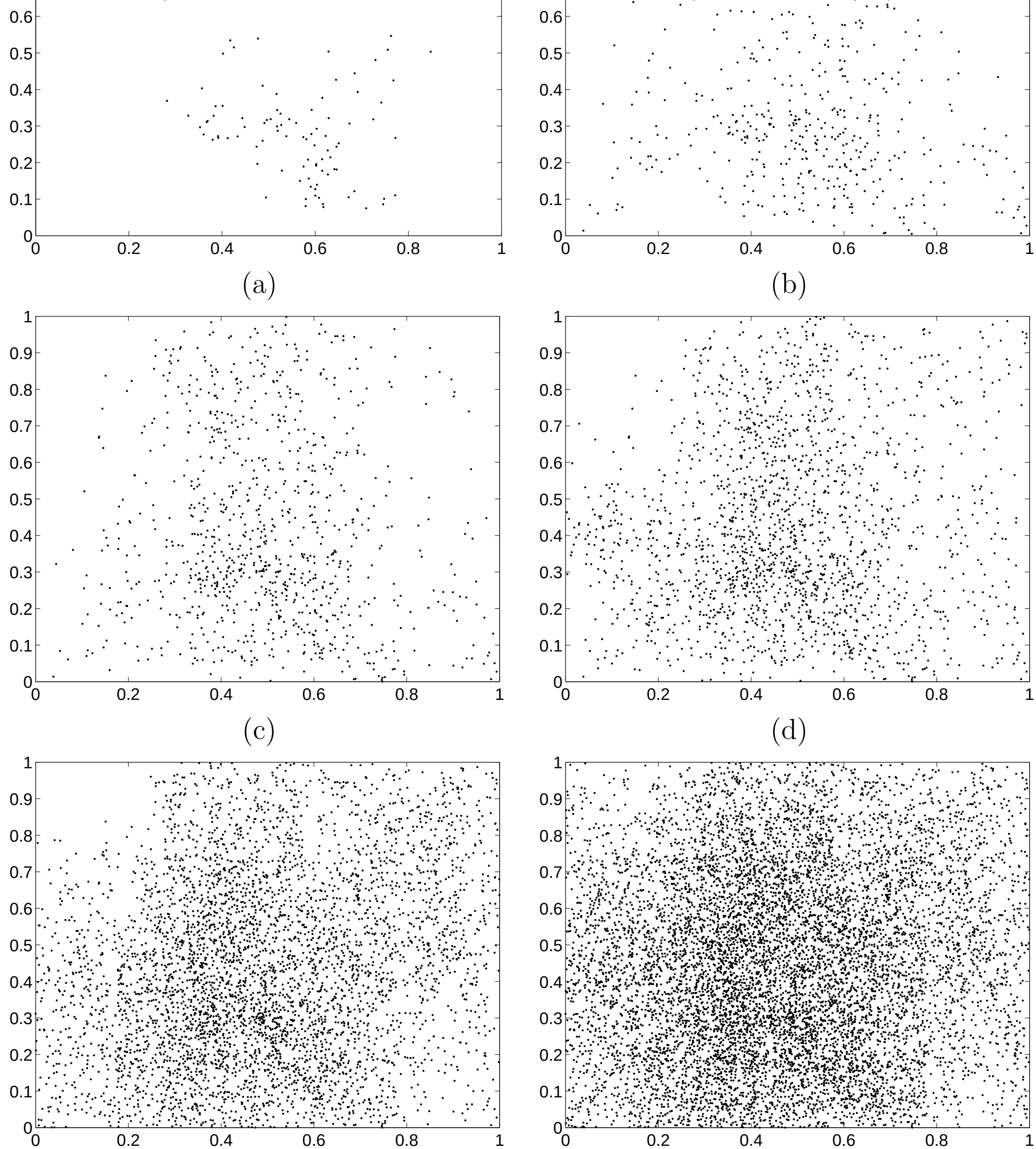


Figure 6.5: Projection of the MCMC samples onto the $u_1 - u_2$ plane ("enjoy" and "hate" attributes in the bookseller example) collected before any observations were made — (a) first 100 samples; (b) first 500 samples; (c) first 1000 samples; (d) first 2000 samples; (e) first 5000 samples; (f) all 10000 samples

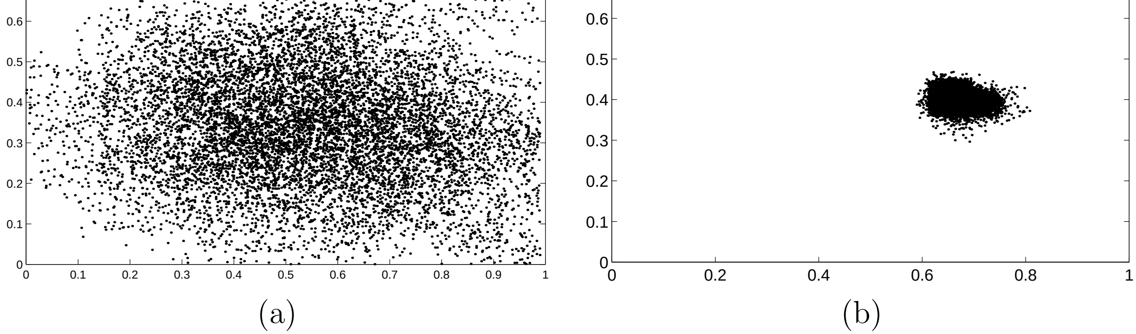
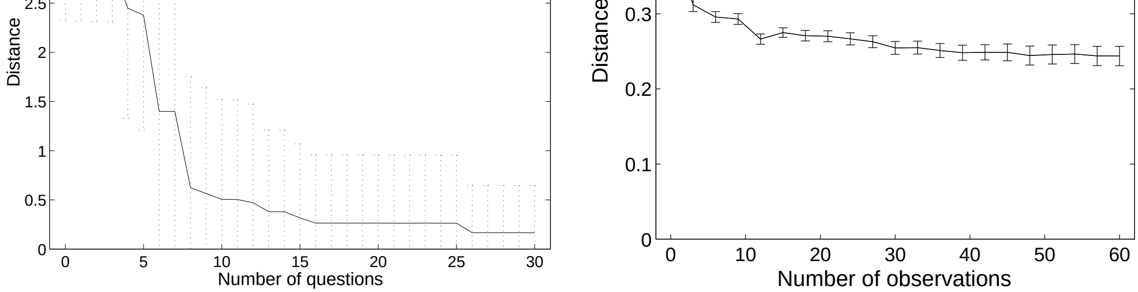


Figure 6.6: Projection of the MCMC samples onto the $u_1 - u_2$ plane ("enjoy" and "hate" attributes in the bookseller example) — (a) after 1 observation; (b) after 17 observations

We also investigated the behavior of our MCMC algorithm. First, we experimented with sampling parameters. We determined that a burn-in phase of 10,000 steps was sufficient to assure convergence to the stationary distribution. After the burn-in phase, we ran the Markov chain for another 100,000 steps, selecting samples at intervals of 10. Figure 6.5 shows the visualization of the sampling process. It displays the projection of MCMC samples onto the u_1–u_2 plane. The samples are taken according to the prior distribution $p(\mathbf{u})$, i.e., before any information about the utility function of a specific user was obtained. For this run, the mean of u_1 was set at 0.5 and the mean of u_2 at 0.4. Figure 6.5 shows the first k samples collected in a MCMC run for various values of k. As we can see, by the end of the run, the utility space is covered adequately.

Figure 6.6 illustrates the admissible utility region $\mathcal{U}_\mathcal{C}$ at different stages of the observation collecting process. We show the samples generated by the MCMC algorithm, projected again onto the u_1–u_2 plane: (a) shows the samples in a relatively early stage, after one behavioral observation, when only few constraints are present (one behavioral observation corresponds to several utility constraints, see Section 8.2), whereas (b) shows the samples after many observations (and many constraints) have



(a) (b)

Figure 6.7: Euclidean distance from the mean of the posterior distribution to the true utility function: (a) conditioning by Gaussian approximation based on the user's answers to utility elicitation questions (prenatal diagnosis domain); (b) conditioning by the MCMC technique based on behavior observations (bookseller example)

been collected. Note that the utility region is narrowly constrained and densely sampled at this stage.

In each of our conditioning algorithms, we are interested in narrowing our uncertainty about the agent's utility only as much as necessary to be able to recommend a nearly optimal course of action or to predict an agent's future behavior. In our experiments, we were able to accomplish that goal fairly quickly. Not surprisingly, the process required fewer steps in the case of utility elicitation, where the algorithm *chose* the information it wanted to acquire about a given user's utilities rather than passively observing the user's behavior.

Figure 6.7 shows the distance from the mean of the posterior distribution to the user's true utility function as a function of the amount of information we have been able to acquire about the user. In Figure 6.7(a), we present results for utility elicitation in the prenatal diagnosis domain (Section 1.4). Using 5-fold cross-validation, we estimate the distribution based on 4/5 of the database and test on the remaining 1/5.[3] At each step, we ask the user a random utility elicitation question and condition

[3]The utility database we use in the experiments is described in Section 5.3.2.

Section 6.4.1 to incorporate new information. The graph shows the average Euclidean distance from the mean of the posterior to the true utility function. As can easily be seen, the distance drops dramatically over the first 15 questions. The error bars indicate the variance: it is quite large in this example, possibly due to the fact that we make unrealistic assumptions about the utility function structure in our prior estimation. The small size of the training set may also contribute to the error in the learned parameters.

Figure 6.7(b) presents analogous results for behavior observations. In this case we use the bookseller domain (Section 1.5) with simulated utility data. We assume a fixed distribution and sample our test data from this distribution. At each step, we randomly generate a new game instance, observe the user's behavior, and condition our distribution on the constraints resulting from this observation using the MCMC technique from Section 6.5. The average distance from the mean of the posterior to the true utility function does not decrease as quickly as in the utility elicitation case (note the difference in scale between the two graphs). Clearly, observing the user's behavior does not result in a precise estimation of the user's utility function. This is not surprising — we cannot distinguish between utility functions causing the same behavior in the observed game instances. The region occupied by these functions can be quite large. However, the precise estimation of the utility function is not necessarily important. In some cases, it is enough to approximate it only as closely as needed to compute a nearly optimal course of action or predict the user's behavior.

In the next two chapters, we describe further experiments with both conditioning algorithms. The Gaussian approximation algorithm for one-outcome constraints is used in the context of a decision recommendation system for the prenatal diagnosis domain; the results of these experiments are presented in Section 7.5. The MCMC algorithm is applied in the context of predicting future actions of an agent based on observations of his past behavior. The domain we used for these experiments was the bookseller example; the results are described in detail in Section 8.5.

Chapter 7

Adaptive Utility Elicitation

As we demonstrated in Chapter 3, full utility elicitation in many real-life domains is infeasible. Apart from the problems with theoretical justification and cognitive difficulties inherent in many utility elicitation methods, in many domains we cannot assess the utilities of all outcomes, because the number of outcomes is just too large.

In order to apply decision-theoretic tools to such situations, we have to address two issues. First, we need to find a way to make optimal or nearly-optimal decisions based on incomplete utility information. As we have shown in Section 4.5.4, we can do this if we represent our beliefs over a given user's utility function as a probability distribution. Second, in order to use the time and attention that our users are willing to give us effectively, we should also carefully choose the questions we ask to elicit utilities. (In other words, a utility elicitation question is only useful insofar as it helps us reach the optimal decision for a given user in the particular decision task at hand.) In this chapter we explore the benefits of using our probabilistic framework in choosing the best utility elicitation questions to ask.

Our key idea is that utility elicitation and decision analysis should not be considered to be two separate tasks, but rather two parts of one process. Each of these parts can influence and inform the other and together they make the decision making process more accurate and more efficient.

same time, we would like to minimize the number of utility elicitation questions whose answers are needed to arrive at such a recommendation. We also need to quantify our confidence that the decisions we recommend are indeed close to optimal.

Our approach is based on an integrated algorithm for decision making and utility elicitation. The answers to our utility elicitation questions inform the decision making procedure, the results of which help us select the most informative utility elicitation question to ask next.

When the system encounters a new user, the only information available about his utility function $\mathbf{u}$ is the prior probability density function (PDF) $p(\mathbf{U})$. The algorithm then cycles through the following steps:

1. It computes the optimal strategy π^* relative to the current PDF $p(\mathbf{U})$.

2. If this optimal strategy meets the stopping criterion, it stops and outputs π^*.

3. Otherwise, it selects a utility elicitation question to ask the user, and asks it.

4. It conditions $p(\mathbf{U})$ on the response.

We consider questions that follow the standard gamble (Section 3.4.1) pattern: "Given the choice between outcome o for sure and a lottery which gives $o_\top$ with probability s and $o_\perp$ with probability $1 - s$, which will you choose?" We translate the response to this question to a constraint of the form $U_o < s$ or $U_o > s$, depending on the response. We call the value of s a *split point*. Each answer reduces the range of values allowed for one of the outcomes.

Consider a cycle of this process. Initially, we have a PDF p over the user's utilities. Let μ be the mean of $\mathbf{U}$ under p, and π^* the strategy that is optimal relative to μ. Now, consider a question regarding an outcome o and a split point s. If the user responds that $U_o < s$, we condition our PDF p, resulting in a new PDF $p_{<s}$; this will give us a new mean $\mu_{<s}$, and as a result, a new optimal strategy $\pi^*_{<s}$. Similarly, if he responds that $U_o > s$, we obtain a PDF $p_{>s}$ with $\mu_{>s}$ and associated optimal strategy

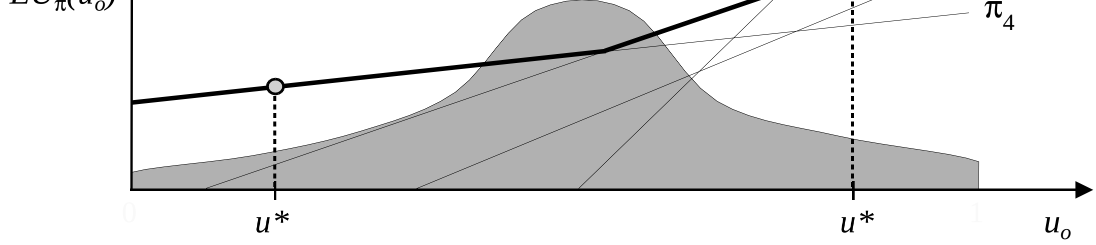


Figure 7.1: Expected utility of different strategies as a function of one outcome

$\pi^*_{>s}$. The usefulness of the question can be measured by its *value of information*: the improvement in expected utility that the user can expect by following either $\pi^*_{<s}$ or $\pi^*_{>s}$ (weighted by the probabilities of the two responses) instead of π^*. As further questions are asked and more information is obtained, our probability distribution p is updated, and the choice of optimal strategy changes to better fit the user's true preferences.

Note that our questioning pattern differs from standard gamble in a significant way: we do not ask about the same outcome for different values of s until the indifference point is reached (see Section 3.4.1). Rather, we choose questions so as to reduce the total number of questions we need to ask the user. A given question will often be for a different outcome than the previous one.

Since the questions we use induce only single-outcome constraints, we condition our prior PDF p using the Gaussian approximation algorithm presented in Section 6.4. Note that we could expand the range of questions also to include preference questions (direct comparisons between two outcomes), which induce diagonal constraints in the utility space. In that case, we would use the MCMC algorithm (Section 6.5) to condition our model on these constraints.

over the user's utility function $\mathbf{U}$. As we showed in Section 4.5.4, the answer to this question is easy: the expected utility of any policy π with uncertainty $p(\mathbf{U})$ over the user's utility function is equal to its expected utility using the mean of $\mathbf{U}$ under p:

$$\mathrm{EU}_{\pi}(\mathbf{u}) = \mathrm{EU}_{\pi}(\mu).$$

Thus, the optimal policy for a fixed utility function equal to the mean of the distribution p is also the optimal policy with respect to the entire distribution.

7.3 Stopping Criterion

After a sequence of utility elicitation questions, we will have a posterior distribution $q(\mathbf{U})$ over the user's utility function, and an associated candidate optimal policy $\hat{\pi}$. We would like to estimate the regret associated with stopping the utility elicitation and recommending $\hat{\pi}$. In other words, we need to know how much better off the user would be if we elicited his full utility function. Assume that the user's true utility function is $\mathbf{u}$, and that the associated optimal strategy is $\pi^*_{\mathbf{u}}$. Then the user's utility loss UL is the difference between his expected utility, under $\mathbf{u}$, of $\pi^*_{\mathbf{u}}$, and his expected utility under the recommended strategy $\hat{\pi}$:

$$\mathrm{UL}(\mathbf{u}) = \mathrm{EU}_{\pi^*_{\mathbf{u}}}(\mathbf{u}) - \mathrm{EU}_{\hat{\pi}}(\mathbf{u}).$$

The regret, or expected utility loss is the expectation of the loss under $q(\mathbf{u})$:

$$\mathcal{E}[\mathrm{UL}] = \int \mathrm{UL}(\mathbf{u}) q(\mathbf{u}) d\mathbf{u}.$$

While computing this integral exactly is impractical — we would need to compute the regions in which every strategy is optimal — we can approximate it quite easily using Monte Carlo methods. We simply sample utility functions $\mathbf{u}$ from q, use the

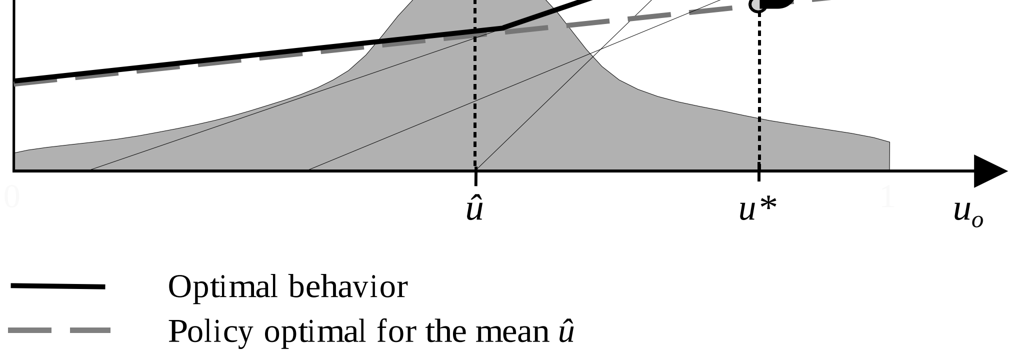


Figure 7.2: Utility Loss

decision tree representing our decision problem to compute the optimal strategy $\pi^*_{\mathbf{u}}$, and compute the utility loss for $\mathbf{u}$.

We can bound the number of samples N needed to estimate the regret using the upper bound on the worst case loss x, the desired threshold for expected utility loss ϵ, and the confidence parameter δ, with Chebyshev's inequality:

$$N > \frac{x^2}{2\epsilon^2\delta}.$$

ϵ is the utility loss we are willing to tolerate. It may be different for different application domains and different people. δ indicates the required degree of confidence in our estimate of the expected loss. Thus, N specifies the minimum number of samples we need to be able to verify (with confidence $1-\delta$) whether the expected utility loss does fall below the threshold ϵ.

It is difficult to find a tight upper bound on the worst case loss x. Fortunately, a loose bound is often sufficient. Consider the utility function corresponding to the point $\mathbf{u}_l$ at the low end of the utility range. (As we start the elicitation process with the utility space being the $[0,1]^n$ hypercube, this point is at the origin.) As the expected utility monotonically increases with each utility parameter, we can guarantee that

$\text{EU}_{\pi^*_{\mathbf{u}_h}}$. Thus, we can easily compute a bound on the worst case loss as the difference between these two: $EU_{\pi^*_{\mathbf{u}_h}}(\mathbf{u}_h) - EU_{\hat{\pi}}(\mathbf{u}_l)$.

The choice of values for the desired threshold for expected utility loss ϵ and the confidence parameter δ is very important. The lower the threshold and higher the confidence parameter, the more questions we can expect to ask on average. On the other hand, we will also have better guarantees that the final utility loss of our recommendations really does fall below the desired threshold.

Our definition of regret, or utility loss, UL, bears some similarities to those of Savage (1951) and Bell (1982) (see Section 3.3.3). In both cases, regret is used to evaluate a given policy by comparing its utility to another policy that is optimal by some standard. The difference is that our comparison is done before we observe the choices made by nature about the event variables in the system. Of course, in our case, the user's exact utility function is not known, so we have to rely on estimating the expected utility loss rather than computing it exactly.

7.4 Choosing the Next Question

One of the important advantages of explicitly modeling our uncertainty over the user's utility is that we obtain a simple metric for evaluating possible new utility information, such as an answer to a utility elicitation question. The *value of information* measures the expected improvement in our decision quality derived from incorporating such new information.

7.4.1 Value of Information

Initially, we have a PDF p over the user's utilities. Let μ be the mean of $\mathbf{U}$ under p, and π^* the strategy that is optimal relative to μ.

Consider a standard gamble question of the form: "Which is more preferable: a

If the user responds that $U_o < s$, we condition our PDF p, resulting in a new PDF $p_{<s}$; this will give us a new mean $\mu_{<s}$, and as a result, a new optimal strategy $\pi^*_{<s}$. Similarly, if he responds that $U_o > s$, we obtain a PDF $p_{>s}$ with $\mu_{>s}$ and associated optimal strategy $\pi^*_{>s}$.

We define the *posterior expected utility* after asking this question as:

$$\mathrm{PEU}(o, s) = \mathrm{EU}_{\pi^*_{<s}}(\mu_{<s}) P(U_o < s) + \mathrm{EU}_{\pi^*_{>s}}(\mu_{>s}) P(U_o > s) \tag{7.1}$$

This is an average of the expected utilities arising from the two possible answers to the question, weighted by how likely these two answers are. The value of information is this expression minus the current expected utility:

$$\mathrm{VOI}(o, s) = \mathrm{PEU}(o, s) - \mathrm{EU}_{\pi^*}(\mu) \tag{7.2}$$

Ideally, we would like to evaluate the value of information of a sequence of utility questions and minimize the length of the sequence while at the same time maximizing its value of information. Due to the apparent intractability of this type of fully general value of information computation, we make the standard approximation and restrict attention to the myopic value of information — the immediate improvement in decision quality — as shown in the formula.

We will start our analysis for the case in which the utilities of different outcomes are probabilistically independent, i.e., the different variables U_o are marginally independent in p. We then relax this assumption in Section 7.4.5.

7.4.2 Discretizing the Problem

The first problem we encounter in our search for the optimal split point is that the utility variables range over a continuous space, so that there are infinitely many potential split points for each outcome. Fortunately, it turns out that we can restrict our attention only to a finite number of them.

$$
\begin{aligned}
\mathrm{EU}_\pi(u_{o'}) &= p(u_{o'})u_{o'} + \sum_{o \neq o'} \int P(o|\pi)p(u_o|u_{o'})u_o du_o & (7.3)\\
&= p(u_{o'})u_{o'} + \sum_{o \neq o'} \int P(o|\pi)p(u_o)u_o du_o \\
&= p(u_{o'})u_{o'} + \sum_{o \neq o'} P(o|\pi)\mathcal{E}_p[U_o],
\end{aligned}
$$

where the second equality is due to our independence assumption about utility variables.

Hence, the expected utility of a given strategy π is a linear function of $U_{o'}$. The value for the optimal strategy for this problem is, for each value of $U_{o'}$, the maximum over all strategies π. Thus, it is a piecewise-linear, convex function of $U_{o'}$. We say that a strategy is *viable for* o' if it is optimal for some value of $U_{o'}$. We say that a particular value s is an *intersection point* if there are two viable strategies π_1 and π_2 that achieve the same expected utility at s, i.e., $\mathrm{EU}_{\pi_1}(s) = \mathrm{EU}_{\pi_2}(s)$.

Proposition 7.4.1: *The split point with the highest value of information will occur at one of the intersection points.*

Proof: Consider a potential split point s, and let π_L^* be the optimal strategy for the distribution $p_{<s}$ and π_R^* be the optimal strategy for the distribution $p_{>s}$. Let s^* be the strategy intersection point where $\mathrm{EU}(\pi_L^*) = \mathrm{EU}(\pi_R^*)$. Let us further assume, without loss of generality, that $s^* < s$. This situation is shown in Figure 7.3.

We want to show that $\mathrm{PEU}(o, s) \leq \mathrm{PEU}(o, s^*)$.

First, note that for any $a < b < c$ and any strategy π

$$
\begin{aligned}
&\mathrm{EU}_\pi(\mu_{[a,c]})P([a,c]) \\
&\quad = P(o \mid \pi) \cdot \mu_{[a,c]} \cdot \int_a^c p(u_o)du_o \\
&\quad = P(o \mid \pi) \cdot \frac{\int_a^c p(u_o)u_o du_o}{\int_a^c p(u_o)du_o} \cdot \int_a^c p(u_o)du_o \\
&\quad = P(o \mid \pi) \cdot \int_a^c p(u_o)u_o du_o
\end{aligned}
$$

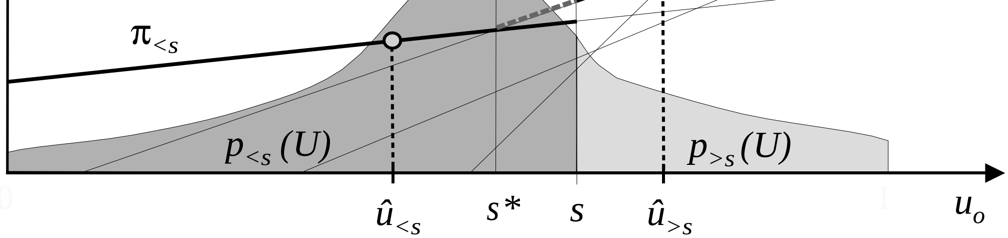


Figure 7.3: Comparison of two split points

$$
\begin{aligned}
&= P(o \mid \pi) \cdot \left(\int_a^b p(u_o) u_o du_o + \int_b^c p(u_o) u_o du_o \right) \\
&= P(o \mid \pi) \cdot \left(\frac{\int_a^b p(u_o) u_o du_o}{\int_a^b p(u_o) du_o} \cdot \int_a^b p(u_o) du_o + \frac{\int_b^c p(u_o) u_o du_o}{\int_b^c p(u_o) du_o} \cdot \int_b^c p(u_o) du_o \right) \\
&= P(o \mid \pi) \cdot \left(\mu_{[a,b]} P([a,b]) + \mu_{[b,c]} P([b,c]) \right) \\
&= \mathrm{EU}_\pi(\mu_{[a,b]}) P([a,b]) + \mathrm{EU}_\pi(\mu_{[b,c]}) P([b,c])
\end{aligned}
$$

We therefore have that

$$
\begin{aligned}
&\mathrm{EU}_{\pi_L^*}(\mu_{<s}) P(U_o < s) + \mathrm{EU}_{\pi_R^*}(\mu_{>s}) P(U_o > s) \\
&\quad = \mathrm{EU}_{\pi_L^*}(\mu_{<s^*}) P(U_o < s^*) + \mathrm{EU}_{\pi_L^*}(\mu_{[s^*,s]}) P(U_o \in [s^*, s]) \\
&\qquad\quad + \mathrm{EU}_{\pi_R^*}(\mu_{>s}) P(U_o > s) \\
&\quad \leq \mathrm{EU}_{\pi_L^*}(\mu_{<s^*}) P(U_o < s^*) + \mathrm{EU}_{\pi_R^*}(\mu_{[s^*,s]}) P(U_o \in [s^*, s]) \\
&\qquad\quad + \mathrm{EU}_{\pi_R^*}(\mu_{>s}) P(U_o > s) \\
&\quad = \mathrm{EU}_{\pi_L^*}(\mu_{<s^*}) P(U_o < s^*) + \mathrm{EU}_{\pi_R^*}(\mu_{>s^*}) P(U_o > s^*)
\end{aligned}
$$

where the inequality is due to the fact that π_R^* dominates π_L^* for every $u_o > s^*$, and therefore also for $\mu_{[s^*,s]}$.

Now, consider the two strategies $\pi^*_{<s^*}$ and $\pi^*_{>s^*}$ that are optimal for the distributions $p_{<s^*}$ and $p_{>s^*}$ respectively. These are not necessarily π_L^* and π_R^*, because the

$$
\begin{aligned}
\mathrm{EU}_{\pi^*_L}(\mu_{<s^*}) &\leq \mathrm{EU}_{\pi^*_{<s^*}}(\mu_{<s^*}) \\
\mathrm{EU}_{\pi^*_R}(\mu_{>s^*}) &\leq \mathrm{EU}_{\pi^*_{>s^*}}(\mu_{>s^*})
\end{aligned}
$$

From this, it follows that:

$$
\begin{aligned}
&\mathrm{EU}_{\pi^*_L}(\mu_{<s})P(U_o < s) + \mathrm{EU}_{\pi^*_{>s}}(\mu_{>s})P(U_o > s) \\
&\quad \leq \mathrm{EU}_{\pi^*_{<s^*}}(\mu_{<s^*})P(U_o < s^*) + \mathrm{EU}_{\pi^*_{>s^*}}(\mu_{>s^*})P(U_o > s*)
\end{aligned}
$$

which, when we subtract the term common to all VOI expressions, gives precisely that $\mathrm{VOI}(o, s^*) \geq \mathrm{VOI}(o, s)$. ∎

Thus, we only need to consider those strategy intersection points where the viable strategies $\pi^*_{<s^*}$ and $\pi^*_{>s^*}$ intersect at s^*; otherwise, as the proof shows, the strategy intersection point of these two strategies would have higher value of information.

7.4.3 Finding Strategy Intersection Points

How many strategy intersection points do we have to examine? Suppose we have N optimal strategies. Let us imagine moving the potential split point s from left to right over the range of U_o. We can mark an interval boundary whenever the optimal strategy for the area to the left or the optimal strategy for the area to the right of our split point changes. Note that once a strategy on the left side changes, it cannot change back: the mean $\mu_{<s}$ of U_o increases monotonically as we widen the region on the left, and the expected utility for any strategy is a linear function of this mean. Hence, the linear functions for any pair of strategies can cross at most once. Similarly, once a strategy on the right side changes, it cannot change back. As each strategy is optimal on each side at most once, we have at most $2N$ intervals, and at most $2N-1$ candidate split points. Thus, we need to consider only $2N-1$ split points, rather

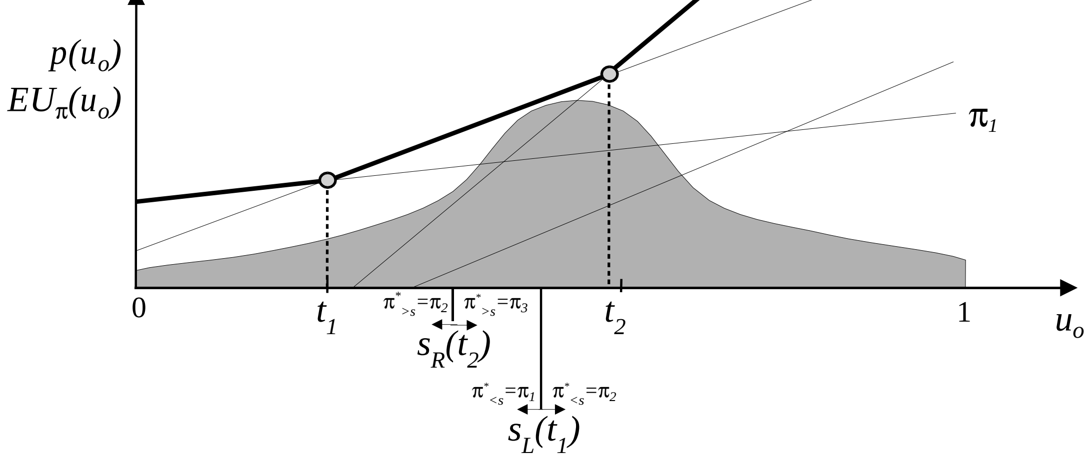


Figure 7.4: Finding strategy intersection points

than N^2. Each of these is only feasible, of course, if it is also a strategy intersection point of the two corresponding strategies.

We can execute this process efficiently using a simple binary search procedure, which utilizes the fact that we can find intersection points analytically.

First, we need to find the set of viable strategies. We can do this by propagating optimal strategy segments up the decision tree (Section 7.4.4). The endpoints of the segments in the piecewise-linear optimal strategy can be found by sorting the viable strategies by their slopes and computing the intersection points between each pair of neighbors in the sorted list. We will refer to the set of these segment endpoints (strategy intersection points on the optimal strategy surface) as T.

Next, we need to find the boundaries of $2N$ intervals where optimal strategies of the areas to the left and to the right do not change. Let I be the set of these intervals. For every optimal strategy segment endpoint $t \in T$, we want to find a point $s_L(t) > t$ such that $\mu_{<s_L(t)} = t$ and a point $s_R(t) < t$ such that $\mu_{>s_R(t)} = t$. Assume the point t is the intersection point between strategies i and j lying on the optimal strategy surface. Then, the point $s_L(t)$ is the boundary point of one of our intervals in I. All

We are not guaranteed to find both $s_L(t)$ and $s_R(t)$ for every $t \in T$; in fact, a point $s_L(t) \in [0,1]$ such that $\mu_{<s_L(t)} = t$ does not exist for any $t > \mu_{[0,1]}$. The mean of a subinterval sharing the left endpoint with its parent interval must be to the left of the mean of the parent interval. Similarly, a point $s_R(t) \in [0,1]$ such that $\mu_{>s_R(t)} = t$ does not exist for any $t < \mu_{[0,1]}$. Also, for the two endpoints of the utility function range we will not be able to find neither s_L nor s_R, since no interval in $[0,1]$ of non-zero length will have either 0 or 1 as its mean. The total number of intervals may be therefore much smaller than $2N-1$.

Figure 7.4 illustrates this process. We have three viable strategies: π_1, π_2, and π_3. The set of segment endpoints includes two points: $T = \{t_1, t_2\}$. We need to find the interval boundary points corresponding to t_1 and t_2: $s_L(t_1)$ and $s_R(t_2)$. Note that we need not compute the points $s_L(t_2)$ or $s_R(t_1)$, since $t_2 > \mu_{[0,1]}$ and $t_1 < \mu_{[0,1]}$. There are three intervals where optimal strategies for the areas to the left and right do not change: $I = \{[0, s_R(t_2)], [s_R(t_2), s_L(t_1)], [s_L(t_1), 1]\}$. The strategies optimal to the left and right for all the points in $[0, s_R(t_2)]$ are π_1 and π_2; for all the points in $[s_R(t_2), s_L(t_1)]$, π_1 and π_3; and for all the points in $[s_L(t_1), 1]$, π_2 and π_3.

Unfortunately, we cannot compute the positions of the interval boundary points (our s_L's and s_R's) in closed form. Instead, we use binary search. Each of these points needs to be approximated only loosely — as long as the ordering of the boundary points is established, we do not have to know their precise location. First, we compute the location of each of the boundary points stopping as soon as the we have narrowed the error margin to a prespecified length. Then, we sort the interval boundary points by their lower bounds and check if any of the pairs of neighbors in the ordered list have overlapping error margins. We reduce the binary search threshold and repeat the search for each of the interval boundary points in the pair until the overlap disappears.

With each of the boundary points, we store the information whether it is the strategy for the area to the left or right that changes and the numbers of the strategies optimal in each of the bordering intervals. Thus, we know for every interval which to the list of potential split points.

7.4.4 Number of Optimal Strategies

The result above suggests that the number of VOI computations required is linear in the number of viable strategies. At first glance, this result might not be very reassuring. After all, there is an enormous number of strategies: exponential in the size of the decision tree. Any computation which requires us to consider all of them is much too expensive in all but the most trivial of decision problems. Fortunately, the number of *viable* strategies is exponentially smaller than the total number of strategies. Indeed, we show that it is linear in the size of the decision tree. Given that we need to traverse the decision tree every time we use the decision model for finding an optimal strategy, this cost is very reasonable.

Proposition 7.4.2: *The number of strategies that are viable for o is at most the number of nodes in the decision tree.*

Proof: We prove this result by induction on the depth of the tree. For the base case, a tree of depth 0 consists of a single leaf, where we have only a single strategy. In this case, the number of nodes is 1, and the number of viable strategies is also 1. For the inductive case, consider a tree of depth $d+1$. Let k be the number of children of the root, and let ℓ_i be the number of nodes in the subtree corresponding to the ith child. By the inductive hypothesis, the number of viable strategies for the ith child is at most ℓ_i. We will refer to the set of viable strategies at ith child node as Π_i and the optimal strategy at this node as π_i^*. Note that the expected utility of π_i^* is a piece-wise linear function for all i.

There are now two cases. Either the root is a max node or an expectation node. In the first case, the expected utility function $\mathrm{EU}(U_o)$ is the maximum of the functions of the children. In the second case, it is a weighted average of the functions of the children, where the weights are the probabilities annotating the edges going out of function to another only at a point where one of the constituent functions changes from one linear function to another.

In case of a decision node, to compute the expected utility function of the optimal strategy at the root, we take the max of all viable strategies present at the children nodes

$$\mathrm{EU}_{\pi^*}(U_o) = \max\{\cup_i \Pi_i\}$$

where i ranges over the children of the root node. Clearly, the maximum of a set of linear functions is a piece-wise linear function. The number of segments in such a function is equal to the number of strategies optimal over some part of the space. If all functions viable at children nodes are still viable at the root, the number of segments in the expected utility function at the root will be equal to the total number of strategies present at the children nodes, $\sum_{i=1}^{k} |\Pi_i|$. Otherwise (i.e., if some of the functions viable at children nodes are dominated by other functions present at their "sibling" nodes), this number will be even smaller.

In case of an event node, we take the weighted average of the functions present at the children nodes. A weighted average of linear functions is itself a linear function. We can mark a segment boundary every time one of the constituent functions $EU_{\pi_i^*}(U_o)$, $1 \geq i \geq k$, changes from one linear function to another. For each i, the number of such changes in $EU_{\pi_i^*}(U_o)$ will be $|\Pi_i| - 1$. We will encounter $\sum_{i=1}^{k}(|\Pi_i| - 1)$ such changes (or fewer, if some of the change points coincide) for the total of $\sum_{i=1}^{k}(|\Pi_i| - 1) + 1$ segments. Thus, the number of segments in the combined function is at most the total number of viable strategies present at the children nodes.

∎

Therefore, the total number of strategies is linear in the size of the decision tree.

Example 7.4.3: Consider the decision tree in Figure 7.5. At each leaf, we have only one strategy. Its expected utility is a linear function of the utility for the given outcome. When we move one step up, to the decision node labeled "Abort", we take

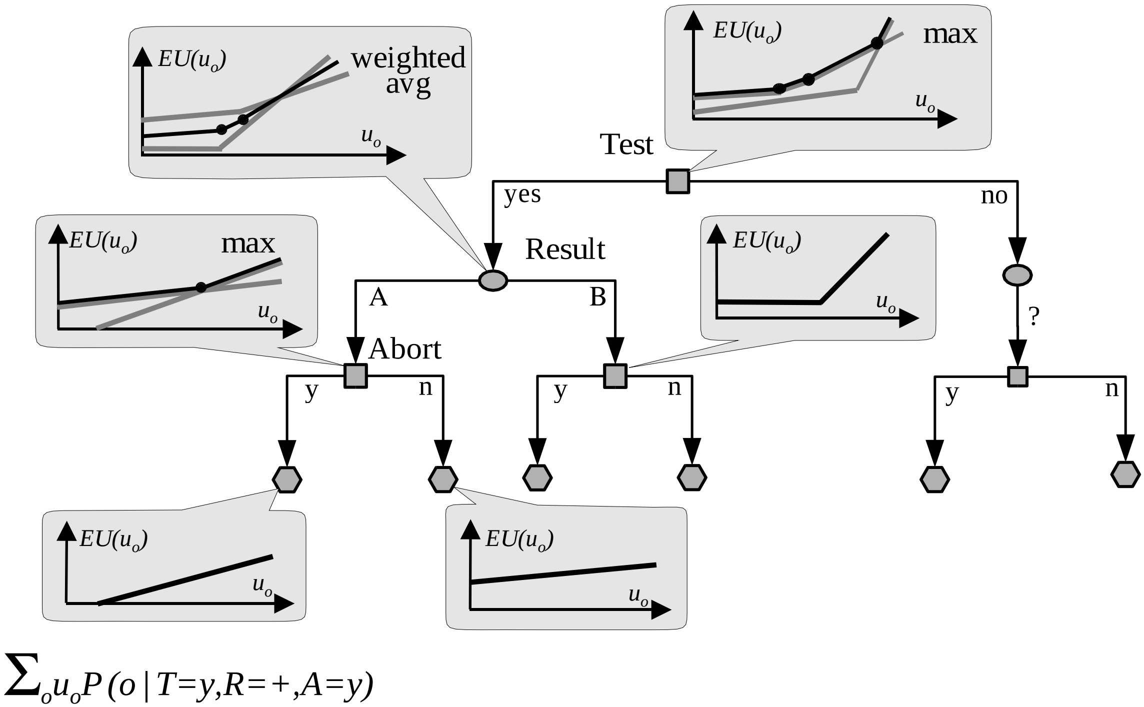


Figure 7.5: Optimal strategy propagation

of the children's optimal strategies. For the two "Result" nodes in the left subtree, this results in a piece-wise linear function with at most 3 segments. For the "Result" node on the right, which has only one child, the number of segments must be equal to the number of segments at its only child, at most 2. Finally, at the root node ("Test"), we take the max of the strategies viable at its 2 children. The number of segments in the resulting function is at most $2+3=5$. (In the example presented in Figure 7.5 one of the strategies viable at the "Result" node on the right is dominated by other strategies, so the actual number of segments is 4.) The number of viable strategies in this entire tree consisting of 12 nodes can be at most 5. ∎

7.4.5 Correlated Outcomes

The assumption that the different utility variables U_o are independent in $p(\mathbf{U})$ is too strong in many cases. In this section, we consider the more general case of a prior $p(\mathbf{U})$ which is a multivariate Gaussian with an arbitrary covariance matrix, constrained to lie within the $[0,1]$ hypercube.

We use a convenient property of multivariate Gaussians to apply the algorithm of the previous section with almost no modifications: given any variable $U_{o'}$, the conditional means of the remaining variables are linear functions of $U_{o'}$. In other words, in Equation (7.3), although we no longer have that

$$p(u_o \mid u_{o'}) = p(u_o),$$

we do have that

$$\int p(u_o \mid u_{o'}) u_o du_o = g(u_{o'})$$

for some linear function g. Thus, when we are enumerating the viable strategies for outcome o', as described in Section 7.4.2, we replace the means of the other U_o variables with their (linear) conditional means. The resulting function $\mathrm{EU}_\pi(U_{o'})$ is a linear function of the other. Second, even if our prior distribution is a multivariate Gaussian, once we condition our distribution on some information $U_{o'} > s$, the resulting posterior is no longer a multivariate Gaussian.

As we have shown in Section 6.4.2, we can address both these difficulties by approximating our desired distribution p (conditioned on all of the relevant information) using a distribution $\hat{p}$ which is a multivariate Gaussian. Once we have $\hat{p}$, we compute the value of information of all possible queries relative to $\hat{p}$, and pick the best one. We then condition p (not $\hat{p}$) on the newly obtained information (see Section 6.4.2), and continue. We believe that, in many cases, this approximation will give reasonable estimates of the true value of information. We note that our stopping criterion is always computed relative to the correct conditional distribution p.

We can extend this approach to the case of a mixture of Gaussians. We use the same idea of approximating each mixture component as a multivariate Gaussian, and then using our algorithm above for finding the optimal split point relative to the approximation.

7.5 Experimental Results

In our experiments we used the dataset described in Section 5.3.2. We ran the tests separately for every possible value of mother's age. Due to the small size of the database, we assumed that the utility function does not change with age and used all functions in the database to run tests for all ages.

The utility functions in the database were elicited in two different ways: by using standard gamble (SG, see Section 3.4.1) and visual analog scale (VAS, see Section 3.4.3) methods. Due to reported discrepancies between results of these two methods, we ran all our experiments separately for each of the two sets of utilities.

We used five-fold cross-validation for experiments. Every experiment (one for each possible value of mother's age), consisted of 5 runs. Reported results are averages over

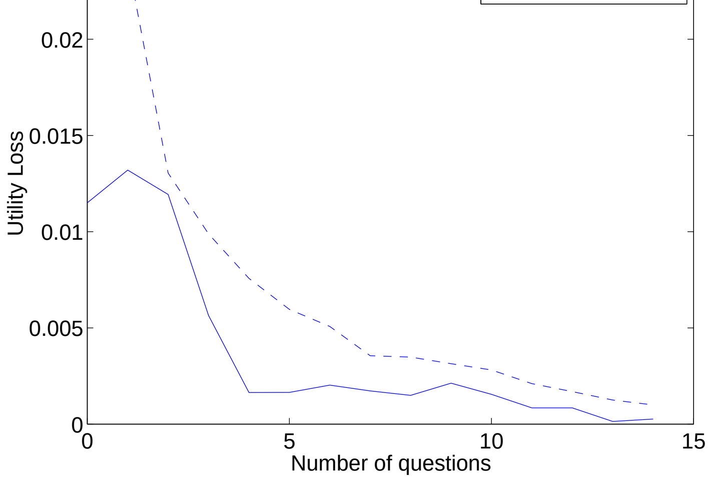


Figure 7.6: Expected and actual utility loss (age 20, SG database)

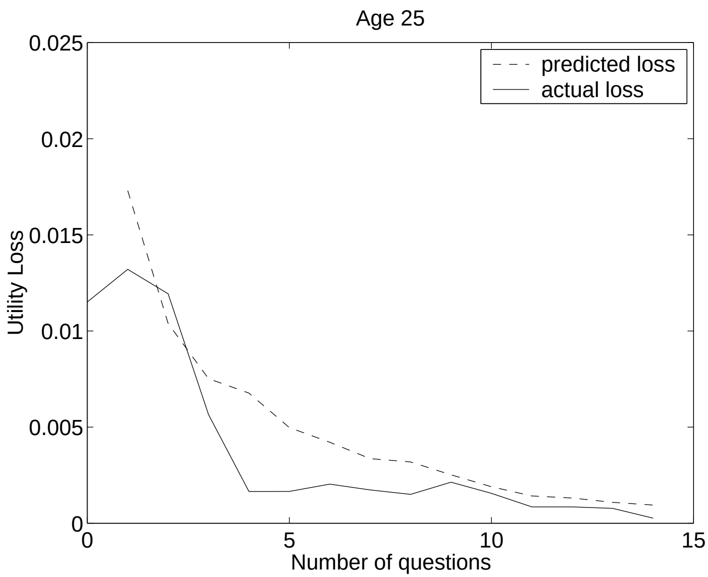


Figure 7.7: Expected and actual utility loss (age 25, SG database)

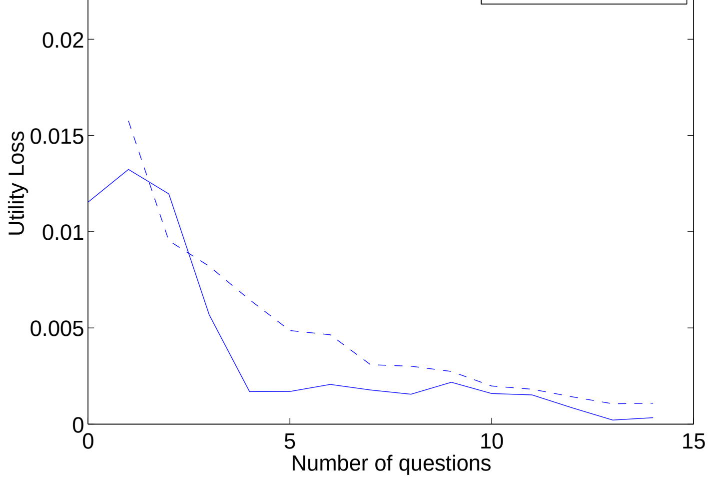


Figure 7.8: Expected and actual utility loss (age 30, SG database)

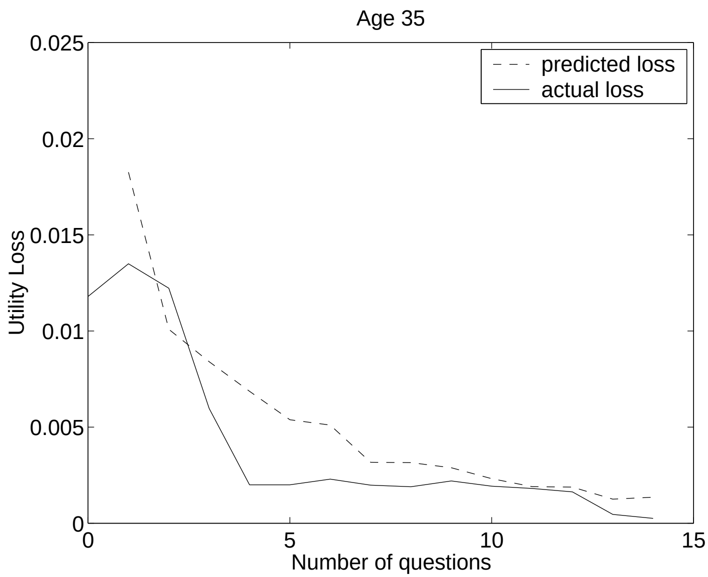


Figure 7.9: Expected and actual utility loss (age 35, SG database)

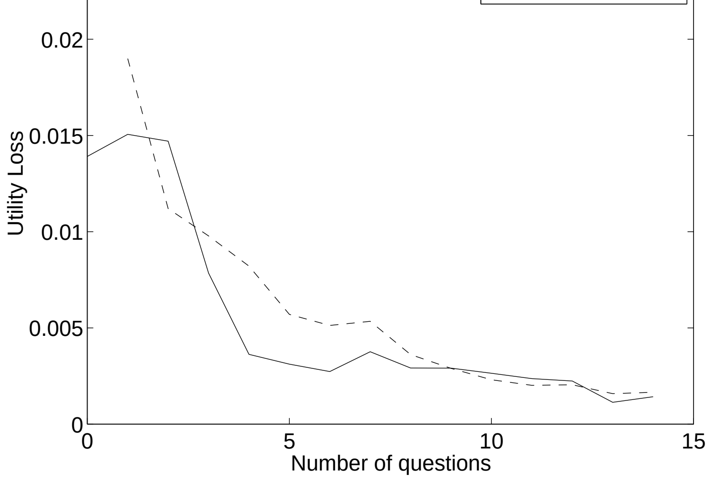


Figure 7.10: Expected and actual utility loss (age 40, SG database)

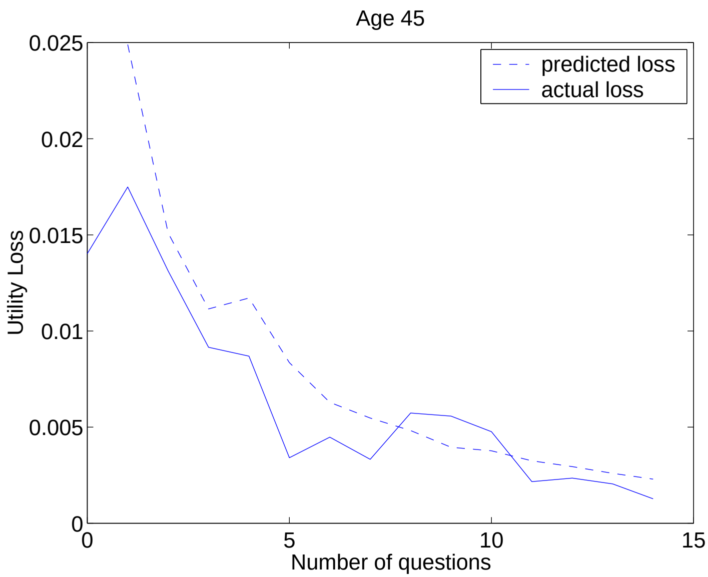


Figure 7.11: Expected and actual utility loss (age 45, SG database)

$\epsilon = 0.02$	1.2–3.6	1	13	0.5–1.7
$\epsilon = 0.01$	1.3–7.6	1	13	0.5–2.6

Table 7.1: Number of questions asked

all runs. The database was divided into 5 subsets. We used four subsets for training and one for testing. In each run, a different subset served as the test set. Training consisted of learning the distribution over the utility functions in the training set. For every utility function in the test set, we simulated adaptive utility elicitation. It consisted of a sequence of questions. After each question, we recorded the actual utility loss (i.e., computed with respect to the true utility function) resulting from following the strategy considered best at this point. We also recorded our estimate of the utility loss resulting from following this strategy. For every question, we computed the distance from the chosen split point to the indifference point. Finally, for every utility function, we kept track of the number of utility elicitation questions considered to be sufficient by our algorithm, the actual utility loss at the end of the elicitation process and our loss estimate at that point. We repeated the experiment for two values of the predicted utility loss parameter ϵ: 0.01 and 0.02. δ was set to 0.95.

We present the results for an uncorrelated Gaussian; our current database is too small to allow reliable learning of more complex densities. Figures 7.6, 7.7, 7.8, 7.9, 7.10 and 7.11 show the evolution of predicted and actual utility loss as more questions are asked. We see that the predicted utility loss starts out quite large, as the distribution is very broad. It gradually converges to the correct utility loss, and both gradually converge to zero. The predicted utility loss is usually an overestimate to the actual utility loss, implying that our algorithm is "safe" in not stopping prematurely. Interestingly, the actual utility loss for ages 18 through 35 followed an almost identical curve; the predicted utility loss for all ages was similar. The overall results are summarized in Tables 7.1, 7.2 and 7.3. The ranges indicate behavior for different ages.

$\epsilon = 0.02$	0.0002–0.016	0	0.13	0.001–0.033
$\epsilon = 0.01$	0.0001–0.006	0	0.12	0.001–0.018

Table 7.2: Utility loss after last question

Q1	Q2	Q3	average over first 15 questions
0.15–0.20	0.09–0.22	0.07–0.26	0.11–0.20

Table 7.3: Distance from indifference point (SG database)

As we can see, the number of questions asked is surprisingly small given the fact that we have 108 outcomes in the model. It increases slightly as we lower the threshold, but stays well within the bounds of what is possible in clinical practice. By comparison, the approach of (Chajewska, Getoor, Norman, and Shahar 1998) applied to the same decision problem and the same data achieves an average of 7.6 questions with an average utility loss of 0.016 on the SG database and an average of 5.2 questions with an average utility loss of 0.034 on the VAS database. With a smaller number of questions, we achieve a utility loss which is substantially lower. Furthermore, their approach provides no guarantees about the utility loss of the final recommendation. Finally, note that the split points our algorithm chooses are usually quite far from the indifference point (see Section 3.4.1), making the questions cognitively easy.

Chapter 8

Non-cooperative Settings

Consider the problem of trying to predict the future actions of an agent A. In a competitive setting, the ability to predict the opponent's actions would be very valuable in the effort to optimize our own decisions.

If A is a rational decision maker, he chooses his actions according to the principle of maximizing the expected utility. To follow his reasoning, we need to know the probabilistic model of the domain A uses in his calculations and A's utility function. In many cases, the domain model is shared among all players of the game. In some others, the history of interactions with A will allow us to estimate this model. A's utility function is typically much more difficult to acquire. In competitive situations, the knowledge of the opponent's utilities may constitute a strategic advantage, so A may not be willing to disclose such information.

The only information we have about A's utility function comes from observing his behavior over time. Using behavioral observations to estimate an agent's utility function was proposed by Ng and Russell (2000) in the context of *inverse reinforcement learning* in *Markov Decision Processes (MDPs)*. As they show, the agent's decisions can be viewed as a set of linear constraints on the space of possible utility (reward) functions. The set of utility functions consistent with the constraints is infinite. Ng and Russell propose a set of heuristics that attempt to select one utility function from within the consistent set.

In our approach, we make use of the notion of the utility function as a random posterior we can predict future actions of agent A and optimize our own actions to increase our payoff.

8.1 Asymmetric Games

Recall the bookseller example from Section 1.5. The game is presented in Figure 1.3. An online bookseller B considers whether to offer a discount to one of his frequent customers A on a newly published title. If A is willing to buy the book at the full price, the discount offer would be a mistake. However, if the high price is the only thing preventing A from buying, it would be better to sell the book at a discount.

We assume that A and B had a chance to interact in the past. Whenever A visits B's website, he looks at several books, checks their prices, sometimes reads a review. Sometimes he buys one or more books. From his choices and browsing patterns, B acquires some important information. On the one hand, he learns which books A expects to enjoy (a probabilistic model for a choice made by nature). On the other, he gains information about A's utility function.

Learning A's utility function is one of B's goals. If he knew it exactly, he would be able to compute easily the optimal course of action for himself. However, it is rarely possible or necessary to learn A's utility function exactly. The observer typically needs to deal with A in the context of a particular interaction, and learning A's utility function is only useful inasmuch as it helps predict A's actions in that context.

To provide a formal framework for this type of interaction, we consider a particular type of two-agent decision problem, where the two agents have very different "levels of awareness." There is one *informed* or *strategic* player B, who is aware of the nature of the strategic interaction, and one *oblivious* player A, who does not perceive the interaction as a game — he assumes that the strategic player follows a predetermined, randomized strategy (i.e., A treats the other agent's actions the same way as the random moves by nature). For example, in the bookseller game, the customer may if both players attempt to learn, then their anticipation of their mutual learning strategies might lead to infinite cycles (Fudenberg and Levine 1998; Selten 1991).

We assume that the strategic player B knows exactly the probabilistic model that A uses. This includes the probabilities of the events governed by nature (such as A's enjoyment of a particular book) and the probabilities that the oblivious player A assigns to B's moves (such as the chances of getting a discount offer). We also assume that A treats the game as a one person decision problem and uses the expectimax algorithm (Section 2.1.2) to compute the strategy he should follow. B is aware of A's inaccurate perception of the interaction.

The strategic player can observe the oblivious player's actions and hopefully learn something about his opponent's utility function. He can later use that knowledge in choosing the optimal actions for himself. This asymmetric type of interaction, although fairly restricted, arises in a variety of situations.

This framework generalizes the decision problem presented in Chapter 7. There, the informed player B is a benevolent one, whose goal is to elicit enough information about the utility function of the oblivious player A so that he can make decisions for A that are close to optimal. In this case, B sets up a series of questions (or other tests), to which A responds. A's utility for the different responses corresponds to his true utility function. B's utility does not depend on A's decisions directly; A's responses only provide him with information. After some number of questions, B makes a decision on A's behalf; the utility he gets for the various outcomes correspond exactly to A's utility function, which he does not know. Thus, the more informed B is about A's utility, the better the decisions he can make on A's behalf, and the better his own utility.

Our goal is to reach conclusions about the agent's utility function U based on observing his actions. This problem is of limited interest if the agent's utility function is specific to one particular decision task. Thus, we often assume that the utility function is derived from more general components that reflect the agent's general or an additional satisfaction from obtaining a bargain. We can acquire information about the agent's subutilities in one decision problem, and then apply it in another. In this case, we assume that the agent's utility function is linearly additive (see Section 3.2.5). In other words, there exists some set of subutility functions, each defined over the set of values of a single utility attribute, such that for any outcome, the agent's utility value is a weighted sum of subutilities of that outcome's attributes' values.

More formally, we assume that there exists some set of subutilities $\mathbf{u} = (u_1, \ldots, u_n) \in [0, 1]^n$ such that for any leaf L in a decision tree,

$$U(L) = \sum_{j=1}^{n} \alpha_{L,j} u_j.$$

An agent may encounter multiple decision problems, but his utility function is always composed of the same set of subutilities.

8.2 Learning Utilities by Observing Behavior

As we discussed above, the goal of the informed agent is to learn about the utility function of the oblivious agent A. In other words, we assume that the values of parameters $\mathbf{u} = (u_1, \ldots, u_n)$ which define A's utility function are unknown. If we assume that A is an expected utility maximizer, we can derive constraints on A's utilities by observing his actions. In this section, we show that A's decisions can be used to derive linear constraints on $\mathbf{u}$. Our result is similar to that of Ng and Russell (2000), but differs in two important ways. First, our linear constraints apply not only to an MDP, but to a general decision problem represented as a tree. Second, and more importantly, we provide an alternative solution to the problem of deriving constraints in cases where we do not observe A's full strategy (see Section 8.2.2).

whose utility function is defined by $\mathbf{u}$? The answer to this question is straightforward in principle if the optimal strategy π^* can be observed entirely. In this case, given the optimality of π^*, we obtain the set of optimality conditions:

$$\forall_{\pi \in \Pi} \mathrm{EU}(\pi^*) \geq \mathrm{EU}(\pi) \tag{8.1}$$

where Π is the space of pure strategies for A. Note that Equation (8.1) translates into a set of linear constraints $\mathcal{C}$ on the utility vector $\mathbf{u}$ due to the linearity of the expectation operator. Hence Equation (8.1) defines a polytope $\mathcal{U}^* \subseteq [0, 1]^n$ which contains all of the possible utility values $\mathbf{u}$ consistent with the observed behavior (Section 6.2).

The problem with using Equation (8.1) in the context of any algorithm is that the number of strategies π and hence the number of constraints implied by Equation (8.1) may be extremely large in practice. However, there exists a set of equivalent constraints that is exponentially smaller. Recall our discussion of viable strategies in Section 7.4.2. A strategy is viable if it is optimal over some part of the utility space. The equivalent set of constraints consists of those that state that the expected utility of the optimal strategy is greater than the expected utility of any other viable strategy:

$$\forall_{\pi \in \Pi'} \mathrm{EU}(\pi^*) \geq \mathrm{EU}(\pi) \tag{8.2}$$

where Π' is the set of all viable strategies. The number of viable strategies is linear in the size of the decision tree (Section 7.4.4).

We can use the structure of the decision tree to obtain this smaller set constraints. The basic idea is quite simple: we simulate the backward induction in the decision tree (Section 2.2.3), deriving the constraints along the way. We define a set of variables V_N representing A's expected utility at each node N, assuming he acts optimally from then on. We then define constraints in terms of these V_N.

the utility parameters $u_1, \ldots, u_n$.

- If A perceives N to be a chance node (whether truly a chance node or a decision node of the informed player), we are given fixed probabilities $p_{N'}$ for each immediate successor (child) $N' \in Ch(N)$. We can then define V_N in terms of the $V_{N'}$: $V_N[\mathbf{u}] = \sum_{N' \in Ch(N)} p_{N'} V_{N'}[\mathbf{u}]$.

- If N is a decision node for A, the observed strategy π^* tells us which child $\sigma(N)$ of N is chosen by the agent. We define $V_N[\mathbf{u}] = V_{\sigma(N)}[\mathbf{u}]$, and then add to $\mathcal{C}$ a set of constraints that implies consistency with the observed behavior at N: $V_N[\mathbf{u}] \geq V_{N'}[\mathbf{u}]$ for all $N' \in Ch(N)$.

The number of constraints generated in this fashion is linear in the size of the decision tree — exponentially less than the number of possible deterministic strategies over the tree, which is the number of inequalities implied by Equation (8.1). Nonetheless, both formulations describe the same region of the utility space $\mathcal{U}^*$.

8.2.2 Partial Strategy Case

In most cases, it is unrealistic to assume that π^* can be observed entirely. Some parts of the decision tree will never be visited, because some decision leading there is dominated by another. For example, in the bookseller scenario of Section 1.5, if A signs up for email notification, B will never learn what would have happened in the other branch of the decision tree. Even if we have multiple observations of the agent in the same decision problem, some parts of the decision tree may never be reached. Therefore, it is critical to have an approach that allows us to deal with partial strategy observations.

To understand the difficulty from a technical perspective, consider again the backward induction algorithm of Section 8.2.1. In A's decision nodes we may not know which of N's children was chosen by A. Hence, we cannot determine the expression

$$V_N[\mathbf{u}] = \max_{N' \in Ch(N)} V_{N'}[\mathbf{u}]. \tag{8.3}$$

The difficulty is that Equation (8.3) is non-linear in the subutility variables, leading to a non-convex region $\mathcal{U}^*$ of feasible utilities.[1] Thus, $\mathcal{U}^*$ can no longer be expressed using a set of linear inequalities, and in fact, can get exponentially complex. For our algorithm (below), as well as in the work of (Ng and Russell 2000), it is critical that our feasible region be described compactly as a set of linear inequalities.

One simple approach is to relax the constraints implied by Equation (8.3) in a way that is consistent with it yet gives rise to linear constraints. In detail, we derive (linear) upper and lower bounds on the expression in Equation (8.3), and use these bounds to specify the induced constraints on $\mathbf{u}$. Recall that each expression $V_N[\mathbf{u}]$ is a linear function of the form $\sum_{j=1}^{n} \alpha_{N,j} u_j$. As the u_j's are non-negative, one possible relaxation of Equation (8.3) is by replacing it with the two expressions

$$\overline{V}_N[\mathbf{u}] = \sum_{j=1}^{n} \max_{N' \in Ch(N)} \{\alpha_{N',j}\} \cdot u_j, \tag{8.4}$$

$$\underline{V}_N[\mathbf{u}] = \sum_{j=1}^{n} \min_{N' \in Ch(N)} \{\alpha_{N',j}\} \cdot u_j, \tag{8.5}$$

which are linear in the u_j's. Formally, Equation (8.4) and Equation (8.5) satisfy the conditions

$$\underline{V}_N[\mathbf{u}] \leq V_N[\mathbf{u}] \leq \overline{V}_N[\mathbf{u}], \tag{8.6}$$

and can hence be used to define relaxed constraints in a modified algorithm.

Our tree propagation algorithm stores two different expressions — one lower and

[1] In fact, the induced constraints are disjunctive. To see that, assume N has two children N_1 and N_2. In the part of the utility space where N_1 is preferred to N_2, we have that $V_{N_1}[\mathbf{u}] \geq V_{N_2}[\mathbf{u}]$ is in $\mathcal{C}$, but the constraints further up the tree involve V_{N_1}; in the other part of the space, we have the converse. In either case, the set $\mathcal{U}^*$ of feasible utilities is a non-convex region.

- At each leaf L, we define $\underline{V}_N[\mathbf{u}] = \overline{V}_N[\mathbf{u}] = \sum_{j=1}^{n} \alpha_{L,j} u_j$.

- For the node that A perceives to be a chance node, we take expectation for both $\overline{V}$ and $\underline{V}$, in the obvious way.

$$\begin{aligned}\underline{V}_N[\mathbf{u}] &= \sum_{N' \in Ch(N)} p_{N'} \underline{V}_{N'}[\mathbf{u}] \\ \overline{V}_N[\mathbf{u}] &= \sum_{N' \in Ch(N)} p_{N'} \overline{V}_{N'}[\mathbf{u}].\end{aligned}$$

- At each *observed* decision node with choice $\sigma(N)$, we define

$$\overline{V}_N[\mathbf{u}] = \overline{V}_{\sigma(N)}[\mathbf{u}]; \quad \underline{V}_N[\mathbf{u}] = \underline{V}_{\sigma(N)}[\mathbf{u}]$$

  and then add to the constraint set $\mathcal{C}$ the constraints:

$$\forall N' \in Ch(N) \;:\; \overline{V}_N[\mathbf{u}] \geq \underline{V}_{N'}[\mathbf{u}].$$

- At each *unobserved* decision node, we define $\underline{V}_j[\mathbf{u}]$ and $\overline{V}_j[\mathbf{u}]$ using Equation (8.4) and Equation (8.5).

After traversing the entire tree, the constraints in $\mathcal{C}$ define a convex region $\mathcal{U}_{\mathcal{C}}$ which is a superset of the *true* region of feasible utilities, $\mathcal{U}^*$.

8.3 Representing the Posterior Distribution

The constraints generated by behavior observations (Section 8.2) specify a region of feasible utility functions. We can use this region to derive an estimate of the true utility function, e.g., by finding the centroid of the region, or using the heuristics of (Ng and Russell 2000). However, any particular choice is, by force, somewhat estimate may not even be consistent with the observed behavior of A.

A more robust approach is to explicitly represent our uncertainty about A's utility function, by maintaining a probability distribution over this space as described in Chapter 4 and Chapter 6. In other words, we view the subutilities $\mathbf{u}$ as a continuous-valued random vector, and use a probability density function (PDF) $p(\mathbf{u})$ to represent our subjective beliefs about its possible values. As shown in Chapter 5, we can obtain p by applying density estimation techniques to a database of utility functions of many agents. The resulting distribution is an estimate of the utility functions in the population, and can be used as a prior distribution over the utility function of a newly encountered agent.

Our observations of the agent's actions are evidence regarding $\mathbf{u}$. Certain utility functions $\mathbf{u}$ are consistent with the agent's actions, whereas others are not. We can *condition* our prior $p(\mathbf{u})$ on this evidence, to derive a more informed *posterior* $q(\mathbf{u})$ (see Section 6.5). We can use the posterior distribution as the basis for reasoning about the future behavior of A, e.g., by using the posterior mean of the distribution, as an alternative to the utility estimate proposed by Ng and Russell. We can also generate samples $\mathbf{u}^{(t)}$ from the posterior distribution $q(\mathbf{u})$. As we discuss in the next section, a set of samples from $q(\mathbf{u})$ is substantially more useful than the constraints specifying the feasible set, and also more useful than a single point estimate of the agent's utility.

Note that these samples come from the distribution $p(\mathbf{u})$ constrained to $\mathcal{U}_{\mathcal{C}}$. In the case where we have fully observed strategies, and $\mathcal{U}_{\mathcal{C}} = \mathcal{U}^*$, this distribution is $q(\mathbf{u})$. In the case where we observe only partial trajectories, our points $\mathbf{u}^{(t)}$ are in $\mathcal{U}_{\mathcal{C}}$ but may not be in $\mathcal{U}^*$. However, we can test each sample utility function $\mathbf{u}^{(t)}$, to see whether it is compatible with A's observed behavior, and reject the ones that are not. This elimination can be realized by explicitly solving the decision problem for the sample utility values and checking their consistency with the observed behavior. Note that we are using the convex set $\mathcal{U}_{\mathcal{C}}$ merely as a computational tool to construct

8.4 Acting Optimally

The set of samples generated from the posterior density $q(\mathbf{u})$ may be useful for several purposes. First, it could be used to approximate the posterior mean $\mathbf{u}^*$, simply by averaging the sample points $\mathbf{u}^{(t)}$. The mean is a point estimate of agent A's utility. It can be used as an alternative to the point estimate provided by the approach of Ng and Russell. Rather than using heuristics to select among the otherwise indistinguishable elements of $\mathcal{U}^*$, it uses the prior density $p(\mathbf{u})$.

It is important to emphasize the difference between partial and full strategy observations in the context of selecting a point estimate. As we mentioned above, the region $\mathcal{U}^*$ is typically non-convex for partial observations. There is no guarantee that the mean of the distribution over this region is itself in the region. Thus, the posterior mean we compute might not itself be consistent with A's observed behavior.

Moreover, a single point $\mathbf{u}^*$ loses the information concerning our certainty about the estimate. This information might be critical in a situation where the agent B needs to act based on his estimate of A's future behavior. An action that might be optimal with respect to A's behavior if his utility is $\mathbf{u}^*$ might be highly suboptimal relative to slightly different utilities. Therefore B's response should be very different is he is fairly confident that A's utility is around $\mathbf{u}^*$, than if there is high potential variability.

A more robust approach for B is to try and plan his actions, explicitly taking into consideration his uncertainty about A's utility function. While this approach is infeasible relative to the entire distribution $q(\mathbf{u})$, we can use the samples from the distribution as a computationally convenient approximation of the posterior $q(\mathbf{u})$. We now show how these samples can be used effectively to optimize B's behavior, in the context of the asymmetric games discussed in Section 8.1.

An appropriate formal framework for representing B's decision problem is similar

$q(\mathbf{u})$. To avoid dealing with infinite trees, we use our samples $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(K)}$ as a discrete set of initial states, representing our approximation to $q(\mathbf{u})$. Nature therefore selects each of our K samples with probability $1/K$. Each choice $\mathbf{u}^{(k)}$ is associated with a first-level subtree T_k, which is precisely our original game tree, but using $\mathbf{u}^{(k)}$ to specify the subutility variables for A.

The informed agent B does not observe the choice of nature, and hence does not know his position in the game tree. Formally, there is a collection of *information sets*, representing B's uncertainty in this case. Recall that an information set is a collection of nodes such that the same player makes a decision at each of these nodes and the same moves are available to him at each of these nodes (Section 2.2.4). Each information set contains a set of nodes among which the player cannot distinguish. In this case, each information set will contain K nodes — one from each first-level subtree T_k — the set of nodes whose path (aside from the initial nature move) is the same.

Figure 8.4 shows the expanded tree for the bookseller example of Section 1.5. At the root, nature makes a choice about A's utility function. First level subtrees are identical except for the utility values at the leaves. B's decision nodes are filled in different colors to distinguish between the two information sets. The nodes filled in grey belong to the information set B finds himself in when he discovers that A signed up for the e-mail service. The black ones belong to the other information set, corresponding to the situation when A refused the service.

8.4.1 Oblivious Agent's Strategy Computation

Given the asymmetric information, players A and B have different perspectives on the resulting game. The oblivious agent A thinks he is playing a single-player game. He assumes that the actions of B are, like nature moves, chosen according to some known distribution. The agent A also knows his own utility function. Hence, from A's

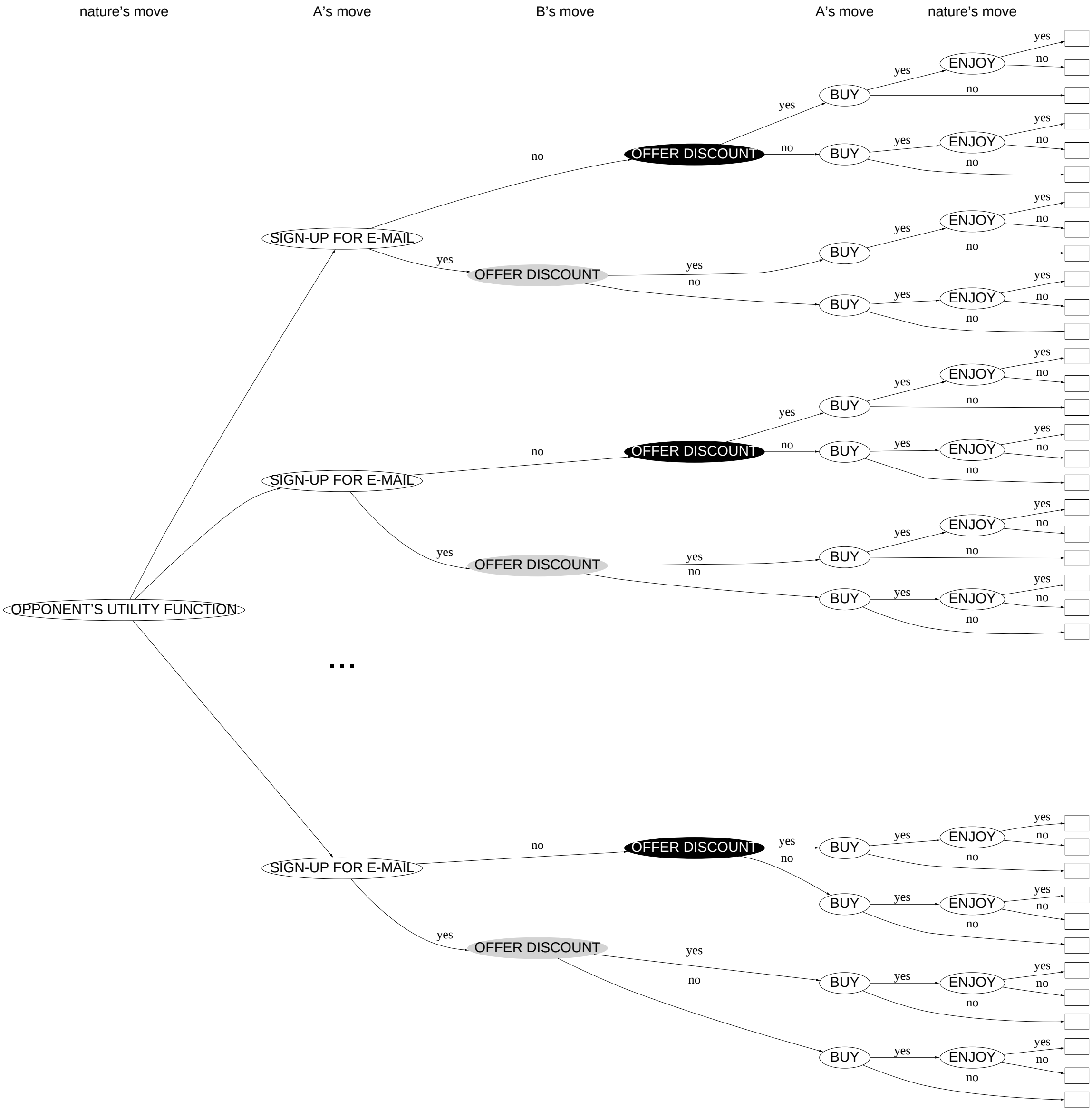


Figure 8.1: Information sets for B in the bookseller example

player. The utility values will depend on the choice of $\mathbf{u}^{(k)}$, where T_k is the subtree in which L is found. At each node N that A perceives to be a chance node (including B's decision nodes), we define

$$v_N^A = \sum_{N' \in Ch(N)} p_{N'} v_{N'}.$$

At each decision node N for A, he maximizes his benefit given his complete knowledge of the position in the tree by choosing the best successor node $\sigma(N)$:

$$\sigma(N) = \arg \max_{N' \in Ch(N)} (v_{N'}^A); \quad v_N^A = v_{\sigma(N)}^A.$$

8.4.2 Informed Agent's Strategy Computation

The situation for the informed agent is more complex. Since B is aware of the structure of the game tree and of the utility function $\mathbf{u}^{(k)}$ in each subtree T_k, he can predict A's actions in each subtree; however, he does not know which subtree he is actually in. As B cannot distinguish between the nodes in an information set, he must choose the same action in each of them. To optimize his strategy, he must compute the expected utility of each of his actions at each information set, where the expectation is taken relative to the possible utility values for A.

Moreover, B, being able to predict A's behavior in each subtree T_k, can conclude that certain actions are incompatible with $\mathbf{u}^{(k)}$. In this case, by the time he gets to an information set, he might be able to eliminate nodes inconsistent with the assumption of rationality for A. His choice of action should be optimal relative only to the remaining consistent nodes.

Formally, the optimal strategy for agent B can be computed using the following algorithm. The treatment for leaves and chance nodes is obvious. At each leaf L, we

Now, let N be a decision node for the oblivious agent A, and let $\sigma(N)$ be the successor chosen by A at this node, in the optimal strategy for A. B can predict A's action at each node; hence, he knows that $v_N^B = v_{\sigma(N)}^B$ (even though he may not know that he is at N).

The most complex case is a decision node N for the informed agent B. He has to select an action that maximizes his expected utility given his uncertainty about his exact location in the tree. We define $I = \{N_1, \ldots, N_K\}$ to be the set of nodes in N's information set. Consider each node $N_k \in I$. If the path from the root to N_k contains a node N' of agent A for which $\sigma(N')$ is not on the path to N_k, eliminate N_k from I. We will call the updated information set I'. Now, consider each decision d available at the nodes in N's information set, and let $\rho(N_k, d)$ be the node reached from N_k by taking the decision d. We define the expected value of decision d

$$\mathrm{EV}(d) = \frac{1}{|I'|} \sum_{N_k \in I'} v^B_{\rho(N_k,d)}.$$

B chooses the action $d^* = \arg\max \mathrm{EV}(d)$, so that we define

$$v^B_{N_k} = v^B_{\rho(N_k,d^*)}$$

for every $N_k \in I'$.

The algorithm for computing the optimal strategy for agent B is shown in Figure 8.2.

At the end of each game in our repeated interaction, A's observed behavior in that game is used to define new constraints, which are added to the constraint set $\mathcal{C}$ guiding the selection of sampled utility functions for the next interaction.

The strategic player should apply the strategy optimization algorithm described above at the beginning of each particular instance of a repeated game. The dynamic

Chance node: $v_N^B = \sum_{N' \in Ch(N)} p(N') v_{N'}^B$
A's decision node: $v_N^B = v_{\sigma(N)}^B$ $(\sigma(N) = \arg\max_{N' \in Ch(N)} (v_{N'}^A))$
B's decision node:
for all nodes N_k in B's information set I
check consistency of $\mathbf{u}^{(k)}$ with A's behavior on path to N_k
if not consistent, eliminate N_k from I forming I'
for each decision d available for nodes in I'
$\mathrm{EV}(d) = \frac{1}{|I'|} \sum_{N_k \in I'} v_{\rho(N_k,d)}^B$
where $\rho(N_k, d)$ is the node reached from N_k by taking d
choose $d^* = \arg\max \mathrm{EV}(d)$
$v_{N_k}^B = v_{\rho(N_k,d^*)}^B$ for all $N_k \in I'$

Figure 8.2: B's strategy computation

strategy update at an intermediate step of a game instance is not necessary, since the subtrees with utility functions incompatible with A's behavior in the current game are eliminated at each step.

Maintaining an Adequate Number of Samples

We note that B is gaining information about A's utility function in two places. At the end of each game in our repeated interaction, A's observed behavior in that game is used to define constraints, which guide the selection of sampled utility functions for the next interaction. And during the course of each interaction, sampled points that were generated based on the previously observed constraints are eliminated, as being incompatible with A's observed behavior in this game.

In effect, these two procedures are performing the same operation. The only difference, in the algorithm as we presented it, is that we only generate new candidate samples between interactions. However, even this difference is merely a design decision; if, during the course of a game, B ends up eliminating a large number of his candidate utility functions for A, it is advisable to generate additional ones (otherwise so far, assuming that no actions were observed at any of the nodes not yet reached in the game; we then use the MCMC algorithm of Section 6.5 with this more restricted constraint set to generate a new set of candidate utility functions for A, with which we can continue the game.

8.5 Experimental Results

To test the described approach in practice, we implemented a simulation of the bookseller example described above. This game has 5 subutility variables, shown in Figure 1.3, so $\mathbf{u}$ is a five-dimensional vector. The game is played repeatedly, with each repetition having different parameters: item price (full and discounted), and probability that the user will enjoy the book. This scenario reflects a situation where the user enters the web-site of the bookseller repeatedly; each time he is confronted with a different book at a different price. A practical advantage of this experimental design is that the constraints on the utilities change slightly during each repetition of the game, providing additional information about the set $\mathcal{U}^*$.

We experimented both with fully observed and partially observed strategies. In both cases, we simulated a sequence of randomly generated instances of the bookseller game. For each instance we formulated an incremental set of linear constraints derived from the play of the game. In the fully observed case, we used A's true full strategy. In the partially observed case, we used the "trajectory" observed in the actual play of the game. After adding each new set of constraints for the observed instances, we ran the MCMC algorithm of Section 6.5 to produce a posterior mean estimate $\mathbf{u}^*$ of the customer's utility function. We used a burn-in phase of 10,000 steps, to allow convergence to the stationary distribution; afterwards, we ran the Markov chain for 100,000 steps, selecting samples at intervals of 10. (These parameters were selected after we experimentally determined they are sufficient to adequately cover the space.)

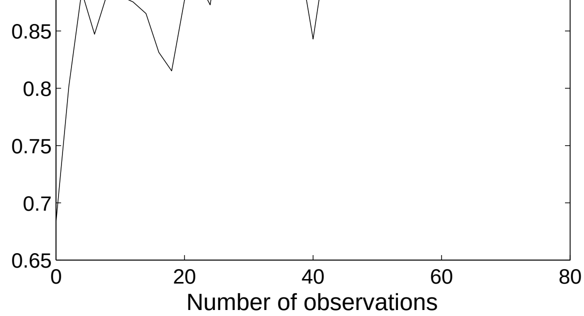


Figure 8.3: Fraction of MCMC samples within $\mathcal{U}^*$

First, we considered the behavior of our MCMC algorithm.[2] In the partially observable case, the MCMC samples are generated from the relaxation $\mathcal{U}_C$. These samples are not necessarily consistent with A's behavior, so we tested each generated sample for consistency with observed behavior, ensuring that our actual sample set contains only samples from $\mathcal{U}^*$. In Figure 8.3, we show the fraction of the generated samples in $\mathcal{U}_C$ that are also in $\mathcal{U}^*$. As can be seen, the fraction of sample points that are preserved is relatively large, and approaches one as the number of steps increases. We suspect that this improvement is due to the fact that observations later in the learning process help compensate for over-relaxation in the early periods.

Next, we tested the performance of our overall learning framework. We simulated the bookseller's interactions with 30 customers. The results we present here are averaged over all customers. For each customer, we sampled a utility function from our prior $p(\mathbf{u})$ and generated 20 game instances to serve as test cases. The bookseller never observed the customers' behavior in test games — they were used only to measure the accuracy of our predictions. Each customer's interaction with the bookseller consisted of a sequence of randomly generated game instances. At each step (consisting of two game instances), we collected the bookseller's observations of the customer's behavior in the form of constraints. After adding these new constraints

[2]Some of the results specific to that algorithm are presented in Section 6.6.

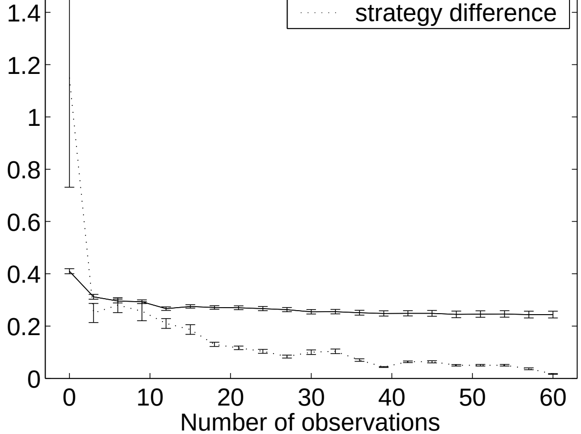


Figure 8.4: Estimation error for fully observed strategies

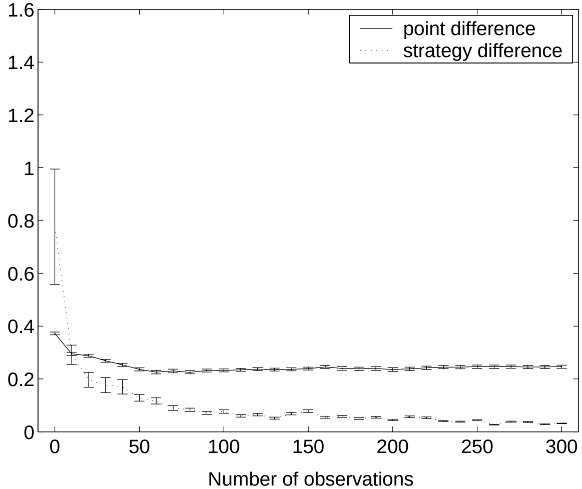


Figure 8.5: Estimation error for partially observed strategies

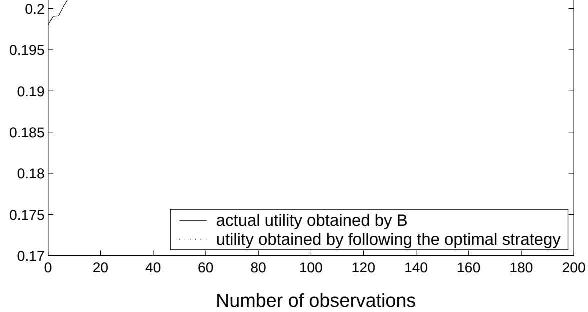


Figure 8.6: Realized utility for B as a function of the number of observations

to the constraint set, we used the MCMC algorithm to produce a new set of samples and a posterior mean estimate $\mathbf{u}^*$ of the customer's utility function. We compared this mean estimate with A's true utility function $\mathbf{u}$. We also computed B's prediction (based on the entire set of samples) of the customer's behavior on the test cases and compared it with A's true behavior. Finally, we computed the best strategy for B in test games and recorded the utility he would receive if these games were actually played. We also recorded the utility resulting from behaving optimally (i.e., given the complete knowledge of A's utility function).

Figure 8.4 and Figure 8.5 show the error as the number of observed game instances increases, for the fully and partially observable case. In both plots, the x axis is the number of observed game instances. The solid curve represents the average Euclidean distance between the true utility function $\mathbf{u}$ and the posterior mean estimate $\mathbf{u}^*$. The dotted curve is the strategy difference: the average number of states with deviating decisions between the strategy we predicted based on $\mathbf{u}^*$ (applied to 20 game instances we did not observe) and π^* — the best strategy for $\mathbf{u}$. We also show the variance as error bars.

We can see that the errors decrease drastically over time. In both cases, the error in the utilities remains bounded away from zero (at about 0.22) whereas the error in the strategies goes to zero. This can be explained by the fact that there will always the partially observable case the strategy prediction error takes considerably longer (requires more observations) to converge to zero.

Finally, Figure 8.6 shows the results for the optimized game play from the perspective of the informed player B. The graph shows the average utility B derives from playing a set of 20 games as a function of the number of games observed up to that point. We can see that B's performance improves over time. The dotted line indicates the optimal realizable utility if the true utility of A were known and is used as a benchmark. The fact that the actual utility approaches the benchmark indicates that the strategy of A is eventually predicted with high accuracy.

Chapter 9

Conclusions

9.1 Summary

This thesis extends the traditional framework of decision theory to the case in which we have some uncertainty over utility information. Using probability density functions to represent our beliefs about an agent's utilities allows us to deal in a principled way with the lack of completeness and precision inherent in utility assessments.

The probabilistic framework has many advantages. It helps us utilize any prior knowledge we may have. It provides us with many tools, such as expectation and value of information. We can learn a model of utility functions in a population of agents and use it as a starting point in our interactions with a new agent about whose utility we have no information. We can update our current belief about an agent's utilities by conditioning that model on new information we acquire in the course of the interaction. We can compute the optimal course of actions in both cooperative and competitive situations with respect to our current utility information, no matter how incomplete and imprecise.

We have shown the usefulness of our probabilistic representation in two specific contexts: in utility elicitation for a decision support system and in a two-player game in which one agent has very fragmentary information about the other's utilities.

in different subgroups of our target population. It accommodates a wide range of possible factorizations, including those corresponding to additive, conditionally additive, and generalized additive independence. At the same time, the model encodes a density function over utility values.

In Chapter 5 we presented an algorithm for learning such a probabilistic model. Our approach uses Bayesian learning techniques, and utilizes some of the same principles that have been used successfully in structure search for probabilistic models. This approach allows us to discover the factorization structure of the utility functions appropriate for a given domain.

Our approach is significantly more expressive than the naive linear-regression approach commonly used in the literature. First, it allows more general notions than simple additive independence; these are far more realistic assumption in many domains. Second, it explicitly accounts for different clusters of users that may use different decompositions. Indeed, our approach discovers interesting structure in the prenatal diagnosis domain of (Kuppermann, Shiboski, Feeny, Elkin, and Washington 1997), where the traditional linear regression model failed to do so.

The statistical learning perspective also has other benefits. By learning a statistical model of utilities in the population, we are able to associate a “confidence” in our assessment of an individual’s utility: if it is extremely unlikely given our model, perhaps fatigue or some other source of noise interfered with the elicitation process. We can also use the model to “smooth” our estimates in a user’s utility function, reducing the effects of noise. Finally and most importantly, we can use this statistical model as a starting point in our interactions with individual users.

9.1.2 Adaptive Utility Elicitation

In Chapter 7 we presented a new approach for making decisions based on limited utility information, and for targeting our utility elicitation process so as to lead to in making rational decisions. The algorithm we described allows us to perform these computations efficiently.

Our results suggest that our approach can make utility elicitation substantially easier for users of the decision model. In our simulated utility elicitation process, in most cases the questions our algorithm has chosen to ask were not in the immediate vicinity of the user's indifference point, thus making the task much easier cognitively. Furthermore, we have seen that our method substantially reduces the overall number of questions we have to ask before a good decision can be made; often, the number is as small as two or three, with a very small utility loss. Indeed, one might expect that the overall decision quality will be better, because our method allows us to avoid errors resulting from the fatigue caused by the utility elicitation process.

9.1.3 Non-cooperative Settings

In Chapter 8, we presented a new algorithm to learn the utility function of an agent A who acts in a sequential decision problem. This inference allows us to solicit utilities simply by observation. It may be used in an asymmetric two-agent setting, with the informed agent optimizing his actions relative to his uncertainty about the utility of the oblivious agent A.

Our approach consists of two parts: first, we use the observations to formulate a set of linear constraints on the utility space. Second, we use these constraints to transform a given prior distribution over utilities into a more informative posterior. We can then obtain "candidate" utility functions for A by generating samples from this posterior distribution. These samples can be used to compute the mean of the posterior, to obtain a point estimate of the agent's utility function. They can also be used as an estimate of the posterior distribution as a whole, capturing our uncertainty about A's utility. As we have shown, these samples are particularly useful in providing a feasible algorithm for the two-agent asymmetric game.

mainly by economists (Chapter 3). The relevance of their results for Artificial Intelligence was a largely unexplored topic for a long time. In the last few years, however, as the field of probabilistic reasoning began finding applications in real-world domains, it became apparent that utility modeling is a necessary component of modern decision-theoretic expert systems. The interest in utility modeling, utility elicitation and their relevance for decision making began to grow. Several researchers' ideas are particularly relevant to the work presented in this thesis.

In this section we review the related work in the areas of utility modeling, utility elicitation and acquiring utility information by observing an agent's behavior. We contrast this work with our own approach described in Chapters 4, 7, and 8, respectively.

9.2.1 Utility Modeling

Representing Structure

The role of structure in the utility function representation was recently analyzed by Bacchus and Grove (1995). They discuss the reduction in the size of the utility representation due to various independence properties between utility attributes and note that the properties offering the largest gains in representation size (additive independence, mutual utility independence) are rarely present in real world domains. They suggest that conditional additive independence (CAI) is likely to be a more realistic assumption in many situations while still offering a useful decomposition. They prove that conditional additive independence satisfies the *graphoid axioms* defined by Pearl (1988) for probabilistic independence properties. Thus, utility functions exhibiting CAI can be represented using undirected graphical models with vertex separation corresponding to additive independence. Bacchus and Grove also define a new property, a generalized additive independence (Section 3.2.5), which generalizes the notion of conditional additive independence to the case of overlapping subsets of pendence and conditional utility to match the properties of their probabilistic counterparts in the hope of exploiting existing techniques which allow us to represent and reason about probabilities efficiently.

He concentrates on a special subset of purely additive utility functions which he calls *TIOLI*. TIOLI utility functions are defined over binary attributes (each with domain $\{0,1\}$) $\mathbf{V} = V_1, V_2, \ldots, V_m$. By definition, a utility function is TIOLI if it decomposes as $U(\mathbf{V}) = \sum_{i=1}^{m} k_i V_i$ for some constants $k_1, k_2, \ldots, k_m$. If the constants are scaled to lie in the interval $[0,1]$ and sum up to 1, the utility function acquires properties of a probability distribution, albeit applied to utility attributes rather than to events. He shows that any utility function U can be converted to TIOLI form U'. Assuming the set of outcomes (states) $\mathbf{O}$ in the original space is ordered according to U and U is normalized, we simply create a new set of utility attributes $\mathbf{W}$, such that $U'(W_1) = U(o_1), U'(W_2) = U(o_2) - U(o_1), \ldots, U'(W_m) = U(o_m) - U(o_{m-1}), U'(W_{m+1}) = 1 - U(o_m)$. It can be easily seen that in this simple conversion the number of new utility attributes is the same as the number of outcomes. Some structured utility functions can be converted to TIOLI form using a smaller number of attributes.

This new formulation allows us to redefine some of the old notions of utility theory. Note that the new set of attributes $\mathbf{W}$ allows marginalization — it becomes meaningful to ask for a utility of a single attribute or (some) subsets of attributes. The notion of conditional utility can now be defined analogously to conditional probability and utility independence analogously to probabilistic independence. Thus, we can create *utility networks*, similar to Bayesian networks, where state variables are replaced by utility attributes.

It is unclear, however, how often we can take advantage of TIOLI representation while keeping the number of utility attributes to a reasonable number. As we have argued repeatedly in this thesis, the representation size of the utility function is critical in many real-world domains. It would be useful to be able to determine what to find a representation of a probability function and a utility function for a given decision problem which would allow them to decompose the expected utility function. The form of the utility function they focus on is a multiplicative version of TIOLI, in which the contributions of different attributes to the utility of an outcome are multiplied rather than added. This new type of utility independence (called *u-independence*) gives rise to a graphical representation, an *expected utility network* and an associated inference mechanism, which is modular in probabilities, utilities and expected utilities.

A different approach to representing structured utility functions was taken by Boutilier and his collaborators (Boutilier, Brafman, Geib, and Poole 1997; Boutilier, Brafman, Hoos, and Poole 1999; Boutilier, Bacchus, and Brafman 2001). They started by creating a graphical representation for preference structures (ordinal rankings of outcomes). Recall that in the case of full certainty about the state variables, we do not need utility functions — all that is required for reaching optimal decisions is a preference ordering a given user has over problem outcomes. Preference modeling is somewhat simpler than utility modeling, since no numerical values are involved. Preferences are also easier to elicit, since people find direct comparisons involving two outcomes relatively easy. However, some of the issues making utility modeling challenging remain, among them the need for structure to reduce the size of the representation.

Boutilier et al. (1999) suggest a new graphical representation called *conditional preference networks*, or *CP-networks* for preference structures. The network structure is based on preference independence and conditional preference independence properties among preference attributes (Section 3.2.3). Specifically, for all nodes in this directed graph, an attribute V_i corresponding to a given node is conditionally preferentially independent of all other attributes except its parent attributes $Pa(V_i)$ given the set of parent attributes $Pa(V_i)$. Unfortunately, such a structure is not always sufficient to recreate a total ordering among preference attributes — there are cases specify an algorithm to determine dominance in cases where it is possible.

In a recent paper Boutilier et al. (2001) propose a new formalism, *UCP-networks*, which can be viewed as an extension of the CP-network model that allows one to represent quantitative utility information along with the simple preference ordering. A UCP-network is a CP-network annotated with *utility factors.* A utility factor for a set of utility attributes $\mathbf{X}$ is a function assigning a numerical value to every instantiation $\mathbf{x}$ of the attributes in $\mathbf{X}$. Every node in a UCP-network has an associated utility factor over the attribute V_i corresponding to that node and its parent attributes $Pa(V_i)$. Utility factors in a UCP-network over attributes $V_i, V_2, \ldots, V_m$ satisfy the following property: $U(\mathbf{V}) = \sum_i f_i(V_i, Pa(V_i))$. Note that this property implies that every UCP-network specifies a generalized additive decomposition of the underlying utility function U. However, the acyclicity of the underlying directed network means that not every generalized additive structure can be represented by a UCP-network. Among those that can, not every structure satisfies the conditional preference independence properties required by the underlying CP-network. The main benefits of this representation lie in the efficiency of the algorithms it supports: all dominance queries are guaranteed to produce an answer and this answer can be found in time linear in the size of the network. Similarly, finding the outcome with the highest utility value requires only one pass through the network.

The representation of structured utility functions presented in this thesis follows that of Bacchus and Grove (1995) by concentrating on conditional additive and generalized additive utility functions. Our empirical results support (albeit on a single domain) their hypothesis that these independence assumptions, less strict than pure additive or mutual utility independence, are in fact applicable to real-world domains. The sets of utility function structures that can be represented by utility networks (Shoham 1997a) or UCP-networks (Boutilier, Bacchus, and Brafman 2001) are proper subsets of generalized additive structures that we can handle in our framework. None of the alternative formalisms have associated learning algorithms that

The idea of representing the uncertainty over utility information in a probabilistic form is not commonly used in the literature. It was introduced by Jimison et al. (1992). Their motivation was similar to ours. The idea was introduced in the context of a medical decision support system, similar to the setting in which we defined our adaptive utility elicitation algorithm. Surprisingly, this simple yet powerful idea was not much further developed until now.

The main goal of the work of Jimison and her collaborators was to automate the process of producing explanations of patient-specific recommendations generated by a decision support system. They suggested treating utilities as random variables and explicitly representing the uncertainty over utility information. In their model, the distribution over utilities is assumed to depend on several environmental variables such as the patient's occupation and age. The authors' intuition is that utility depends on lifestyle and experience. For example, they assume that an older patient with largely sedentary lifestyle will not mind a decrease in his or her physical capabilities as much as a young, physically active one. The authors point out that the distribution over utility functions should be based on data from large patient databases, but do not give a learning algorithm they would use to acquire such a distribution. As the information regarding the patient's age, occupation and lifestyle is acquired, the utility values are conditioned on it. The authors do not provide the details of the conditioning procedure or the assumptions made about the type of the distribution over the utility function. The assessment questions (most of them regarding event variables; utility elicitation is considered very costly and rarely used) are ranked by their value of information. The questioning ends when there are no more questions whose values exceed their costs. The mean of the posterior distribution is used to compute the optimal sequence of decisions.

Similarly, Jameson et al. (1995) assumed a probability distribution over a system's belief about user's preferences in the context of the software system called Pracma, which was designed to advise users on the purchase of a used car. They represented different attributes. The utility attributes in their system form a hierarchical structure with lower-level attributes contributing to the utilities of attributes at higher levels of the hierarchy. The functional form of these contributions (usually additive) is assumed to be known, although the exact parameters are not known precisely; the beliefs about the values of these parameters are encoded by the Bayesian network together with the beliefs about the utility values of individual attributes. The authors do not discuss the process of acquiring the probabilistic model of utilities.

As far as we are aware, no work other than ours attempts to integrate the representation of the uncertainty over utility values with uncertainty over utility function structure.

Model Refinement

In our work, we have assumed that the decision model for a given domain and the choice of the utility function attributes modeled are fixed ahead of time. In some cases, however, the difficulty of utility elicitation may be caused by the use of a model which is too simple to represent the application domain adequately. It may be difficult, for example, to assess the utility of an outcome involving pregnancy loss if the attribute "future pregnancy" is not modeled. In such cases, it is possible to reason about the value of extending (or refining) the model by introducing additional variables which will help in the elicitation process. We have not explored model refinement in this thesis; however, the subject is closely related to our work and the tools used are similar to the ones we apply to utility modeling in our framework.

Heckerman and Jimison (1989) suggest using *value of information* to decide whether or not a given probability or utility value should be reassessed by introducing new conditioning variables. The decision maker from whom the values are elicited is willing to provide a point estimate for the given variable, but is not confident that this value precisely reflects his beliefs or preferences. According to the proposed method, he is asked to specify a probability distribution over the variable in question. He also to find the values of these variables. The authors point out that, in simple decision problems, the decision whether or not to refine the model is usually intuitively obvious. They assume that their method will be useful when the number of variables is large and the resolution of assessment trade-offs more complicated. However, it is hard to imagine the decision maker to be able to estimate to any reasonable precision either the distribution over possible values of the given variable or the cost associated with the further assessment.

Estimating the value of refining a decision model was further explored by Poh and Horvitz (1993). One of the refinement types they consider is a *preference refinement*. They assume that in some cases we will only know that the values of utility parameters fall within a specific interval. Assuming a given distribution (uniform in their example) the value of utility refinement (i.e., of finding a precise value for the parameter) is the expected utility of the best decision given that knowledge minus the expected utility of the best decision computed for the means of every utility interval. It is not discussed how we can acquire the precise utility values. The refinements are carried out in the order of decreasing value (the value of refinement is its value of information minus the cost).

9.2.2 Utility Elicitation

Certainty Case

Jameson et al. (1995) and Linden et al. (1997) investigated the problem of eliciting partial utility models and reasoning with such models in case of certainty. They are interested in developing software agents which could function as personal assistants in tasks such as shopping and making airline reservations on the Web. They argue that the behavior of such agents should resemble that of a human assistant: in the course of the interaction, different options should be suggested to the user for evaluation, wishes. Thus, the preference elicitation process is interleaved with the presentation of specific options. Note that assessment is not always conducted in a straightforward manner: it may include both direct elicitation questions and indirect evaluation questions related to a specific option which had been presented to the user. This framework is called *evaluation-oriented information provision* in (Jameson, Schäfer, Simons, and Weis 1995) and *candidate/critique* model in (Linden, Hanks, and Lesh 1997).

In PRACMA (Jameson, Schäfer, Simons, and Weis 1995), the system attempts to help the user in choosing a used car to buy. The utility model assumes additive decomposition among top level attributes, such as "Reliability" and "Safety." In addition, several lower-level attributes are modeled. It is assumed that the subutility of a higher-level attribute can be decomposed additively into subutility functions defined over lower-level attributes. The system has a prior distribution over the user's utility function. In addition, the user's potential reactions in the course of the interaction are modeled probabilistically, dependent on his utility function and the option under discussion. The options are presented in an incremental manner, a few attributes at a time. Whenever a new option (a new car instance) is introduced, the system augments its beliefs with a representation for the user's expectations with respect to this option given the information he has been given so far. For example, if the car under discussion has low mileage for its age and the user was only told of the mileage, the system may expect a surprise at the news of car's age.

The system acquires a more accurate model of the user's preference structure using a variety of elicitation actions:

- by prompting the user with a general question and registering the information volunteered,

- by asking a direct preference elicitation question and incorporating the information contained in the answer, The goal of the system is to elicit enough preference information to be able to recommend a satisfactory car for the user without making the interaction unnecessarily long. The best elicitation action is chosen based on the extent to which it reduces the system's uncertainty about the important variables. This amounts to the reduction in the variance of the distribution over a given variable. It is not the only criterion, however. In the effort to make the interaction more natural, Pracma uses a planning algorithm to divide the conversation into phases and maintain some coherence within each phase.

A *candidate/critique agent (CCA)* (Linden, Hanks, and Lesh 1997) also assumes the additive decomposition of the utility function, but represents its belief about the user's utilities in a slightly different form. The user model in their framework consists of a set of (soft) constraints over the attributes' domains and a set of attributes' weights. A constraint is a function from the domain of an attribute to $[0, 1]$. Such a user model provides a partial ordering over all options (solutions).

The interaction proceeds as follows. On each iteration, the CCA presents several possible solutions to the user. The user can accept one of the solutions and end the interaction, add a constraint, modify an existing constraint, or adjust the importance weights for one or more attributes. This can be done either directly or through the use of a natural language or graphical user interface. The user's action provides the CCA with new information, which is used to update the model. Note that the CCA does not perform any direct elicitation actions, it only collects the preference information given by the user as a part of an evaluation of a specific option.

The most straightforward approach would be to present at each step several best options (best according to the partial user model available at the time). However, it is not possible to compute a total ordering among solutions based on a partial preference model. The authors suggest augmenting the model with default preferences. For example, if monetary considerations are involved, we can add a default preference for cheaper options; if there are no clear indications to the contrary, we the best solutions would not provide them with enough preference information and would therefore cause the interaction to be unnecessarily long. They refine the choice of the solutions by using two heuristics: preferring significantly different solutions and preferring extrema — solutions optimizing one attribute. They argue that their choice of solutions prompts the user to volunteer more preference information and speeds up the elicitation process.

Both of these algorithms differ significantly from our utility elicitation algorithm (Chapter 7) in many aspects: the assumptions about the user's utility function, the choice of default or prior over the utility space, the set of elicitation actions, and the criteria for choosing the best next action/question. Some of these differences are simply a matter of design decisions: assuming an additive decomposition of a utility function simplifies both PRACMA's and CCA's design, but is not necessary. Both systems could easily be modified to consider a larger set of utility function structures (in particular, conditionally additive and generalized additive functions). Similarly, both systems could use a learning algorithm similar to ours to acquire a prior distribution over the utility functions rather then resort to "intuitive" default assumptions. On the other hand, such effort may not be justified in low-risk domains such as shopping. Our framework was motivated by situations where the user has a lot to lose.

Some differences, however, are more fundamental. One of these is the criterion used to choose the next question. Both PRACMA and CCA use reduction in the system's uncertainty over the important variables. PRACMA does this directly, CCA by preferring extremal solutions, which is supposed to motivate the user to correct inaccurate assumptions. In contrast, our utility elicitation algorithm uses value of information.

The candidate/critique framework is, of course, impossible to use in a decision problem with uncertainty, where we not only need to find a preference ordering among outcomes, but also the risks the user is willing to take to achieve a better outcome erybody, they suggest finding a preference function in a database of previously elicited preferences that most closely matches the preference function of the given user. This closely matching function can be used as a default. They define a distance measure on partially elicited preferences (partial orders over outcomes). Each partial order defines a set of complete orders which are consistent with it. Such consistent complete orders are called linear extensions of a given partial order. The distance between two partial preference orders $\succ_1$ and $\succ_2$ is defined as the average of the distances between pairs of complete orders that are consistent with $\succ_1$ and $\succ_2$ respectively. The distance measure between complete orders is defined in turn as the probability that the two orders will disagree in their preference over two uniformly randomly selected outcomes. This measure can be approximated by the use of a Markov chain Monte Carlo technique. The measure has high variance when the information about the new user's preferences is limited (i.e., in the beginning of the elicitation process). In addition, it is not clear which defaults to use if the most closely matching function is itself partially specified.

Ha and Haddawy's framework does not assume any specific preference structure, which makes it very general. It is not clear, however, how their system could take advantage of a specific structure to reduce the size of the representation and the length of the elicitation process if some appropriate structural assumptions were warranted. The authors do not provide an algorithm to select elicitation questions.

Uncertainty Case

Eliciting utilities in the case of decision making with uncertainty was the focus of further work by Ha and Haddawy (1997, 1999).

In their first paper on this subject (1997) Ha and Haddawy present an algorithm for incremental utility elicitation. It differs from ours (Chapter 7) in the assumptions they make regarding the user's abilities to reason about the decision problem and the utility function structure. In their formulation, the main problem is the number due to the complexity of the decision problem. In addition, they assume that the utility function is purely additive, while we allow all possible additive, conditionally additive and generalized additive decompositions. Furthermore, they assume that all the subutility functions have been fully elicited — a task we deem both cognitively difficult (see Section 3.2.2) and unnecessarily time consuming.

Let $\mathbf{V} = \{V_1, V_2, \ldots, V_m\}$ be a fixed set of m attributes. Each attribute V_i has a domain Dom_{V_i} of two or more elements. For each continuous attribute V_i, $\mathrm{Dom}_{V_i} = [v^i_\perp, v^i_\top]$. For each discrete attribute V_j, $\mathrm{Dom}_{V_j} = \{v^j_\perp, \ldots, v^j_\top\}$. The form of the utility function is $U(\mathbf{V}) = \sum_{i=1}^m k_i f_i(V_i)$. Since all $f_i(V_i)$ have been elicited, all that remains to be assessed are the scaling constants $k_1, k_2, \ldots, k_m$, one for each attribute.

Ha and Haddawy consider two types of elicitation questions:

- The classical standard gamble question in which the user must choose between a lottery resulting in the best outcome $\mathbf{v}_\top$ with probability p and the worst outcome $\mathbf{v}_\perp$ with probability $1-p$ on the one hand and the outcome in which a selected attribute V_i is set to its least preferred value $v^i_\perp$ and all other attributes to their most preferred values on the other. Note that this is equivalent to a series of questions for various values of p. Once the indifference point is found, we can easily compute the scaling constant for the subutility function of the selected attribute V_i.

- A new type of question which does not involve uncertainty. Suppose that $v^j_r \in \mathrm{Dom}_{V_j}$. The question asks the user for a specific value of attribute V_i, v^i_s, for which the user would be indifferent between outcomes that yield $v^i_\perp, v^j_r$ and $v^i_s, v^j_\perp$ respectively, and that agree on all other attributes at some fixed level. Note that this question is only applicable to continuous-valued attributes. The answer to this question gives us the ratio between the scaling coefficients of the two outcomes: k_i/k_j.

to the user, the elicitation stops. Since the second type of question is more efficient in eliminating strategies, it is preferred wherever applicable. The two attributes are chosen according to their *rank correlation coefficient (RCC)*, which measures their disagreement on the set of possible strategies.

In subsequent work (1999), Ha and Haddawy relax some of the assumptions placed on the utility function structure. They require only that all utility attributes are utility independent of the rest, which results in a multilinear decomposition of the utility function (Section 3.2.4).[1] However, they maintain the assumption that all subutility functions are elicited in advance. The algorithm presented eliminates suboptimal strategies based on the partially specified utility functions and the user's answers to preference questions over pairs of outcomes.

Utility elicitation was also considered by Boutilier et al. (2001) in the context of UCP-networks (see Section 9.2.1). Similarly to Ha and Haddawy, they assume that the utility factors (subutility functions) are already elicited and only the scaling constants remain unspecified. They further assume that the structure of the UCP-network is fixed, which means that every individual in the target population must have the same utility function structure. Note that this is in sharp contrast to our framework where we allow the utility function to have different (and unknown) decompositions for different subgroups in the target population.

The structure of the elicitation process that Boutilier et al. propose closely matches the one we presented in Chapter 7. At every step, the best strategy π^* is chosen given current (partial) utility information for a given user. If this strategy satisfies a fixed stopping rule (evaluating our confidence in its optimality), the algorithm stops and outputs π^*. If not, we choose a preference elicitation question and incorporate the user's answer into our representation of his utility function. Armed with this new

[1]Note that the conditionally additive and generalized additive utility functions we consider in our framework do not necessarily exhibit utility independence for every utility attribute. Therefore, we can represent some functions that Ha and Haddawy cannot. On the other hand, we cannot take advantage of the multilinear decomposition even if it is present.

strategy chosen is the strategy that has the lowest worst case utility loss over the space of possible utility functions. The minimax criterion can also be used to establish the stopping rule: we can terminate the assessment process as soon as MMR falls below a pre-specified threshold. The questions are chosen to maximally reduce the worst case loss. Specifically, the score for each question is computed by choosing the minimum improvement (over the possible responses to the question), where the improvement is defined as the reduction in MMR. Then, the question with the highest score is selected.

While it can be argued that the worst case loss constitutes an appropriate stopping rule, especially in medical domains where the decisions concern very important issues, the use of MMR in choosing the next question seems poorly justified. It automatically eliminates all questions with even a single answer that does not improve MMR, even if other answers improve it greatly.

9.2.3 Learning from Observations

The need to learn about an agent's goals and preferences without questioning the agent directly has long been recognized.

The assumption of rationality allows us to treat a player's preference ordering as revealed by his behavior. According to the theory of *revealed preference*, first introduced by Samuelson (1938), preference ordering of a given player can be reconstructed from observation of his actions in the world. Consider an agent faced with a choice between move d_i resulting (deterministically) in outcome o_i and move d_j resulting in outcome o_j. If the agent is observed to choose d_i, we conclude that outcome o_i is preferred to outcome o_j. Given enough observations, we can reconstruct the agent's entire preference structure.

Following this early work, many researchers have attempted to create models of human behavior. In this section, we will focus on the work which is closest to ours, A large body of work in probabilistic user modeling concerns discovering users' goals, needs, beliefs, and intentions. In most cases, these goals are treated as state variables. There is a direct probabilistic dependence between goals and actions, so users' actions provide direct evidence about their goals. This approach is very different from ours. We want to learn the underlying utility function of a given user. If the user is rational, he will choose the goals to pursue according to his utility function. Once we know the utility function, we are able to predict the user's future actions not only in the context of another instance of a decision problem we have observed, but also in the context of new decision problems. Such an extrapolation from one decision problem to another is impossible within the framework of a direct probabilistic dependence.

The area of probabilistic user modeling is very large; its full description is beyond the scope of this dissertation. Here, we briefly mention two recent projects, which are both interesting and representative. They also provide many references for the interested reader to follow.

In a recent work, Horvitz et al. (1998) discuss the task of predicting goals and needs of software users in order to provide them with assistance. Their application, Lumière, uses a probabilistic model of the user's background, competency profile, and history of software use. Based on these user characteristics and observed actions it attempts to guess what task the user is trying to accomplish and whether he needs assistance. A Lumière prototype served as the basis for the *Office Assistant* in Microsoft Office '97.

Goals in Lumière are target tasks at the focus of the user's attention. Needs are understood as information or automated actions that will reduce the time or effort required to achieve goals. Goals depend (probabilistically) on the context and the task history. Needs depend on goals and the user's competence. Needs directly influence patterns of activity that the system is able to observe. Once we have evidence about the patterns of activity, we can compute the probability that the user is engaged in a given task using standard inference. A fully specified utility model allows us to activities observed by Lumière. The authors construct a simple probabilistic model to represent the dependence of word use on the task at hand. Using this model, we can infer the user's goal given the choice of words in the query.

Discovering a user's intentions based on his utterances in a conversation is the subject of work by Paek and Horvitz (Paek and Horvitz 2000; Horvitz and Paek 2000). In the *Quartet* and *DeepListener* systems, they model conversation as action under uncertainty. In human conversations, the participants typically make sure that what they say is attended to, heard and understood by the others. Quartet models all of these aspects of the conversation using a probabilistic framework. One of the modules of the system, the *intention module*, deals directly with the recognition of the intentions of the speaker. Again, the system assumes a direct dependence between the intentions and utterances: thus, evidence about users' actions is propagated via probabilistic inference to update the system's beliefs about the user's goals.

Modeling another agent's beliefs and their impact on his decision making process was studied by Milch and Koller (2000). They introduce a *probabilistic epistemic logic (PEL)* that uses Bayesian networks as a compact representation for an agent's beliefs. The basic Bayesian network model of the domain can be augmented with nodes corresponding to indicator variables for more complex formulas, including those with modal operators. Using this framework, it is possible to evaluate arbitrary PEL formulas.

If it can be assumed that the agent is rational, Milch and Koller propose an extension of this framework which allows us to model the agent's decision process and derive conditional probability distributions over possible actions the agent may take. Then, having observed some of his actions, we can reach conclusions about events in the domain we were not able to observe. The framework can also be extended to allow for some uncertainty about the agent's preferences, represented by discrete-valued nodes modifying the way utility depends on other variables. In such cases, observing the agent's actions leads us to conclusions about his utilities.

More closely related to our framework is the problem of inferring an agent's utility (reward) function based on behavioral observations in *Markov Decision Processes (MDPs)*. The problem was described and analyzed by Ng and Russell (2000). They refer to it as the *inverse reinforcement learning* problem.

They characterize the reward function space which is consistent with the observed behavior assuming that the agent is following the optimal strategy. Our derivation of constraints in the utility space for fully observed strategies (Equation (8.1) in Section 8.2.1) is a direct analogue of their formulation. Ng and Russell note that the space of possible reward functions is infinite and includes many degenerate functions, such as all functions that assign the same value to all states.

They propose a set of heuristics to choose a reward function π from the feasible region. First, they favor solutions that make any single-step deviation from π as costly as possible. Second, they prefer strategies which assign small rewards to most states. Such strategies are considered "simpler." An adjustable parameter balances the influence of these two heuristics. They consider both the simple case of observing the optimal policy directly and the more realistic case of observing only partial strategies, or "trajectories" in the MDP.

The main difference between our approach and that of Ng and Russell lies in the selection of the reward function from the feasible region. We assume a probability density function over the utility space, which can be conditioned on the observed behavior to derive a more informed posterior over the feasible region. Having this posterior density allows us easily to compute the mean of the distribution which can be used as a representative utility (reward) function. We can also sample from the posterior using the MCMC algorithm described in Section 6.5 and use the set of samples as an approximation of the posterior distribution.

pursue.

9.3.1 Utility Modeling

We have restricted our attention to utility functions exhibiting various kinds of additive independence. While we believe that factorizations based on generalized additive independence apply to a broad range of functions in many real-life domains, we would like to extend our work to utility functions in multiplicative or multilinear form.

So far, most work (including ours) has focused on notions of independence at the level of variables. In probabilistic settings, this notion has been refined to that of *context-specific independence* (Boutilier, Friedman, Goldszmidt, and Koller 1996), which allows independence of two variables X and Y in the context of a particular value z of a third variable Z, but not in the context of a value z' for Z. An analogous notion can also be defined for utilities. We hope to extend our approach to handle these more refined factorizations of utility functions.

In another extension, we hope to capture relations between utility variables and other (environmental) variables. For example, it has been observed that people who have experienced an outcome tend to assign it a higher utility value than those for whom the outcome is imaginary (Lenert, Treadwell, and Schwartz 1999). This type of correlation can be represented very naturally as a dependence in our probabilistic model; we hope to extend our approach to handle this type of situation.

A less obvious application is to represent the probability that a user's preferences will change over time. We believe that the tools provided by probabilistic models — value of information, statistical learning, and others, will turn out to be extremely useful in this new type of modeling.

Finally, we have assumed that the prior over utility functions was learned in advance, and then applied as is to each new user. It could prove to be valuable to use information gained during the use of the system — the very partial data obtained by asking a handful of questions or viewing the behavior of the different agents —

9.3.2 Applications

In our adaptive elicitation algorithm, we select questions with the highest value of information, assuming that the cost of asking a question is constant. However, some questions are cognitively more difficult than others. Questions near the indifference point are hard, a second consecutive question about the same outcome is "cheaper" (i.e., easier to answer) than a question about an outcome discussed a few questions back, etc. It would be easy to incorporate the cognitive cost of questions into the value of information computation. A somewhat more subtle extension is to expand the range of questions that we allow to incorporate more general lotteries.

Note that the explicit modeling of our uncertainty over the utility function allows us to reason about the value of various kinds of information at the same time. We can evaluate the value of information of answers to specific utility questions and compare it to the value of information of other findings in the system, such as discovering the value of a state variable. For example, in some cases it may be more valuable (bring us closer to making the optimal decision) to conduct a medical test rather than a utility elicitation procedure.

In our game-playing application, we can think of the interaction between the informed and oblivious player as a repeated game, with each interaction giving the informed agent more information about the utility function of the oblivious agent. This view immediately leads to an exploration/exploitation tradeoff. If the informed agent can choose the parameters of each new game instance, he must decide at every step whether to select a parameter setting giving him a large payoff with high probability (thus exploiting his knowledge about his opponent's utility function) or a setting for which he cannot predict his opponent's actions with high accuracy (to explore the utility space further). This tradeoff would be interesting to analyze.

While in many domains it is reasonable to assume that the oblivious agent is oblivious agent. Assuming that both agents are perfect reasoners would, of course, lead to infinite cycles of mutual anticipation of each other's moves. However, the assumption of full rationality is not realistic either. We can, perhaps, find some middle ground between perfect reasoning capabilities and complete oblivion which would be appropriate from the modeling perspective and still computationally tractable.

Bibliography

Allais, M. (1953). Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica 21*, 503–546.

Anderson, N. H. (1974a). Algebraic models in perception. In *Handbook of perception*, Volume 2, pp. 215–298. New York, New York: Academic Press.

Anderson, N. H. (1974b). Information integration theory: a brief survey. In *Contemporary developments in Mathematical Psychology*, Volume 2, pp. 236–305. San Francisco, California: W. H. Freeman and Company.

Anderson, N. H. (1976). How functional measurement can yield validated interval scales of mental quantities. *Journal of Applied Psychology 61*(6), 677–692.

Anderson, N. H. and J. C. Shanteau (1970). Information integration in risky decision making. *Journal of Experimental Psychology 84*, 441–451.

Applegate, D. and R. Kannan (1991). Sampling and integration of near log-concave functions. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing (STOC '91)*, pp. 156–163.

Bacchus, F. and A. Grove (1995). Graphical models for preference and utility. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pp. 3–10.

Bacchus, F. and A. Grove (1996). Utility independence in qualitative decision theory. In *Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning (KR '96)*, pp. 542–552.

Bárány, I. and Z. Füredy (1986). Computing the volume is difficult. In *Proceedings* *search 30*(5), 961–981.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* New York, New York: Oxford University Press.

Boutilier, C., F. Bacchus, and R. Brafman (2001). UCP-networks: a directed graphical representation of conditional utilities. In *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pp. 56–64.

Boutilier, C., R. Brafman, C. Geib, and D. Poole (1997). A constraint-based approach to preference elicitation and decision making. In *AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning.*

Boutilier, C., R. Brafman, H. Hoos, and D. Poole (1999). Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 71–80.

Boutilier, C., N. Friedman, M. Goldszmidt, and D. Koller (1996). Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI '96)*, pp. 115–123.

Chajewska, U., L. Getoor, J. Norman, and Y. Shahar (1998). Utility elicitation as a classification problem. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 79–88.

Chajewska, U. and D. Koller (2000). Utilities as random variables: density estimation and structure discovery. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pp. 63–71.

Chajewska, U., D. Koller, and D. Ormoneit (2001). Learning an agent's utility function by observing behavior. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, pp. 35–42.

Chajewska, U., D. Koller, and R. Parr (2000). Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference*

(Eds.), *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press.

Chickering, D. M. and D. Heckerman (1996). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. Technical Report MSR-TR-96-08, Microsoft Research.

Coombs, C. H., R. M. Dawes, and A. Tversky (1970). *Mathematical Psychology*. Englewood Cliffs, New Jersey: Prentice-Hall.

DeGroot, M. (1970). *Optimal Statistical Decisions*. New York, New York: McGraw-Hill.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B 39*, 1–38.

Fischer, G. W. (1979). Utility models for multiple objective decisions: Do they accurately represent human preferences? *Decision Sciences 10*, 451–479.

Fishburn, P. C. (1970). *Utility Theory for Decision Making*. New York, New York: John Wiley & Sons.

Fishburn, P. C. (1982). *The Foundations of Expected Utility*. Dordrecht, The Netherlands: D. Reidel Publishing Company.

French, S. (1986). *Decision Theory: An Introduction to the Mathematics of Rationality*. Chichester, U.K.: Ellis Horwood Limited.

Friedman, N. (1997). Learning Bayesian networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, pp. 125–133.

Fromberg, D. G. and R. L. Kane (1989a). Methodology for measuring health-state preferences—I: Measurement strategies. *Journal of Clinical Epidemiology 42*(4), 345–354.

bridge, Massachusetts: MIT Press.

Fudenberg, D. and J. Tirole (1991). *Game Theory*. Cambridge, Massachusetts: MIT Press.

Geiger, D. and D. Heckerman (1994). Learning Gaussian networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '94)*, pp. 235–243.

Ha, V. and P. Haddawy (1997). Problem-focused incremental elicitation of multi-attribute utility models. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '97)*, pp. 215–222.

Ha, V. and P. Haddawy (1998). Toward case-based preference elicitation: similarity measures on preference structures. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 193–201.

Ha, V. and P. Haddawy (1999). A hybrid approach to reasoning with partially elicited preference models. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 263–270.

Ha, V., P. Haddawy, and J. Miyamoto (2001). Similarity measures on preference structures, part II: utility functions. In *Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pp. 186–193.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 97–109.

Heckerman, D. (1996). A tutorial on learning Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research.

Heckerman, D. and E. Horvitz (1998). Inferring informational goals from free-text queries: a Bayesian approach. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 230–238.

Horvitz, E., J. Breeze, D. Heckerman, D. Hovel, and K. Rommelse (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 256–265.

Horvitz, E. and T. Paek (2000). DeepListener: harnessing expected utility to guide clarification dialog in spoken language systems. In *Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP 2000)*.

Howard, R. A. and J. E. Matheson (1984). Influence diagrams. In R. A. Howard and J. E. Matheson (Eds.), *The Principles and Applications of Decision Analysis*. Menlo Park, California: Strategic Decisions Group.

Jameson, A., R. Schäfer, J. Simons, and T. Weis (1995). Adaptive provision of evaluation-oriented information: tasks and techniques. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, pp. 1886–1893.

Jansen, S. J. T., A. M. Stiggelbout, P. P. Wakker, M. A. Nooy, E. M. Noordijk, and J. Kievit (2000). Unstable preferences: a shift in valuation or an effect of the elicitation procedure? *Medical Decision Making 20*(1), 62–71.

Jensen, F., F. V. Jensen, and S. L. Dittmer (1994). From influence diagrams to junction trees. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '94)*.

Jimison, H. B., L. M. Fagan, R. D. Shachter, and E. H. Shortliffe (1992). Patient-specific explanation in models of chronic disease. *AI in Medicine 4*, 191–205.

Jordan, M. I. (Ed.) (1998). *Learning in Graphical Models*. Kluwer Academic Publishers.

Kahneman, D. and A. Tversky (1979). Prospect theory: an analysis of decision under risk. *Econometrica 47*(2), 263–291.

Kuppermann, M., J. D. Goldberg, R. F. Nease Jr., and A. E. Washington (1999). Who should be offered prenatal diagnosis? The thirty-five year-old question. *American Journal of Public Health 89*(2), 160–163.

Kuppermann, M., S. Shiboski, D. Feeny, E. P. Elkin, and A. E. Washington (1997, Jan–Mar). Can preference scores for discrete states be used to derive preference scores for an entire path of events? *Medical Decision Making 17*(1), 42–55.

La Mura, P. and Y. Shoham (1999). Expected utility networks. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 366–373.

Lenert, L. A., D. J. Cher, M. K. Goldstein, M. R. Bergen, and A. Garber (1998). The effect of search procedures on utility elicitations. *Medical Decision Making 18*(1), 76–83.

Lenert, L. A., J. R. Treadwell, and C. E. Schwartz (1999). Associations between health status and utilities implications for policy. *Medical Care 37*(5), 479–489.

Lenert, L. A., J. Ziegler, T. Lee, C. Unfred, and R. Mahmoud (2000). The risks of multimedia methods: effects of actor's race and gender on preferences for health states. *Journal of the American Medical Informatics Association 7*(2), 177–185.

Linden, G., S. Hanks, and N. Lesh (1997). Interactive assessment of user preference models: the automated travel assistant. In *Proceedings of the Sixth International Conference on User Modeling (UM '97).*

Lipscomb, J. (1989). Time preference for health in cost-effectiveness analysis. *Medical Care 27*, 233–253.

Llewellyn-Thomas, H., H. J. Sutherland, R. Tibshirani, A. Ciampi, J. E. Till, and N. F. Boyd (1982). The measurement of patients' values in medicine. *Medical Decision Making 2*(4), 449–462.

games. *Handbook of Computational Economics 1*, 87–142.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics 21*, 1087–1092.

Milch, B. and D. Koller (2000). Probabilistic models for agents' beliefs and decisions. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pp. 389–396.

Nash, J. F. (1951). Noncooperative games. *Annals of Mathematics 54*, 289–295.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

Ng, A. Y. and S. Russell (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*.

O'Leary, J. F., D. L. Fairclough, M. K. Jankowski, and J. C. Weeks (1995). Comparison of time-tradeoff utilities and rating scale values of cancer patients and their relatives. *Medical Decision Making 15*(2), 132–137.

Paek, T. and E. Horvitz (2000). Conversation as action under uncertainty. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pp. 455–464.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Francisco, California: Morgan Kauffman.

Poh, K. L. and E. Horvitz (1993). Reasoning about the value of decision-model refinement: methods and application. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI '93)*, pp. 174–182.

Massachusetts: MIT Press.

Read, J. L., R. J. Quinn, D. M. Berwick, H. V. Fineberg, and M. C. Weinstein (1984). Preferences for health outcomes: comparison of assessment methods. *Medical Decision Making 4*(3), 315–329.

Rissanen, J. (1987). Stochastic complexity (with discussion). *Journal of the Royal Statistical Society, Series B 49*, 223–239 and 253–265.

Samuelson, P. A. (1938). A note on the pure theory of consumers' behavior. *Economica 5*, 61–71.

Sanders, G. D. (1998). Automated creation of clinical-practice guidelines from decision models. Technical Report SMI-09-0712, Stanford Medical Informatics, Stanford University, Stanford, California. Ph.D. Dissertation.

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical Association 46*, 55–67.

Savage, L. J. (1954). *Foundations of Statistics*. New York, New York: John Wiley & Sons.

Schkade, D. A. and D. Kahneman (1998). Does living in California make people happy? A focusing illusion in judgments of life satisfaction. *Psychological Science 9*(5), 340–346.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*, 461–464.

Selten, R. (1991). Anticipatory learning in two-person games. In *Game Equilibrium Models*, Volume 1, pp. 98–153. Berlin-Heidelberg-New York: Springer-Verlag.

Shachter, R. and C. Kenley (1989). Gaussian influence diagrams. *Management Science 35*, 527–550.

Shoham, Y. (1997b). Two senses of conditional utility. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '97)*.

Stiggelbout, A. M., M. J. C. Eijkemans, G. M. Kiebert, J. Kievit, J.-W. H. Leer, and J. C. J. M. D. Haes (1996). The "utility" of the visual analog scale in medical decision making and technology assessment. *International Journal of Technology Assessment in Health Care 12*(2), 291–298.

Stiggelbout, A. M., G. M. Kiebert, J. Kievit, J. W. F. Habbema, and J. C. J. M. D. Haes (1995). The "utility" of the time trade-off method in cancer patients: feasibility and proportional trade-off. *Journal of Clinical Epidemiology 48*(10), 1207–1214.

Strang, G. (1980). *Linear Algebra and Its Applications* (2nd ed.). Orlando, Florida: Academic Press.

Tatman, J. A. and R. D. Shachter (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics 20(2)*, 365–379.

Torrance, G. W. (1976). Social preferences for health states: an empirical evaluation of three measurement techniques. *Socio-economic Planning Sciences 10*, 129–136.

Torrance, G. W., W. H. Thomas, and D. L. Sackett (1972). A utility maximization model for evaluation of health care programs. *Health Services Research 7*, 118–133.

Tversky, A. and D. Kahneman (1986a). Judgment under uncertainty: heuristics and biases. In *Readings in Uncertain Reasoning*, pp. 32–39. San Mateo, California: Morgan Kauffman.

Tversky, A. and D. Kahneman (1986b). Rational choice and the framing of decisions. In *Readings in Uncertain Reasoning*, pp. 91–104. San Mateo, California: Morgan Kauffman.

von Neumann, J. and O. Morgenstern (1947). *Theory of Games and Economic Behavior* (2nd ed.). Princeton, New Jersey: Princeton University Press.