

Elicitation of Factored Utilities

*Darius Braziunas
and Craig Boutilier*

■ *The effective tailoring of decisions to the needs and desires of specific users requires automated mechanisms for preference assessment. We provide a brief overview of recent direct preference elicitation methods: these methods ask users to answer (ideally, a small number of) queries regarding their preferences and use this information to recommend a feasible decision that would be (approximately) optimal given those preferences. We argue for the importance of assessing numerical utilities rather than qualitative preferences and survey several utility elicitation techniques from artificial intelligence, operations research, and conjoint analysis.*

As automated decision support becomes increasingly accessible in a wide variety of AI applications, addressing the preference bottleneck is vital. Specifically, since the ability to make reasonable decisions on behalf of a user depends on that user's preferences over outcomes in the domain in question, AI systems must assess or estimate these preferences before making decisions. Designing effective preference assessment techniques to incorporate such user-specific considerations (that is, breaking the preference bottleneck) is one of the most important problems facing AI.

In this brief survey, we focus on explicit *elicitation* techniques where a system actively queries a user to glean relevant preferences.¹ Preference elicitation is difficult for two main reasons. First, many decision problems have exponentially sized outcome spaces, defined by the possible values of outcome attributes. As an illustrative example, consider sophisticated flight selection: possible outcomes are defined by attributes such as trip cost, departure time, return time, airline, number of connections, flight length, baggage weight limit, flight class, (the possibility of) lost luggage, flight delays, and other stochastic outcomes. An ideal decision support system should be able to use, for example, precise flight delay statistics and incorporate a user's relative tolerance for delays in making recommendations. Representing and eliciting preferences for all outcomes in a case like this is infeasible given the size of the outcome space. A second difficulty arises due to the fact that quantitative strength of preferences, or *utility*, is needed to trade off, for instance, the odds of flight delays with other attributes. Unfortunately, people are notoriously inept at quantifying their preferences with any degree of precision, adding to the challenges facing automated utility elicitation.

Within AI, decision analysis, operations research, marketing, and other areas of research, a number of elicitation techniques have been devel-

oped that attempt to address these problems. In this article, we survey a selection of these techniques with a specific focus on two key themes. First, we focus on the use of *factored utility models* (Fishburn 1967; Keeney and Raiffa 1976): these models decompose a utility function over outcomes (all combinations of attributes) into *local utility functions* over subsets of attributes, which are then combined to produce a “global” utility function. This breaks the combinatorial explosion in representation size and can dramatically reduce the number of preference parameters that need to be assessed. Widely used *additive models* are a prime example of this.

Second, we deal with methods that provide an explicit representation of *utility function uncertainty*. An important trend in preference elicitation, especially within AI, is the recognition of the trade-off between obtaining more utility information—and thus making a better decision—and the cost of further elicitation effort. Elicitation costs can be cognitive (human effort in answering questions), computational (for example, calculating a value of certain alternative by performing intensive optimization or simulation), financial (for example, hiring a team of experts to analyze potential business strategies), or involve time and opportunity costs. Often elicitation costs will outweigh the “value” of the information it provides. In such a case, decisions should be made with *partial* utility information. For instance, suppose a travel-planning agent has narrowed the list of potentially optimal flights (measured by, say, willingness to pay) to Flight *A*, whose utility is between \$800 and \$1000, and Flight *B*, whose utility is between \$980 and \$1200. Further utility elicitation *could* determine that *A* is optimal, but at best it can be only \$20 “better” than *B* (whereas *B* may be up to \$400 better than *A*). If the cost (time, cognitive, annoyance, and so on) of differentiating these two flights further is greater than \$20, it is best to simply recommend Flight *B*. However, such deliberations require some explicit representation of utility function uncertainty.

This article provides a brief survey of decision-making and elicitation techniques for such factored utility models, with an emphasis on those methods that maintain an explicit representation of user utility function uncertainty and make decisions with this representation (as opposed to, say, point estimates of utility functions). As a result, heuristic techniques (for example, based on heuristic measures of similarity or product navigation methods that guide a user through a database with no quality guarantees) are not reviewed here. Moreover, we focus our discussion of elicitation on only a few possible interaction or query methods whose interpretation is decision-theoretically unambiguous. We do not claim that these are the

most natural or effective interaction modes; for instance, recent work on example critiquing (Viappiani, Faltings, and Pu 2006) offers promise for more natural user-directed interaction. But more research is needed to determine the precise interpretation of, say, example critiquing or “outcome tweaking” in sound decision-theoretic terms. Our presentation will focus on intuitive examples and refer the reader to the relevant literature for key mathematical details.

We begin with a review of additive and generalized additive utility models for discrete domains and then introduce a few fundamental preference queries upon which the elicitation methods discussed here are based. The following two sections describe two explicit representations of uncertainty as well as associated methods for elicitation: Bayesian techniques, in which a density is maintained over possible utility functions; and feasible set methods, in which utility function uncertainty is represented by defining the space of feasible utility functions. We conclude with a short summary and discussion of future research directions.

Multiattribute Utility Representations

We begin with a brief informal review of multiattribute utility theory (MAUT) and the two classic factored utility models: additive and generalized additive. We also motivate the importance of (numerical) utilities over qualitative preferences in automated decision support and elicitation. The text of Keeney and Raiffa (1976) remains the definitive exposition for many of these concepts.

Utility Functions

We suppose a user faces the following decision problem: a decision d must be chosen from a set D of possible decisions, such as the set of all flights from Toronto to Seattle on a specific day. We take D to be finite for this review. Each decision $d \in D$ induces a distribution over a (finite) set of possible outcomes defined over attributes of interest to the user. Let attributes X_1, X_2, \dots, X_n each have finite domains; the set of all outcomes X is the Cartesian product of the attribute domains. In our simple example, we will use the following attributes: flight Class (which takes values first, business, coach), number of Connections (0, 1, 2), total flight Duration in hours (3, 4, 5, 6, 7+), and flight Delay in hours (0, 0–1, 1–2, 2–3, 3–5). Note that the flight choice uniquely determines the first three attributes; the fourth attribute is stochastic, with the probability of each outcome determined by flight statistics. Obviously, many other deterministic (for example, airline, nominal arrival time, and so on) and stochastic (for example, lost luggage, cancellation, actual arrival time) attributes may be includ-

ed as well. In addition, flight Cost is a major attribute of interest, but we treat this separately below.

The preferences of a user, on whose behalf decisions are made, are captured by a utility function $u: X \mapsto \mathbb{R}$. A utility function serves as a quantitative representation of strength of preferences. While a purely qualitative ranking of (in our example) flight options may seem to suffice, there are several important reasons to use quantitative utility functions. First, it is useful to trade off the relative desirability of flights with their price. Indeed, it is often reasonable to make a *quasilinearity* assumption: the utility of an outcome x is given by its “objective” utility $u(x)$ less its price. This allows one to focus on the underlying “value” of specific flight options during elicitation. Without strength of preference, trade-off against price is not possible: if Flight A is preferred to B (independent of price), but B is cheaper than A , some knowledge of the degree to which A is preferred is required to make a choice. Second, when some outcome attributes are stochastic, again trade-offs involving strength of preference are needed. For example, suppose A and B are identical on all attributes except odds of delay: A will almost certainly not be late while B has a 25 percent chance of being 30 minutes late and a 10 percent chance of arriving 30 minutes early. Without assessing strength of preference for actual arrival time, the trade-off between A and B cannot be made. Finally, as we will see later, since many utility elicitation schemes rarely insist on pinning down user utilities with complete precision, the use of quantitative utilities allows one to provide error or quality estimates when making decisions without precise utilities in a way that is not possible with qualitative preferences.

Importantly, utilities can be seen as reflecting qualitative preferences over lotteries, or distributions over outcomes (Keeney and Raiffa 1976), with one lottery preferred to another if and only if its expected utility is greater. Let $x^\top \in X$ denote the best outcome for a specific user (for example, Dur = 3hrs, Class = first, Conn = 0, Delay = 0) and x^\perp the worst (for example, Dur = 6hrs, Class = coach, Conn = 2, Delay = 3–5).² We use $\langle p, x^\top; 1-p, x^\perp \rangle$ to denote the lottery or standard gamble where the best outcome x^\top is realized with probability p , and x^\perp with probability $1-p$; we refer to the best and worst outcomes as *anchor* outcomes. Since utility functions are unique up to positive affine transformations, it is customary to set the utility of the best outcome x^\top to 1, and the utility of the worst outcome x^\perp to 0. If a user is indifferent between some outcome x and the standard gamble $\langle p, x^\top; 1-p, x^\perp \rangle$, then $u(x) = p$. Thus lotteries give us a means to calibrate utilities (as we will see below) using comparisons of outcomes to gambles. Such utilities can also then be calibrated against price.

Additive and Generalized Additive Utilities

In many practical applications, specifying the utility $u(x)$ of each outcome $x \in X$ is infeasible since the outcome space is exponential in the number of attributes of interest. Fortunately, preferences often exhibit internal structure that can be used to express u concisely. *Additive independence* (Keeney and Raiffa 1976) is commonly assumed in practice; in this case u can be written as a sum of single-attribute subutility functions:

$$u_A(x) = u_1(x_1) + u_2(x_2) + \dots + u_n(x_n). \quad (1)$$

In our example, we would define separate subutilities for each attribute Dur, Class, Conn, and Delay, and the utility of any outcome would be given by the sum of the subutilities. For example, $u_A(4\text{hrs}, \text{coach}, 0, 1\text{hr}) =$

$$u_{\text{Dur}}(4\text{hrs}) + u_{\text{Class}}(\text{coach}) + u_{\text{Conn}}(0) + u_{\text{Delay}}(1\text{hr})$$

(see figure 1a for a graphical illustration). An additive decomposition is possible if and only if user preferences over each attribute are independent of the values of other attributes.³

Additive models, although popular in practice, are restrictive in their assumptions of attribute independence. A more general utility decomposition, based on *generalized additive independence* (GAI), has recently gained attention in AI because of its additional flexibility (Bacchus and Grove 1995; Boutilier, Bacchus, and Brafman 2001; Gonzales and Perny 2004; Braziunas and Boutilier 2005; Boutilier et al. 2006; Engel and Wellman 2007; Braziunas and Boutilier 2007).⁴ It can model “flat” utility functions with no internal structure as well as additive utilities. Most realistic problems arguably fall somewhere between these two extremes. For example, for many users the strength of preference for Class will depend on Dur (with preference for Class=business increasing with duration). In such a case, an additive utility function cannot represent the true preferences. In contrast, a GAI function with three factors is more expressive:

$$u_{\text{GAI}}(5\text{hrs}, \text{business}, 1, 0\text{hr}) = u_{\text{Delay}}(0\text{hr}) + u_{\text{Dur,Class}}(5\text{hrs}, \text{business}) + u_{\text{Class,Conn}}(\text{business}, 1).$$

One can intuitively think of such factors as a grouping of preferentially dependent attributes. Figure 1b illustrates this GAI model and its contrast with the additive model.

More generally, assume a collection of m factors; each factor j is associated with a subset of attributes I_j . We define $X_j = \times_{i \in I_j} \text{Dom}(X_i)$ to be the set of partial outcomes (or suboutcomes) restricted to attributes in factor j ; $x_j \in X_j$ is a particular instantiation of attributes in factor j (such as 5hrs, business in the first factor of our example). GAI models (Fishburn 1967; Bacchus and Grove 1995) additively decompose a utility function over possibly overlapping factors. The factors are generalized additively inde-

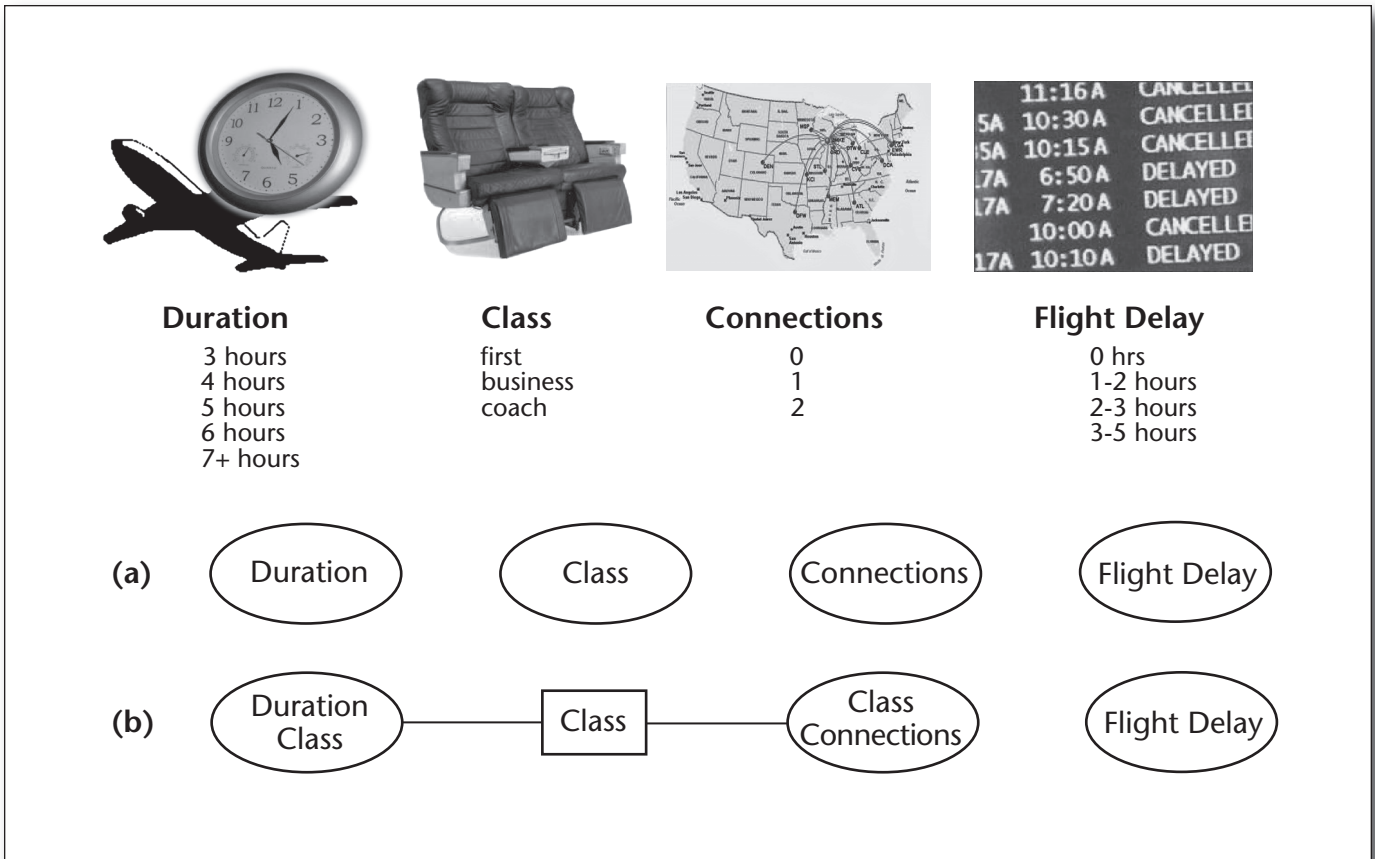


Figure 1. A Graphical Representation of the (a) Additive and (b) GAI Model from Our Four-Attribute Example.

Each node (oval) is a factor comprising a set of attributes (by definition, single attributes in the additive case). GAI factors are linked if their attributes overlap, with shared attributes appearing in square boxes.

pendent if and only if the user is indifferent between any two lotteries with the same marginals on each factor. Furthermore, if GAI holds, the utility function can be written as a sum of real-valued functions u_j such that (Fishburn 1967):

$$u_{GAI}(x) = u_1(x_1) + \dots + u_m(x_m).$$

We shall refer to u_j as a *subutility* function for factor j .

Utility Function Semantics

The presence of additive or GAI structure in preferences allows us to represent utility functions compactly. However, user utility functions are rarely known a priori; they must be assessed based on observed user preferences (for example, responses to preference queries). Therefore, to ensure a consistent interpretation of user preferences—and to design unambiguous queries for use by elicitation schemes—utility function parameters should have a well-defined decision-theoretic interpretation. In both additive and GAI models, parts of the utility function can be assessed locally: this involves querying only single attributes or

small subsets of attributes. Other parts require global assessment involving, for example, the comparison of complete outcomes over all attributes. We illustrate the distinctions with additive models and provide a brief description of similar considerations for GAI models.

Assume an additive model as defined in equation 1. We can conveniently decompose the subutility functions $u_j(x_j)$ into local value functions (LVFs) v_j and scaling constants λ_j , so that $u_j(x_j) = \lambda_j v_j(x_j)$. Then,

$$u_A(x) = \lambda_1 v_1(x_1) + \dots + \lambda_n v_n(x_n). \tag{2}$$

This allows us to separate the representation of preferences into local parameters v_j and global parameters λ_j (Keeney and Raiffa 1976). Significantly, LVFs reflect only the relative strength of preference among single attribute values, and thus can be defined using “local” lotteries that involve no other attributes. Without loss of generality, we can set $v_i(x_i^+) = 1$ and $v_i(x_i^-) = 0$, where x_i^+ and x_i^- are the best and worst levels of attribute i . In our additive example, assume the best value for Class is first, so $v_{Class}(\text{first}) = 1$, and the worst value is coach, so $v_{Class}(\text{coach}) = 0$. If a user is indifferent between

flying business for sure and the lottery in which he or she gets first class with 70 percent and coach with 30 percent, then the LVF for class has $v_{Class}(\text{business}) = 0.7$. Critically, the user does not have to consider the values of the remaining attributes. Under the additive model, this preference holds for any fixed value of the other attributes. Thus we can assess the relative strength of preference for each value of each attribute independently, without reference to other attribute values. This focus on preferences over individual attributes has tremendous practical significance, because people have difficulty taking into account more than five or six attributes at a time (Green and Srinivasan 1978).

The scaling constants λ_i are required to properly calibrate LVFs across attributes. If a user has very strong feelings about the number of connections (for example, is willing to sacrifice the preferred value of most other attributes to ensure a direct flight), then the differences in Conn should have a much stronger influence on the relative utility of an outcome than differences in Class. This is reflected in the scaling constants, which must, by their very nature, be determined in a “global” fashion, involving queries that use all attributes.

Assessment of scaling constants involves the notion of a *reference outcome*, denoted by $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$. The reference outcome can be any full outcome. It is common to choose the worst outcome x^\perp as x^0 ; but generally, any outcome that is especially salient for the user (for example, the most commonly taken flight) will prove useful, and it can be best to let the user choose this. In our example, let $x^0 = (\underline{4\text{hrs}}$, business, 0, 30min) be the reference outcome (we indicate reference values by underlining). We now define $x^{\top i}$ to be the full outcome where the i th attribute is set to its best level while other attributes are fixed at their reference levels; we denote its utility as u_i^\top . In our example, $x^{\top \text{Class}} = (4\text{hrs}$, first, 0, 30min). $x^{\perp i}$ and u_i^\perp are defined similarly. It suffices to define $\lambda_i = u_i^\top - u_i^\perp$. As a consequence, to fully assess scaling constants λ_p , we need only elicit the utilities of $2n$ global outcomes ($x^{\top i}$ and $x^{\perp i}$ for each attribute i). We discuss specific query types in the next section. The ease of representation and assessment makes additive utility the model of choice in most practical applications.

The key difference between additive and GAI models with regard to elicitation lies in the semantics of subutility functions u_i . In additive models, the quantities $u_i(x_i) = \lambda_i v_i(x_i)$ have a clear decision-theoretic meaning and are unique assuming straightforward normalization of LVFs and scaling constants. In contrast, GAI subutility functions are not unique and, in the absence of further qualifications, do not have a well-defined semantic interpretation. Intuitively, the overlap in factors means

that a change in one subutility function can be compensated for by a change in another with which it shares an attribute, leaving the overall utility function intact. Indeed, in general there are infinitely many valid GAI decompositions of a utility function (Braziunas and Boutilier 2005). Fishburn (1967) shows how the use of a reference outcome (as defined previously) can be used to provide a unique canonical decomposition of a GAI function (with an implicit elicitation process that involves global queries). Braziunas and Boutilier (2005) demonstrate how GAI structure can be exploited further to determine the LVFs for GAI models using local queries only and scaling constants using a small number of global queries.

To illustrate, consider our GAI flight example. We cannot simply ask a local query about the relative utilities of the instantiations of factor 1 (Dur, Class): because of the overlap with factor 2 (Class, Conn), a user could assign many different relative preference strengths in factor 1 by compensating for these in factor 2 (thus giving rise to the same utility function in each case). However, if we fix the value of Conn to its reference level (Conn = 0) when asking local queries about the LVF for factor 1, this is sufficient to cut off any influence on the user’s strength of preference from the values in other factors. In general, we can define a unique *conditioning set* for each factor consisting of a small number of attributes (Braziunas and Boutilier 2005). When any local query regarding the LVF for a factor i is asked, the user is to suppose that the attributes in the conditioning set are fixed to their reference levels. In our example, for the (Dur, Class) factor, queries about the relative preference of two instantiations (for example, 4hrs,coach versus 5hrs,business) would be posed in the context of Conn = 0. As a result, LVFs can be elicited using the local queries as in the additive model, but with a small number of extra attributes fixed to provide context. The scaling constants can be estimated in exactly the same fashion as in the additive model (with calibration of $2m$ global outcomes, that is, two per factor). We refer to Fishburn (1967), Gonzales and Perny (2004), and Braziunas and Boutilier (2005) for further details.

Elicitation Queries

Additive and GAI models and the semantics of their parameters naturally lend themselves to elicitation of these parameters using direct utility queries, where a user is asked to assess the global (or local) utility of an outcome (or suboutcome over a single attribute or factor value). Direct utility queries, however, require direct calibration on the part of the user. Equivalently, standard gamble queries can be used, where a user is asked to assess the probability p with which he or she is (roughly)

indifferent between some (local or global) outcome x and a lottery $\langle p, x^\top; 1 - p, x^\perp \rangle$. These too are difficult for a user to answer accurately (Keeney and Raiffa 1976). Instead, we focus on cognitively simpler bound and comparison queries, with both local and global variants of each. These queries can be used in a variety of elicitation schemes as we discuss in the following sections.

In interactive elicitation settings, the queries we pose to the user affect not only the difficulty or accuracy of user responses, but also the modeling and computational efficacy of elicitation schemes. Each query response imposes a constraint on the set of possible utility functions the user could have in mind. The shape and structure of that space in turn influences how hard it is to compute an optimal decision or a good elicitation policy. Some queries are easy to answer, but do not provide much information, while more informative queries are often costly (for example, in terms of cognitive burden on the user).

Local Queries

In decision problems with more than a handful of attributes, local queries allow the user to concentrate only on a small subset of relevant attributes, namely, the attributes in a specific factor j and its conditioning set (or in additive models, single attributes without conditioning variables). The user does not have to consider the values of remaining attributes, since such attributes are utility independent of the attributes in factor j (given an instantiation of the conditioning set). Furthermore, in the types of queries described below, the user is always asked to assume that the attributes in the conditioning set are *fixed at their reference level*. In our example, as discussed previously, local queries for parameters of the first factor would involve instantiations of Dur and Class; the conditioning set contains a single attribute Conn, whose value would be fixed at its reference level \underline{Q} in each local query.

A *local comparison* query simply asks a user to compare two local outcomes: “Would you prefer a 4hrs,business flight to 6hrs,first flight, assuming Conn = \underline{Q} ?” If the user answers “yes,” then $v_1(4\text{hrs,business}) \geq v_1(6\text{hrs,first})$; otherwise, the first local value is less than the second. Compared to other types of queries, local comparison queries are arguably the most natural and easiest to answer. Note that the advantage of local queries becomes especially apparent when the number of domain attributes is large.

In theory, we could ask the user to directly specify the local value of some outcome x_j by using the definition of LVFs involving local lotteries. In our example, suppose the top anchor in the first factor (that is, best outcome given Conn = \underline{Q}) is 3hrs,first, and the bottom anchor is 7+,coach. We could find

the local value $v_1(6\text{hrs,business})$ by querying the user for the probability p at which she or he is indifferent between flying 6hrs,business for sure and the lottery $\langle p,(3\text{hrs,first}); 1 - p,(7+,coach) \rangle$. Of course, assessing this precise probability p is hard for a user. Therefore, less informative, but easier to answer bound queries are of more practical use. Here, the probability p is fixed in the query, and the user has only to *compare* two local outcomes, for example, “Would you prefer 6hrs,business to the lottery $\langle 0.2,(3\text{hrs,first}); 0.8,(7+,coach) \rangle$, assuming Conn = \underline{Q} ?” If the answer is “yes” ($v_1(6\text{hrs,business}) \geq 0.2$), then the lower bound becomes 0.2, if “no”, then the upper bound becomes 0.2. Instead of eliciting the precise indifference level p , we simply ask the user to bound it.

Global Queries

Global queries are needed to properly calibrate LVFs across factors by assessing the values of global parameters $u_1^\top, u_1^\perp, \dots, u_m^\top, u_m^\perp$. The anchor utility u_j^\top (u_j^\perp) is the global utility of the outcome in which the j^{th} factor is set to its best (worst) value, and all the other attributes are fixed at reference levels. In our four-attribute example, u_1^\top is the utility of the full outcome (3hrs, first, \underline{Q} , 30min), since the best values of the first factor are (3hrs, first), and the reference outcome is (4hrs, business, \underline{Q} , 30min).

Global queries do require consideration of all attribute values and as such should be minimized. However, for both additive and GAI models, we need only assess the utilities of $2m$ anchor outcomes (two for each factor); furthermore, these anchor outcomes are not arbitrary—most of their attributes are fixed at reference levels. As such, global queries need only be asked about a small number of salient outcomes.

As with local queries, direct utility or probability queries in the global case are problematic. Instead of asking users to assess precise parameter values, we can again employ easier *global comparison* and *global bound* queries, analogous to local comparison and bound queries above, in which users are asked to make a choice (assert a yes or no preference) between two global outcomes, or a global outcome and a global lottery. As above, these impose linear constraints on global parameter values.

There are many other types of queries that can be used for assessing utility function parameters.⁵ For example, a comparison query could be extended to select the most preferred outcome from the set of k alternatives; such a task would impose $k - 1$ inequalities among available choices. A total ranking query expects the user to rank all specified alternatives, yielding order information relating every pair of (possibly local) outcomes. Queries could also be less “explicit”; a decision support sys-

tem could pose implicit queries by changing the user environment (such as options available on the web page), and obtain responses by observing the user's behavior (links followed, time spent on the page, and so on).

Bayesian Elicitation

Representation of uncertainty over user preferences is a foundational issue in preference elicitation, affecting decision and query selection criteria, and influencing the algorithmic methods used to compute decisions and queries. Here, we concentrate on two distinct *explicit* uncertainty representation paradigms: Bayesian, where uncertainty over utilities is quantified probabilistically, and feasible set uncertainty, where the space of feasible utilities is defined by constraints on a user's utility function parameters.

In both cases, two questions have to be answered. First, if utility function is not fully known and further elicitation not possible, what criterion should be used for making good decisions with the available preference information? Second, if one can obtain more information about user preferences, what is the best query to ask next?

Uncertainty Representation

In a Bayesian paradigm, utility functions are modeled as random variables drawn from a prior distribution; so, even though the decision support system does not know the user's exact preferences, it has probabilistic information regarding his or her utility function parameters. These "beliefs" are updated using Bayes's rule, and the value of any decision or query is estimated by taking expectations with respect to possible user utility functions. An important precondition for representing uncertainty in terms of probability distributions over a set of utility functions is for these functions to be *extremum equivalent*, that is, share the same best and worst outcomes (Boutilier 2003).

While most recent work on decision making using distributions over utility functions has been done within the AI community (Chajewska and Koller 2000; Chajewska, Koller, and Parr 2000; Boutilier 2002; Brazuinas and Boutilier 2005), the origins of this approach can be traced back to much earlier research in game theory and decision theory: probabilistic modeling of possible payoff functions provides the foundation to the well-established field of Bayesian games (Harsanyi 1967; 1968); a related concept of adaptive utility is discussed in Cyert and de Groot (1979); de Groot (1983); and Weber (1987) proposes using expectations over utility functions as a possible criterion for decision making with incomplete preference information. The hierarchical Bayesian techniques used in the marketing field of conjoint analysis (we

discuss special forms of conjoint analysis in the next section as well) also treat utilities as random variables, drawn from a distribution that aggregates the utilities of users from a specific population.

An important issue in the Bayesian approach to modeling uncertainty over utility functions is the choice of prior probability distributions. Ideally, the probability model would be closed under updates (otherwise, it needs to be refit after each response) and flexible enough to model arbitrary prior beliefs. Mixtures of Gaussians (Chajewska, Koller, and Parr 2000), mixtures of truncated Gaussians (Boutilier 2002), mixtures of uniforms (Boutilier 2002; Wang and Boutilier 2003; Brazuinas and Boutilier 2005), and Beta distributions (Abbas 2004) are among possibilities proposed in the literature. Priors can also be learned from data: Chajewska et al. (1998), for example, cluster utility functions using a database of utilities from a medical domain.

Decisions with Partial Information

Generally speaking, good (or even optimal) decisions can be realized without complete utility information. If a prior density π over the utility function parameters is available, the best decision with respect to the prior is

$$d = \arg \max_{d \in D} \sum_x Pr_d(\mathbf{x}) E^\pi [u(\mathbf{x})]$$

Here, $Pr_d(\mathbf{x})$ is the probability that outcome \mathbf{x} is realized under decision d . Figure 2a offers an intuitive illustration of the expected utility of a decision given a density representing utility function uncertainty (with a sum or integral over possible utility functions and their probability depending on whether the set of possible utility functions is discrete or continuous).

To illustrate, consider the example in table 1 (ignoring flight attributes for simplicity). Suppose we've narrowed down a user's preferences for flights from Toronto to Seattle to one of two utility functions, u_1 and u_2 , with "willingness to pay" for the only three available flights listed in table 1.

If the system believed that the user possessed utility function u_1 with probability 0.6 and u_2 with probability 0.4, then if no additional information were available (or were deemed too costly), Flight C would be recommended (since its expected value is $0.6 \cdot \$300 + 0.4 \cdot \$600 = \$420$, compared to \$380 for both A and B).

In the case of GAI utilities, we might assume a prior distribution over the anchor utilities u_j^+ and u_j^- and the LVF parameters v_j . Because of GAI structure, the optimal (in expectation) outcome \mathbf{x}^* can usually be computed efficiently by some variant of the variable elimination algorithm (Brazuinas and Boutilier 2005; Boutilier et al. 2006).

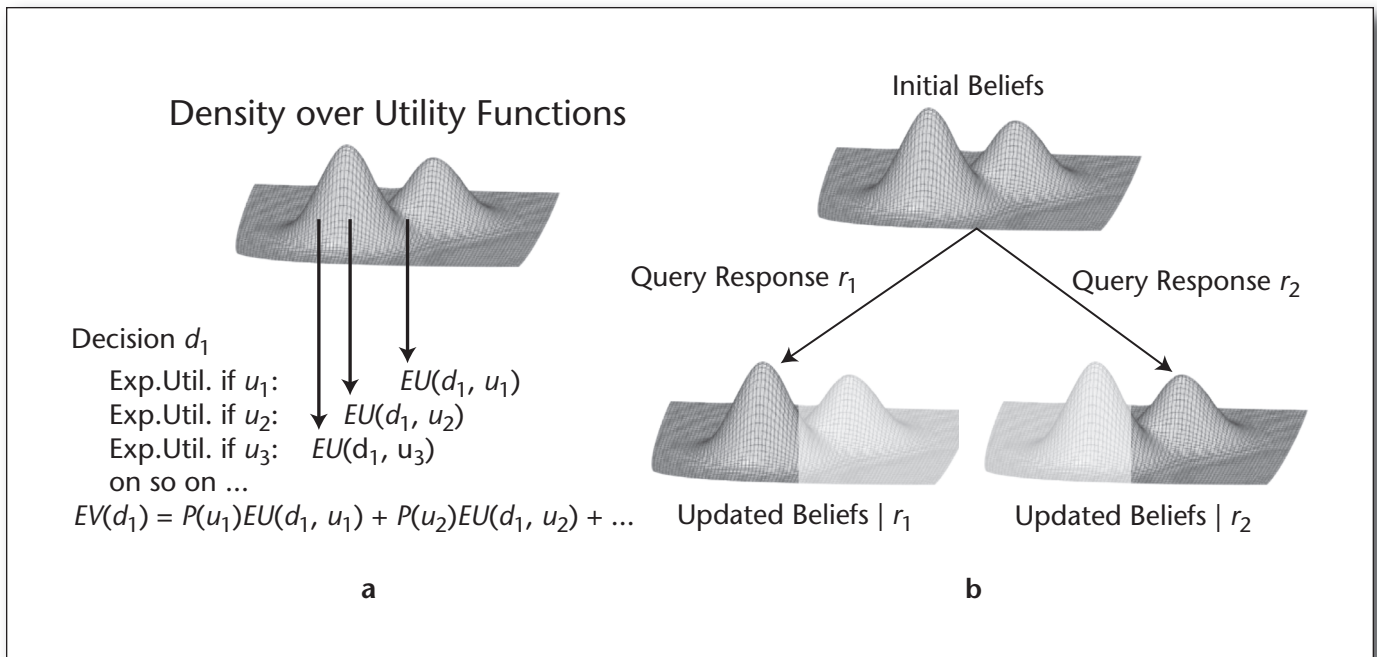


Figure 2. Utility Functions.

(a) Expected utility of a decision given a density representing utility function uncertainty. (b) Update of beliefs about utility functions given possible responses to a query.

Elicitation Strategies

If uncertainty over utilities is quantified probabilistically, the value of a query can be computed by considering the values of updated belief states (one for each possible query response) and weighting those values by the probability of corresponding responses. This leads to a simple elicitation algorithm: at each step, the query with the highest expected value of information (EVOI) is asked, and the distribution over utilities is updated based on user responses; the process stops when the expected value of a decision meets some termination criterion. Figure 2b illustrates the update of utility function beliefs given possible responses to a (in this case binary) query.

In our simple three-flight example above, suppose a query q regarding some outcome attribute might be answered positively (response r_1) with probability 0.8 and negatively (r_2) with probability 0.2 if the user has utility u_1 ; but positively with $p = 0.1$ if the user has utility u_2 . By Bayes's rule, if the user responds r_1 —which, given the system's beliefs, will happen with $p = 0.52$ —the beliefs regarding u_1 and u_2 become roughly (0.92, 0.08), and the expected value \$396 makes flight A the best decision. Conversely, response r_2 would produce beliefs (0.25, 0.75) over these utility functions, leading to flight B (with expected value \$537) being assessed the best. Thus the query q is quite valuable: no matter what the user responds, the system's decision will change (from C to A or

B). The EVOI of this query is \$44: this is the amount by which the expected value of the post-query decision, $\$464 = 0.52 \cdot \$396 + 0.48 \cdot \$537$, improves the expected value of the prequery decision, \$420.

Sequential EVOI takes into consideration all possible future questions and answers, providing an optimal trade-off between query costs (the burden of elicitation) and the potentially better decisions one can make with additional preference information. Computing a sequential elicitation policy amounts to solving a partially observable Markov decision process (POMDP) with a continuous state space (Boutilier 2002). The state space of the elicitation POMDP is the set of possible utility functions U ; actions are either queries or (terminal) decisions from the set D ; and, the observations in the POMDP are the possible user responses to queries, which are related to the user's underlying utility function (perhaps stochastically if the possibility of erroneous responses is admitted). The advantage of the sequential POMDP model lies in its ability to determine the value of a sequence of queries, where myopic methods (see below) may see no value in any single query in that sequence.

Of course, solving any POMDP is generally computationally difficult, and elicitation POMDP is no exception. Chajewska et al. (1998) instead compute an offline querying policy by greedily constructing a decision tree that assigns an unknown utility function to one of the "prototypical" clus-

ters of utility functions from a database of utilities. Other approaches use myopic online strategies for query selection. A myopic strategy selects a query that greedily maximizes its EVOI without considering the impact of future queries (Chajewska, Koller, and Parr 2000; Braziunas and Boutilier 2005). Intuitively, at each elicitation step, the query is selected as if it were the last query to be asked before a decision is made. In figure 2b, just as in our simple example above, the myopic value of a query is determined by the values of the subsequent belief states, each weighted by the probability of receiving the corresponding query response from the user (which itself depends on the prior beliefs).

For GAI models, if the priors over LVF parameters are specified using independent mixtures of uniforms, it can be shown that the best myopic bound query can be computed analytically (Braziunas and Boutilier 2005). This is due to the fact that mixtures of uniforms are closed under updates resulting from bound queries, which makes it possible to maintain an exact density over utility parameters throughout the elicitation process. An extension to comparison and other types of queries is still an open research question.

Elicitation with Feasible Utility Sets

Another way to model uncertainty over utilities is to maintain an explicit representation of a set U of *feasible* utility functions, namely, those consistent with our knowledge of the user's preferences (for example, based on responses to elicitation queries asked so far). We refer to this form of "set-based" uncertainty as *strict uncertainty*, following French (1986). The set U is updated—reduced in size—when new preference information is received during the elicitation process. Unlike the Bayesian case, we do not have probabilistic information about the relative likelihood of the different $u \in U$. If attention is restricted to the types of queries described in the Elicitation Queries section previously, the space U is conveniently characterized by a set of linear constraints on utility function parameters; if each parameter is bounded, U is simply a convex polytope in utility parameter space (see figure 3a).

As in Bayesian elicitation, two main issues have to be addressed: how to make decisions with the feasible utility set and how to select good elicitation queries when a user is willing to provide additional preference information.

Decision Criteria

In the Bayesian case, maximizing expected utility by taking expectations over possible user utility functions is a natural objective for making decisions under utility function uncertainty. Without

	Flight A	Flight B	Flight C
u_1	\$400	\$200	\$300
u_2	\$350	\$650	\$600

Table 1. An Illustrative Example for Decision and Elicitation Criteria.

u_1 and u_2 are two possible utility functions, with different valuations of the three flight outcomes. In general, the set of possible utility functions is continuous.

probabilistic information, the choice of an appropriate decision criterion is much less obvious. In this section, we discuss a sample of the most common rules for decision making under strict utility function uncertainty. Since different decision rules generally prescribe different alternatives, little consensus exists in this area. French (1986) provides an extensive critique of various decision criteria under strict uncertainty.

To illustrate the various concepts presented in this section, we use the simple three-flight example from the previous section, detailed in table 1. If a decision support system were to recommend a flight without knowing the user's utility function (either u_1 or u_2), what criterion should it use?

Undominated Outcomes. Imprecisely specified multiattribute utility theory (ISMAUT) (White, Dozono, and Scherer 1983; White, Sage, and Dozono 1984; Anandalingam and White 1993) is one well-studied model for decision making with partial preference information. Similar frameworks have been explored in Fishburn (1964); Sarin (1977); Kirkwood and Sarin (1985); Hazen (1986); Weber (1987); and Blythe (2002). ISMAUT generally applies to situations where the set of feasible outcomes X is of computationally manageable size. Prior information on local value functions, scaling constants, and comparisons between pairs of outcomes is used to narrow down the set of outcomes to those that are *undominated* by any other alternative. We say one outcome *dominates* another if its utility is greater for any $u \in U$ (in our example, none of the flights is pairwise dominated by another flight).⁶ This smaller set of nondominated alternatives is then presented to the decision maker; the optimal outcome is guaranteed to be in it. ISMAUT does not propose a decision criterion; however, its main ideas have strongly influenced further developments in preference elicitation.

Maximin Return. Without distributional information about the set of possible utility functions U , it might seem reasonable to select an outcome whose worst-case return is highest:

$$\mathbf{x}^* = \arg_{\mathbf{x} \in X} \max \min_{u \in U} u(\mathbf{x})$$

In this section, for simplicity, we assume that each decision $d \in D$ leads to a distinct outcome $\mathbf{x} \in X$, so the decision and outcome spaces are equivalent. Maximin decision is robust because it provides a minimum utility guarantee. In our example, the maximin optimal flight is Flight A, with a guaranteed minimum utility of \$350. Maximin was introduced by Wald (1950), and proposed in 2004 by Salo and Hamalainen⁷ as a possible criterion for dealing with uncertain utilities.

Uniform Distribution. This criterion dates back to Laplace and Bernoulli, who maintained that complete lack of knowledge about the likelihood of world states should be treated as if all states have equal probability. Following this principle, an optimal decision maximizes the mean value (over possible $u \in U$) of possible outcomes. Flight C would be optimal in our example with an average utility of \$450. The uniform distribution criterion is also known as the central weights decision rule in reasoning with utility function uncertainty (Saló and Hamalainen 2001), and is implicitly employed in Iyengar, Lee, and Campbell (2001) and Ghosh and Kalagnanam (2003), as well as in polyhedral conjoint analysis (Toubia, Hauser, and Simester 2004). The feasible utility set is represented by a convex polytope in the space of multiattribute utility parameters. The “center” of such a polytope then serves as a representative utility function for the whole feasible set.

Minimax Regret. The minimax regret criterion was first described by Savage (1951) in the context of uncertainty over world states, and has been advocated more recently for robust decision making with incompletely specified utility functions (Boutillier, Bacchus, and Brafman 2001; Salo and Hamalainen 2001; Boutillier et al. 2006). It requires that our system recommend an outcome \mathbf{x}^* that minimizes *maximum regret* with respect to all possible realizations of the user’s utility function. This guarantees worst-case bounds on the quality of the decision made under strict uncertainty and is therefore reasonable in many real-world scenarios (Wang and Boutillier 2003; Boutillier, Sandholm, and Shields 2004; Boutillier et al. 2006; Braziunas and Boutillier 2007). The *pairwise regret* of choosing \mathbf{x} instead of \mathbf{x}' given the feasible utility set U is $R(\mathbf{x}, \mathbf{x}', U) = \max_{u \in U} u(\mathbf{x}') - u(\mathbf{x})$. The maximum regret of choosing outcome \mathbf{x} is $MR(\mathbf{x}, U) = \max_{\mathbf{x}' \in X} R(\mathbf{x}, \mathbf{x}', U)$. Finally, the outcome that minimizes max regret is the minimax optimal decision:

$$\mathbf{x}^* = \arg \max \min_{\mathbf{x} \in X} MR(\mathbf{x}, U)$$

In our example, the max regret of choosing Flight A is \$300: if we recommend A, the user may in fact have utility u_2 , in which case A is \$300 worse than B, the optimal flight given u_2 . Similarly, Flight B

has a max regret of \$200, and Flight C \$100. Thus C is the *minimax optimal choice* as it has the smallest max regret. Flight A and utility function u_1 serve as the “adversarial witnesses” that “prove” how much we could regret recommending C (since under u_1 , A is \$100 better than C).

In a GAI model, utility structure can be exploited in order to express pairwise regret as a sum of “local” factor regrets. Employing the type of queries described above (and thus ensuring the locality of constraints on utility parameters) allows us to efficiently compute pairwise regret using linear programming. It can then be shown that the maximum regret and minimax regret optimizations can be effectively solved using suitable mixed-integer formulations (Boutillier et al. 2006; Braziunas and Boutillier 2007).

Elicitation Strategies

When further utility information can be elicited, a natural question is what query to ask next. With feasible utility sets, we can distinguish two general elicitation paradigms: uncertainty reduction and decision quality improvement.

Uncertainty Reduction. Many methods exist in diverse research areas, such as conjoint analysis and ISMAUT, whose central idea is to choose queries according to criteria based on the size and shape of the feasible utility set. In most cases, a normalized additive utility function is assumed (see equation 2) in which either scaling constants λ_i or local value parameters v_i (but not both)⁸ are not fully known. Figure 3a provides a graphical representation of the feasible utility polytope. Information about utility parameters can be obtained by using bound queries on parameter values, or, more commonly, by asking the user to compare pairs of full outcomes. In classical ISMAUT, the elicitation strategy problem is not addressed; instead, the user is asked to order randomly selected pairs of outcomes until the set of nondominated alternatives is reduced to acceptable size.

More recent methods consider several heuristic query strategies based on the size and shape of the convex polytope that represents possible utility parameter values. The queries are limited to full outcome comparison queries. The Q-Eval algorithm (Iyengar, Lee, and Campbell 2001) follows a querying strategy whose goal is to reduce the volume of the utility polytope U as quickly as possible. Since a response to a query is not known beforehand, Q-Eval (heuristically) chooses the query that comes closest to bisecting the polytope into two equal parts. A similar approach is proposed by Ghosh and Kalagnanam (2003); the differences lie in approximations used to implement the querying strategy. Both methods are myopic, since they do not consider the impact of a sequence of

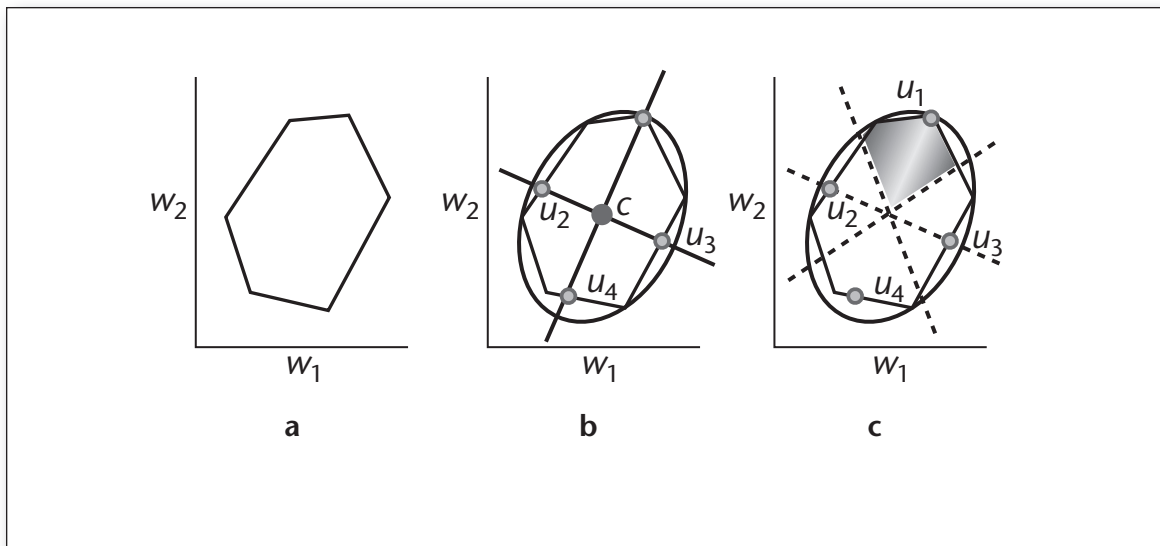


Figure 3. Elicitation with the Feasible Utility Polytope.

(a) A two-dimensional utility polytope induced by responses to elicitation queries; (b) the analytic center c and bounding ellipsoid used in polyhedral conjoint analysis—the intersection of its axes with the polytope boundaries define utility prototypes; (c) the indifference hyperplanes between product profile x_1 (optimal for u_1) and the other three product profiles, and the feasible region (shaded) that remains if x_1 is preferred to the other three.

queries. Holloway and White (2003) present a POMDP model for sequentially optimal elicitation in an ISMAUT-like setting, where an optimal policy can minimize the number of queries to reach a provably optimal decision.

Recent developments in conjoint analysis also fall under uncertainty reduction category. Conjoint analysis is a set of techniques for measuring consumer tradeoffs among multiattribute products and services. Despite differences in terminology and methodology, conjoint analysis and multiattribute decision analysis (in particular, ISMAUT) deal with similar issues in preference elicitation and modeling. Adaptive query design for individual users in the manner of ISMAUT was first considered in Toubia et al. (2003) and Toubia, Hauser, and Simester (2004). Just as in ISMAUT, this approach, termed the *polyhedral method*, works by iteratively constraining the polytope of feasible utility parameter values as a result of responses to comparison queries. The goal is to reduce the volume of the polytope as fast as possible, while also considering the polytope's shape. The polytope is approximated by a bounding ellipsoid centered at the analytic center of the polytope (a point with maximal distance from each facet), as illustrated in figure 3b. Queries are designed to partition the bounding ellipsoid into approximately equal parts; in addition, shape heuristics are used to favor cuts that are perpendicular to the longest axes of the ellipsoid. The rationale is not only to reduce over-

all uncertainty, but also to balance uncertainty in each dimension. Consider the example in figure 3b, where the two longest axes are shown (assume that axes in other dimensions are shorter). Prototype utility functions are designated where these axes intersect the boundaries of the polytope. Then a choice query is posed using specially designed (generally hypothetical) product configurations that are optimal for these prototypes and whose indifference profiles roughly equally bisect the polytope between the prototypes. Figure 3c shows the indifference profiles for product x_1 (which is optimal at u_1) versus the other three profiles (x_2 , x_3 , and x_4). The user is presented with the choice query “Which of x_1 , x_2 , x_3 or x_4 do you prefer?” If he or she answers x_1 , then the shaded region (where the inequalities $u(x_1) \geq u(x_i)$, $i \neq 1$ hold) is the new feasible region.

The main limitation of uncertainty reduction methods is the absence of decision quality considerations in elicitation. For instance, one can be quite confident in the quality of a decision (for example, even possibly confirming its optimality) despite the fact that considerable uncertainty exists in the user utility function. Next, we discuss one way of guiding utility elicitation using decision quality.

Decision Quality Improvement. The minimax regret criterion can be used both for making robust decisions under strict uncertainty and for guiding the elicitation process itself. In contrast to Bayesian

elicitation, the quality (difference from optimal) of the minimax-optimal decision can be bounded; and these bounds can be tightened with further elicitation. We can query the user until minimax regret reaches an acceptable level, elicitation costs become too high, or some other termination criterion is met. Each response to a preference query results in a new decision situation inducing a new level of minimax regret (note that regret cannot increase with more preference information).

One way to choose queries is described in Brazunas and Boutilier (2007). The basis of this heuristic myopic strategy is a generalization of the current solution (CS) elicitation strategy (Boutilier et al. 2006), which has been shown empirically to be very effective in quickly reducing minimax regret in several domains (Boutilier et al. 2006; Boutilier, Sandholm, and Shields 2004). The CS strategy considers only parameters involved in defining minimax regret and asks a query about the parameter that offers the largest potential reduction in regret. In our example, the current solution consists of the minimax-optimal outcome Flight *C* and the witness outcome Flight *A*. Therefore, the CS strategy would ask only queries that involve the attribute instantiations occurring in either Flight *A* or Flight *C*, but not *B* (or any other flight). Instead of reducing uncertainty uniformly across the feasible utility space, the CS method concentrates elicitation effort on relevant regions of utility space. In Brazunas and Boutilier (2007), the CS method is developed for GAI models, using both the local and global queries described above. Heuristic techniques for making tradeoffs involving estimated regret reduction and the cognitive difficulty of queries (for example, where global queries are considered more difficult than local queries) are also investigated.

Summary and Future Directions

We have provided an overview of factored utility representations and several techniques for preference elicitation designed specifically for multiattribute models. Our emphasis has been on models that maintain an explicit representation of utility function uncertainty and use this model both to make decisions and to determine which queries to pose to a user. We divide these techniques into two broad classes. Bayesian models require a prior probability distribution over user utility functions. Optimal decisions and elicitation strategies are prescribed by the natural application of well-accepted notions like maximum expected utility and expected value of information. Feasible set methods avoid problems of prior distributions and complicated computational approximations by assuming no distributional knowledge about user utilities. In most cases, the feasible set of utility

functions is represented by a convex polytope in utility parameter space. Among the possible criteria for optimization and query selection, minimax regret stands out because of its focus on decision quality improvement rather than simple utility uncertainty reduction.

Preference elicitation research touches many disciplines, from AI, operations research, and economics, to human-computer interaction, behavioral psychology and marketing. While progress has been significant in recent years, a number of fascinating research challenges must be tackled to extend the reach of preference elicitation, whether scaling to larger domains, or improving the nature of user interaction. We list a few important research directions here.

Obviously, computational improvements in both Bayesian and feasible set approaches will continually be required to scale to domains with large numbers of attributes and outcomes. Improving user interaction can be dealt with on a number of fronts. First, we can try to minimize the number of user queries by computing optimal sequential query policies. Almost all current approaches are myopic, focusing on the single next best query to ask (one exception being (Boutilier 2002)). Considering new modes of interaction is also vital. For instance, integrating elicitation methods that focus on simple-to-answer queries could be combined with more user-directed exploration methods such as example critique (Viappiani, Faltings, and Pu 2006), especially if the “open-ended” responses of users can be given a precise semantic interpretation. Other hybrid preference assessment techniques could combine elicitation specific to a particular user with collaborative filtering methods (in which the preferences of a user are partially induced by extrapolating the preferences of other users). Methods for active collaborative filtering (Boutilier, Zemel, and Marlin 2003; Jin and Si 2004) can be viewed as taking a step in this direction, but currently use only impoverished utility models. Similarly, integrating active elicitation with methods for inferring preferences from passive observation is especially important for the long-term, repeated interaction with individual users. The careful experimental study of query costs for different modes of interactions will certainly influence the future design of elicitation schemes.

Notes

1. Passive or “active” observation of user behavior, social filtering techniques like collaborative filtering (Konstan et al. 1997), or their combination with direct preference elicitation (Pennock and Horvitz 2000; Boutilier, Zemel, and Marlin 2003) can also be used to assess preferences.
2. These are the best and worst outcomes ignoring price, reflecting our quasilinearity assumption. Note that preferences need not be monotonic in quantitative attrib-

utes; for example, a 7+ hour flight may be preferred to a 6-hour flight by a certain user (possibly dependent on the number of connections) because it provides a better opportunity to sleep.

3. Formally, attributes should be *additively independent* of each other; that is, a user should be indifferent between any two lotteries whose marginal probability distributions for each attribute domain are identical. For example, if a user's preferences over two Boolean attributes A and B were additively independent, then the user would be indifferent between receiving: (1) outcomes ab and \underline{ab} each with $p = 0.5$; and (2) ab , \underline{ab} , $\underline{a}\underline{b}$, and $\underline{a}\underline{b}$ each with $p = 0.25$. In each lottery, the marginal distributions over A ($p(a) = p(\underline{a}) = 0.5$) and B are the same.

4. Fishburn (1967) used the term *interdependent value additivity*; Bacchus and Grove (1995) dubbed the same concept *GAI*, which is more commonly used in the AI literature currently.

5. An extensive survey of various query types is provided by Farquhar (1984).

6. Note, however, that C is setwise dominated, in the sense that, for either utility function u_1 or u_2 , there exists a better flight than C .

7. See Preference Programming by A. Salo and R. P. Hamalainen. Manuscript available at www.sal.hut.fi/Publications/pdf-files/msal03b.pdf.

8. If both parameters were unknown, then responses to full outcome comparison queries would impose quadratic, rather than linear constraints on parameters, thus complicating the uncertainty representation and the computation of optimal decisions.

References

- Abbas, A. 2004. Entropy Methods for Adaptive Utility Elicitation. *IEEE Transactions on Systems, Science and Cybernetics* 34(2): 169–178.
- Anandalingam, G., and White, C. C. 1993. A Penalty Function Approach to Alternative Pairwise Comparisons in ISMAUT. *IEEE Transactions on Systems, Man, and Cybernetics* 23(1): 330–333.
- Bacchus, F., and Grove, A. 1995. Graphical Models for Preference and Utility. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, 3–10. San Francisco: Morgan Kaufmann Publishers.
- Blythe, J. 2002. Visual Exploration and Incremental Utility Elicitation. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 526–532. Menlo Park, CA: AAAI Press.
- Boutilier, C. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 239–246. Menlo Park, CA: AAAI Press.
- Boutilier, C. 2003. On the Foundations of Expected Utility. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 285–290. Menlo Park, CA: AAAI Press.
- Boutilier, C.; Bacchus, F.; and Brafman, R. I. 2001. UCP-Networks: A Directed Graphical Representation of Conditional Utilities. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, 56–64. San Francisco: Morgan Kaufmann Publishers.
- Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2006. Constraint-Based Optimization and Utility Elicitation Using the Minimax Decision Criterion. *Artificial Intelligence* 170(8–9): 686–713.
- Boutilier, C.; Sandholm, T.; and Shields, R. 2004. Eliciting Bid Taker Non-Price Preferences in (Combinatorial) Auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, 204–211. Menlo Park, CA: AAAI Press.
- Boutilier, C.; Zemel, R. S.; and Marlin, B. 2003. Active Collaborative Filtering. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI-03)*, 98–106. San Francisco: Morgan Kaufmann Publishers.
- Braziunas, D., and Boutilier, C. 2005. Local Utility Elicitation in GAI Models. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, 42–49. San Francisco: Morgan Kaufmann Publishers.
- Braziunas, D., and Boutilier, C. 2007. Minimax Regret-Based Elicitation of Generalized Additive Utilities. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI-07)*, 25–32. San Francisco: Morgan Kaufmann Publishers.
- Chajewska, U., and Koller, D. 2000. Utilities as Random Variables: Density Estimation and Structure Discovery. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, 63–71. San Francisco: Morgan Kaufmann Publishers.
- Chajewska, U.; Getoor, L.; Norman, J.; and Shahar, Y. 1998. Utility Elicitation as a Classification Problem. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 79–88. San Francisco: Morgan Kaufmann Publishers.
- Chajewska, U.; Koller, D.; and Parr, R. 2000. Making Rational Decisions Using Adaptive Utility Elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00)*, 363–369. Menlo Park, CA: AAAI Press.
- Cyert, R. M., and de Groot, M. H. 1979. Adaptive Utility. In *Expected Utility Hypothesis and the Allais Paradox*, ed. M. Allais and O. Hagen, 223–241. Dordrecht: D. Reidel.
- de Groot, M. H. 1983. Decision Making with an Uncertain Utility Function. In *Foundations of Utility and Risk Theory with Applications*, ed. B. P. Stigum and F. Wenstop, 371–384. Dordrecht: D. Reidel.
- Engel, Y., and Wellman, M. P. 2007. Generalized Value Decomposition and Structured Multiattribute Auctions. In *Proceedings of the Eighth ACM Conference on Electronic Commerce (EC-2007)*, 227–236. New York: Association for Computing Machinery.
- Farquhar, P. H. 1984. Utility Assessment Methods. *Management Science* 30(11): 1283–1300.
- Fishburn, P. C. 1964. *Decision and Value Theory*. New York: Wiley.
- Fishburn, P. C. 1967. Interdependence and Additivity in Multivariate, Unidimensional Expected Utility Theory. *International Economic Review* 8: 335–342.
- French, S. 1986. *Decision Theory*. New York: Halsted Press.
- Ghosh, S., and Kalaganam, J. 2003. Polyhedral Sampling for Multiattribute Preference Elicitation. In *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC-2003)*, 256–257. New York: Association for Computing Machinery.
- Gonzales, C., and Perny, P. 2004. GAI Networks for Util-

Join Us for IJCAI-09 and IAAI-09 in Pasadena, California!

Please mark your calendars now for the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09) and the Twenty-First Innovative Applications of Artificial Intelligence Conference (IAAI-09)! The conferences will be held July 11–17, at the Pasadena Convention Center in Pasadena, California.

ity Elicitation. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR2004)*, 224–234. Menlo Park, CA: AAAI Press.

Green, P. E., and Srinivasan, V. 1978. Conjoint Analysis in Consumer Research: Issues and Outlook. *Journal of Consumer Research* 5(2): 103–123.

Harsanyi, J. C. 1967. Games with Incomplete Information Played by Bayesian Players. Part I: The Basic Model. *Management Science* 14: 159–182.

Harsanyi, J. C. 1968. Games with Incomplete Information Played by Bayesian Players. Part II: Bayesian Equilibrium Points. *Management Science* 14: 320–334.

Hazen, G. B. 1986. Partial Information, Dominance, and Potential Optimality in Multiattribute Utility Theory. *Operations Research* 34(2): 296–310.

Holloway, H. A., and White, III, C. C. 2003. Question Selection for Multiattribute Decision-Aiding. *European Journal of Operational Research* 148(3): 525–533.

Iyengar, V. S.; Lee, J.; and Campbell, M. 2001. Q-Eval: Evaluating Multiple Attribute Items Using Queries. In *Proceedings of the Third ACM Conference on Electronic Commerce*, 144–153. New York: Association for Computing Machinery.

Jin, R., and Si, L. 2004. A Bayesian Approach toward Active Learning for Collaborative Filtering. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI-04)*, 278–285. San Francisco: Morgan Kaufmann Publishers.

Keeney, R. L., and Raiffa, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. New York: Wiley.

Kirkwood, C. W., and Sarin, R. K. 1985. Ranking with Partial Information: A Method and an Application. *Operations Research* 33(1): 38–48.

Konstan, J. A.; Miller, B. N.; Maltz, D.; Herlocker, J. L.; Gordon, L. R.; and Riedl, J. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM* 40(3): 77–87.

Pennock, D. M., and Horvitz, E. 2000. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, 473–480. San Francisco: Morgan Kaufmann Publishers.

Salo, A., and Hamalainen, R. P. 2001. Preference Ratios in Multiattribute Evaluation (Prime)-Elicitation and Deci-

sion Procedures under Incomplete Information. *IEEE Transactions on Systems, Man and Cybernetics* 31(6): 533–545.

Sarin, R. K. 1977. Screening of Multiattribute Alternatives. *Omega* 5(4): 481–489.

Savage, L. J. 1951. The Theory of Statistical Decision. *Journal of the American Statistical Association* 46(253): 55–67.

Toubia, O.; Simester, D. I.; Hauser, J. R.; and Dahan, E. 2003. Fast Polyhedral Adaptive Conjoint Estimation. *Marketing Science* 22(3): 273–303.

Toubia, O.; Hauser, J. R.; and Simester, D. I. 2004. Polyhedral Methods for Adaptive Choice-Based Conjoint Analysis. *Journal of Marketing Research* 41(1): 116–131.

Viappiani, P.; Faltings, B.; and Pu, P. 2006. Preference-Based Search Using Example-Critiquing with Suggestions. *Journal of Artificial Intelligence Research (JAIR)* 27: 465–503.

Wald, A. 1950. *Statistical Decision Functions*. New York: Wiley.

Wang, T., and Boutilier, C. 2003. Incremental Utility Elicitation with the Minimax Regret Decision Criterion. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 309–316.

Weber, M. 1987. Decision Making with Incomplete Information. *European Journal of Operational Research* 28: 44–57.

White, C. C.; Dozono, S.; and Scherer, W. T. 1983. An Interactive Procedure for Aiding Multiattribute Alternative Selection. *Omega* 11(2): 212–214.

White, III, C. C.; Sage, A. P.; and Dozono, S. 1984. A Model of Multiattribute Decision-Making and Trade-Off Weight Determination under Uncertainty. *IEEE Transactions on Systems, Man and Cybernetics* 14(2): 223–229.

Darius Braziiunas is a Ph.D. student at the University of Toronto. His research interests include various decision-theoretic aspects of preference elicitation, in particular multiattribute utility models, sequentially optimal elicitation strategies, and decision making with partial utility information. He received a B.A. degree from Middlebury College, VT, in 2000 and an M.Sc. from the University of Toronto in 2003.

Craig Boutilier received his Ph.D. in computer science (1992) from the University of Toronto, Canada. He is a professor and chair of the Department of Computer Science at the University of Toronto. He was previously an associate professor at the University of British Columbia, a consulting professor at Stanford University, and a visiting professor at Brown University. He has served on the Technical Advisory Board of CombineNet, Inc., since 2001. Boutilier's research interests span a wide range of topics, with a focus on decision making under uncertainty, including preference elicitation, mechanism design, game theory, Markov decision processes, and reinforcement learning. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI) and the recipient of the Isaac Walton Killam Research Fellowship, an IBM Faculty award, and the Killam Teaching award. He has also served in a variety of conference organization and editorial positions, and is program chair of the upcoming 21st International Joint Conference on Artificial Intelligence (IJCAI-09).