RZ 3450 (# 93513) 08/26/02
Computer Science 18 pages

Research Report

Numeric Estimation of Partial Utility Models for Automated Preference Interviewing

Lauren Aaronson, Markus Stolze and Michael Ströbel

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{laa, mrs, mis}@zurich.ibm.com

Numeric Estimation of Partial Utility Models for Automated Preference Interviewing

Lauren Aaronson, Markus Stolze, Michael Ströbel
IBM Zurich Research Laboratory
Säumerstrasse 4
8803 Rüschlikon
Switzerland
email: {laa, mrs, mis}@zurich.ibm.com

Abstract: Many online systems today attempt to personalize their interaction with users. For example, electronic shops may want to provide decision support to users by recommending products, and portals may try to customize their content in a unique fashion for each user. Personalization requires a model of a user's preferences, and in many cases, preferences are represented in the form of a utility function. Because it is difficult to elicit the information necessary to specify completely a user's utility function, many online systems that employ user modeling must reason with partial utility models; for instance, an online recommender system might have to use incomplete information about a user's preferences when choosing a product to suggest. Although a ranking of alternatives based on partial utility models cannot be certain, such a ranking can sometimes be beneficial. In that case, we require an easily interpretable numeric estimate of the utilities of alternatives. Existing systems that use numeric estimates of partial utility models have the problem that the estimates do not consistently improve in accuracy. Some sequences of questions that might be asked in order to elicit a user's utility function can result in a final numeric estimate of an alternative that is actually more inaccurate than the starting estimate. We propose a simple means of numeric utility estimation that tracks the lower bound of a user's utilities. Our defensive estimation technique has the advantages that it is guaranteed to improve in accuracy over time and that its estimates can be clearly understood by the user to represent the minimum utility of an alternative. These properties allow us to use an automated interview construction algorithm with full confidence that at termination the estimate of the user's utility will be more accurate than at the beginning. The technique introduced in this paper also allows us to display a reasonable ranking of choices to the user at any point in an interview. In addition, we present a particular interview construction algorithm that, because of its use of our defensive utility estimation technique, is able to select the question that results in the greatest expected reduction in uncertainty of a partial utility model.

1 The Need for Reasonable Numeric Estimates of Partial Utility Models

One way to assist users in choosing among alternatives, such as in the selection of products in an electronic shop or in the customization of online content, is to apply the methods of decision theory, which holds that the best choice for a user to make in a given situation is the one that maximizes her expected utility. The expected utility of a decision is computed by summing the utility for the user over its possible outcomes, weighted by their probabilities. An accurate model of a user's preferences, in the form of a utility function that maps outcomes to utilities, is thus a critical piece of a decision-theoretic system, and so it is a critical piece of many systems that aim to provide user-specific recommendations or content.

Traditional decision theory assumes that a user's utility function is completely specified before the decision-making process begins. However, the task of eliciting a user's full utility function is difficult and time-consuming (Ha and Haddawy 1997; Chajewska et al. 2000). The user's utility function is usually elicited by conducting an interview about her preferences, but for many automated systems in which decision theory is a useful tool, eliciting a complete model of a user's utility function is infeasible. For instance, time constraints or user frustration may limit the number of questions an online recommender system can ask a user during a utility elicitation interview. In these cases, the system must present the user with choices before the utility model has been completely assessed. The system must then present these choices on the basis of a partial utility model, in which some outcomes are not mapped to utilities. In order to do this well, the system must be able to reason with and about partial utility models in a sensible way. Since the number of questions the system can ask is limited, a related goal is for the system to select those utility elicitation questions that tend most to improve its model of the user's utility function.

Researchers in artificial intelligence have recently looked into the intricacies of reasoning with incomplete utility models, as well as into techniques for choosing interview questions. Much of the work that has been done on partial utility models concerns the qualitative representation of preferences (Bacchus et. al. 1995; Boutilier et al. 1997; Tan and Pearl 1994) or reasoning in a non-numeric way about alternatives (Ha and Haddawy 1997). Most of these approaches, such as that of Ha and Haddawy (1997), use partial information about a user's utility function to eliminate alternatives that can be proven to be suboptimal. The elimination process results in a set of unranked alternatives that, as long as the utility model is incomplete, cannot be known to be satisfactory, but have not yet been shown to be unsatisfactory.

The aim of approaches that eliminate dominated alternatives is to narrow down the set of alternatives sufficiently so that a user can view the remaining options herself and choose the one that best suits her needs. This is quite a valuable approach, but in some cases, the remaining set of alternatives may still be quite large. In cases such as this, the user may benefit from a ranking of the remaining alternatives. If a partial utility model is used to rank nondominated alternatives in some way, the ranking cannot be certain, no matter what method of computing the ranking is used—new information about the form of the utility function may alter the ranking. As such, any ranking method will be merely a heuristic for ordering the alternatives. But there may be some ranking methods that are easily interpretable, so that the information they provide to the user is valuable, despite the possibility of change. Such a ranking method could be used to present alternatives to a user either during the course of an interview, after a user has answered all the questions she wishes to answer, or after an interview has ended. Ranking methods such as this would be particularly useful in online recommender and navigation aid systems. In these domains,

users are not likely to wish to spend a great deal of time answering questions or sorting through numerous options, and they may appreciate being presented with alternatives as they answer or if they decide to leave an interview early.

In order to rank alternatives, we need a measure with which to order them, and the most natural measure to use is a numeric estimate of their utilities. Some systems use ranges or probabilities of numeric utilities to model uncertainty about a utility function, and most of these systems produce numeric estimates of utilities by using the average according to the ranges or probabilities (Jameson et al. 1995; Chajewska et al. 2000; Chin and Porage 2001; Porage 1999). However, such estimates have an undesirable property when used by algorithms that automatically choose elicitation questions or in interviews that a user may terminate at any point. An estimate produced using averages is not guaranteed to be moving closer to the final utility that would result from a completely assessed utility model. Depending on which and how many questions are chosen, the estimate of the user's utility function—and thus the selection of alternatives presented to her—may actually be worse at the end of the interview than at the beginning.

To counteract the problem of potentially worsening estimates, some automated interview construction methods use another mechanism to determine the distance of the current numeric estimate from that of the complete utility function, such as an estimation of the regret the user would experience by stopping with the current recommendation (Chajewska et al. 2000). However, there are still cases in which a consistently improving estimate, rather than a way to assess the accuracy of the current estimate, is desirable. In particular, a consistently improving estimate is crucial if recommendations are to be presented to a user before the end of an interview is reached. (This is a valuable tactic in online recommender systems, since users may often decide to exit a utility elicitation interview early.) Another benefit of a consistently improving estimate is that it could simplify interview construction for implementors of recommender systems if they, or the question-choosing algorithms they design, were allowed to ask any sequence of the available questions in their database. For instance, implementors of electronic shops may often have their own preferences about which questions to ask a user.

A further advantage of a consistently improving estimate is that it may be more readily interpretable by users. In the approaches that use averages, it is not clear at each point in the interview what the numeric estimates indicate about a user's actual utilities, whether or not the associated interview construction technique ends questioning only when the estimate has improved. A user may have difficulty viewing the averaged numeric estimates as the system's "best guesses" about her actual utilities, since, counterintuitively, the guesses can decrease in accuracy as more information about the user becomes known . In addition, the possible decrease in accuracy could make it difficult for an automated system to assign meaning to the numeric estimates in order to use them consistently in calculations. Thus, we believe that there are purposes for which another kind of numeric estimate is necessary.

2 Our Proposal

Our solution to the problem of finding an interpretable numeric estimate of partial utility models is a simple one, but it nonetheless accomplishes the task of ensuring a strict progression of accuracy. We propose using a defensive technique for numeric estimation of partial utility models, one which tracks the lower bound of the utilities of the alternatives. In addition to improving with time, this technique has the advantage that the numeric utility can be easily understood to represent the minimum satisfaction that a user will gain from a

certain alternative, and so the utility estimate serves as an understandable basis for ranking alternatives.

The defensive estimation technique we present here can be used by any automated interview construction algorithm that requires estimates of partial utility models, with full confidence that at the close of an interview the utility estimate will be more accurate than at the beginning. As a second contribution of this paper, we present one particular interview construction algorithm, based on our earlier paper (Stolze and Ströbel 2001), that is able to use the properties of our defensive estimation technique in order to select the questions that tend most to reduce uncertainty about the partial utility model. Defensive utility estimation thus helps us to achieve two important tasks for systems that rely on decision-theoretic methods: on its own, it allows us to reason with partial models in a consistent way, and, when applied in our algorithm for question selection, it allows us to select questions that tend most to improve our knowledge about them.

In the next section, we briefly describe utility theory and explain terms that will appear in the rest of the paper. Then we present a technique for defensively estimating utility functions in general, and, in particular, for the additive utility functions that are commonly assumed by user modeling systems (Linden at al. 1994; Ha and Haddawy 1997). In Section 5, we apply this technique in an algorithm that constructs utility elicitation interviews. In Section 6, we discuss related work on reasoning with and refining partial utility models, as well as benefits, problems, and potential extensions related to our system.

3 Multiattribute Utility Theory

In this section, we give a brief overview of some key terms and ideas from decision theory and multiattribute utility theory. We concentrate primarily on definitions and techniques relevant to additive utility functions, since our discussion focuses on such functions. The explanation in Sections 3.1 and 3.2 is based on information in Keeney and Raiffa (1976) and in Clemen's textbook (1996). For more details on utility theory, particularly on independence conditions, conditions of uncertainty, and nonadditive utility functions, we refer the reader to the previously mentioned books or to another textbook on decision theory.

3.1 Definitions

The goal of multiattribute utility theory is to represent the satisfaction that a person will get from a variety of different *outcomes*, or states. A *utility function* maps outcomes to *utilities*, which are values on a scale from 0 to 1. An outcome is generally assumed to consist of different *values* for certain variables, or *attributes*. If certain independence conditions are met, the utility function can be decomposed into a combination of *individual* or *subutility* functions. A *subutility function* is a function that maps the values of an attribute, or a combination of attributes, into utilities, also between 0 and 1. One kind of decomposable utility function is the *multilinear* function, which is capable of representing certain kinds of interactions among attributes. A multilinear function is computed by summing, for each possible set of attributes, the product of the subutility functions of the attributes in the set. (For simplicity, we will follow Ha and Haddawy (1997) in referring to sets of attributes as attributes themselves.) A special case of multilinear functions is the *additive* function, for which the utility is a weighted sum of all subutility functions.

In the case of additive utility functions, each attribute has a *weight* by which we multiply the result of the subutility function. This weight represents how important that

attribute is to the user, and the weights for all attributes must add up to 1. The result will be a total utility between 0 and 1. Formally, the utility of a given outcome is calculated as follows, where k_i is the weight of attribute i, x_i is the value of attribute i for the given alternative, and U_i is the subutility function for attribute i (Clemen 1996):

$$U(x_{1,\ldots,}x_n) = \sum_{i=1}^{n} k_i U_i(x_i)$$

Technically, there is a difference between decisions with certain outcomes and cases in which the outcomes of decisions are uncertain. The term *utility function* generally refers to the mapping of outcomes to utilities in the latter case, and a *value function* refers to a similar mapping in the case of certain outcomes. However, in this paper we will refer to both as *utility functions*. The expected utility of a decision, according to traditional decision theory, is the sum of the utilities of its possible outcomes, weighted by their probabilities. In the case of certain outcomes, there is no probability involved; the utility of the relevant outcome is achieved with certainty.

The techniques presented in the rest of this paper apply to utility functions under conditions of both certainty and uncertainty, although the likely domain of application will be in product recommenders, for which outcomes are generally certain.[1] In our discussion, we will call outcomes *alternatives*, since we will use throughout the paper an example of an online recommender system for computers, and it is more natural to think of presented products as alternatives than as states of the world.

To illustrate how multiattribute utility theory works, suppose that we are trying to recommend a computer to a user, and suppose that how much she will like a computer depends on four things: the price, the hard disk size, the amount of memory, and the speed of the computer. These four things are the attributes of a computer, or, to keep in line with the terminology of utility theory, the outcome represented by this particular computer. Suppose that in the world in which the user lives—or in a particular computer store—each attribute has four possible values. For instance, the price can be $1000, $1500, $2000, or $2500, the hard disk size can be 10 gigabytes, 15 gigabytes, 20 gigabytes, or 30 gigabytes, and, similarly, the memory and the speed each have four possible values. Then there are 4^4 different combinations of values, and so there are 256 possible alternatives (not all of which may be actually available or possible). A utility function for a user maps each of the alternatives described above to a utility. If we assume that the function is additive, then the utility function will consist of a weighted sum of four subutility functions, one for each attribute. For example, the subutility function for price will map each of the four values for price—$1000, $1500, $2000, and $2500—to a utility. If the buyer prefers less expensive computers, then the subutility function might map the value of $1000 to a utility of 1, the value of $1500 to .6, the value of $2000 to .3, and the value of $2500 to a utility of 0. To calculate the utility to the user of a particular computer, we plug its attribute values into the utility function. This means that we use the subutility functions to compute the utility of each particular attribute value. Then we multiply each subutility result by the weight for that attribute, and we sum all of these weighted subutilities together to get the utility of the computer.

3.2 Assessment of Utility Functions

In order to use multiattribute utility theory to help a particular user make decisions, we need to find out what her utility function is—that is, we have to elicit her utilities. Here,

[1] See (Lukacs 2000) for a description of cases in which products are not considered to have certain attribute values.

we describe briefly two common methods used in assessing utility functions. For cases in which the utility function is too complicated to be decomposed, the utilities of outcomes or alternatives are assessed directly, generally using the first of the methods described here. In multilinear or additive cases, the task of eliciting utilities reduces to the task of eliciting the user's subutility functions and the weights for the attributes. We will deal with the simpler case, sometimes also assumed by others (Ha and Haddawy 1997), in which the subutility functions are assumed to be given and only the user's weights need to be learned.[2]

One of the best known methods for assessing the weights of attributes is the *standard gamble* approach (von Neumann and Morgenstern 1947), sometimes also called the *lottery weights* approach, a version of which can also be used to assess the utilities of alternatives directly. In the method used for assessing weights, an individual is told that she could receive a particular outcome with certainty or, alternatively, a lottery between two outcomes. The lottery offers her the worst possible value for all attributes with a probability of p and the best possible value for all attributes with a probability of $(1 - p)$. The certain option is an outcome with the best possible value for one particular attribute—call it Z— and the worst possible value for all other attributes. The individual is then asked which value of p makes her indifferent between the lottery and the certain option. Another method for asking the user about p is to present her with this kind of choice for different specific values of p. For each value of p, she is asked which option she prefers, until the point of indifference is reached; the sequence of values for p can be chosen in several ways, such as by choosing alternately high and low values or by gradually reducing p. Intuitively, if p is high, then the individual requires a very high chance of getting the best possible outcome in order to make up for not getting the best value for attribute Z for sure. This means that she values attribute Z very highly, and so the weight for Z is set accordingly. In fact, it is set to exactly p. If the additive utility model is appropriate, then the sum of the weights should be 1.

In *swing weighting* (von Winterfeldt and Edwards 1986), attribute *weights* are determined based on *ratings* that the user has assigned to all attributes, generally on a scale from 0 to 100. Each rating is divided by the sum of all ratings in order to arrive at normalized weights that sum to 1. The ratings of attributes are determined by comparing a number of hypothetical alternatives, one for each attribute. Each of these outcomes has the worst possible value for all attributes except for the attribute in question, for which it has the best possible value. In addition, there is another hypothetical outcome that has the worst possible value for all attributes. The user must rank these outcomes in order of preference. The worst-case outcome is given a rating of 0, and the highest-ranked outcome is given a rating of 100. The user must then assign the rest of the outcomes ratings between 0 and 100, perhaps by thinking of the satisfaction she would get from a good value for a particular attribute in terms of the percentage of satisfaction she would get from a good value for her most highly preferred attribute (Clemen 1996). The ratings are then normalized to obtain the weights in the utility function. Formally, once the user has assigned the ratings, the utility function is calculated as follows, where s_i stands for the rating of the hypothetical alternative associated with attribute i:

$$U(x_{1,\ldots,}x_n) = \sum_{i=1}^{n} \left(\frac{s_i}{\sum_{j=1}^{m} s_j} \right) U_i(x_i)$$

Both of the assessment methods described above, as is usual for traditional utility elicitation methods, assume that all questions are answered before the answers are used to

[2] This is a reasonable assumption to make in the domains of online recommenders and navigation aids, because these domains tend to have commonly recurring evaluation problems with objective assessment structures. For example, a processor with a speed of 833 mHz can fairly objectively be considered better than a processor with a speed of 133 mHz.

calculate a user's utility function. Some other methods for assessing weights include *pricing out*, which asks the user to think about tradeoffs between attribute values in terms of the amount of money it would take her to switch between them, as well as some newer approaches that have come out of the fields of market and operations research. Among these are the Analytic Hierarchy Process (Saaty 1980), which involves pairwise comparisons of all combinations of attributes, and Conjoint Analysis, which estimates weights using statistical methods, based on the way a user evaluates several hypothetical alternatives (Green and Wind 1973). For the purposes of this paper, we need not go into detail on the different assessment techniques; to understand our method of maintaining defensive numeric estimates, the reader need only be aware that there are some methods, such as the standard gamble approach, that assess weights or alternatives directly and some methods, such as swing weighting, that determine weights based on ratings of some kind.

3.3 Partial Utility Models

Partial utility models are models constructed on the basis of less information than a complete utility elicitation interview provides. In general, a utility model is partial if the mapping from alternatives or outcomes to utilities is not completely specified. In the particular case of multilinear functions, including additive utility functions, a partial utility model is one in which some attribute weights or some subutility functions are not completely specified.[3] For multilinear or additive utility functions with specified subutility functions, a utility model is partial if some of the attribute weights have not been specified.

If we use a method that assesses weights by means of ratings, such as swing weighting, a partial utility model is one in which ratings have not been given for some attributes. Clearly, this means that the weights of the attributes without ratings are unknown. It also means that even the attributes for which we have ratings are not known with certainty, since we do not know the normalizing factor that is used to transform ratings into weights.

4 Defensive Utility Estimation

The idea behind defensive utility estimation is straightforward. We want to track the lower bound of the user's utility for each alternative, at least with respect to our model. In fact, what we want to track is the greatest lower bound that can be known with certainty. This can be done for utility functions of any form, whether they are additive or a more general form of a multilinear function, or whether they are too complicated to be decomposed. Our focus is on multilinear, and particularly additive, utility functions for which the subutility functions are specified. The main contribution of this section is the technique for obtaining a defensive estimate for partial models of such functions when they are assessed via ratings. The example we will use is of an additive utility function, since this tends to be the kind of utility function for which ratings-based assessment methods are more commonly used. We also briefly treat the case in which the weights are assessed directly; this case is analogous to the one in which the utilities of alternatives, rather than weights, are directly assessed, and so we cover both decomposable and nondecomposable utility functions, although we concentrate on the multilinear cases.

---

[3] An additive utility model can also be partial if it is not known which attributes are included in it. For instance, this is the case in Linden, et. al. However, we treat only the case in which the relevant attributes are already known.

4.1 Estimation Technique

Because of the form of multilinear functions, we will be able to track the lower bound of the utilities for alternatives if we track the lower bound of each attribute weight. To do this, we need to specify both a way of interpreting unknown weights and a way of adjusting these weights, especially if weights, or the ratings used to compute them, can be changed by an assessment method not just once, but rather can be altered incrementally.

If an interview uses questions that set weights directly, such as the standard gamble questions described earlier, then to maintain a defensive estimate, we need one restriction in order to ensure a consistently improving estimate. This restriction is that, if a weight is changed more than once, it is adjusted only in an upwards direction. This is quite easy if the questions asked are of the style, described in Section 3.2, that ask the user to specify p directly, since each weight is affected by only one question. In that case, we simply treat unassessed weights as if they are 0, and we sum the weighted values of the attributes with known weights. If the user's value for p is determined by asking her about several different values of p, however, and we want to incorporate the information learned at each step into our estimate, then we must ask the questions so that the value of p is always increasing. If this requirement is met, then it is straightforward to maintain a defensive estimate of the utilities. Note that this approach can also be applied if the standard gamble questions are used to assess the utilities of individual alternatives, and so a similar approach can be used to estimate the lower bound of alternatives when a utility function cannot be decomposed.

A more complex situation arises when we use a method of assessing weights that requires normalizing ratings, as in swing weighting. In this case, we need to decide how to treat both unknown ratings and the normalizing factor in order to arrive at an estimate for weights. Analogous to the case of assessing the weights directly, in order to keep track of the lower bounds of weights, we need to restrict our adjustment of the utility model so that each rating can only be adjusted in an upward direction. Clearly, if a rating can decrease, the associated attribute weight can also decrease, which can result in a lower utility for some alternative. For example, consider the case of an online computer recommender system, as described in the previous section. Suppose that the system is designed so that if the user answers "no" to the question "Do you run more than two programs at once?", the rating for the memory attribute is decreased by 10 points, from 20 to 10. If the sum of the ratings for all attributes prior to the decrease is 50, then the weight for the memory attribute decreases from .4 to .25. Accordingly, the utilities of all alternatives will also decrease, and so the previous estimates were not lower bounds on their utilities.

On its own, the upward movement of each individual rating is not enough to guarantee that each attribute weight, and thus the estimate for the overall utility of each alternative, is also increasing. If a rating is increased during the course of an interview and the weights are then renormalized, the normalized weights of other attributes will decrease. Consider again the computer example. Suppose that at a given point in time, the ratings for price, hard disk size, memory, and speed are 10, 30, 20, and 10, respectively. The associated weights are then approximately .14, .43, .29, and .14. If the answer to a question increases the rating of price to 40, then once the ratings are normalized, the associated weights become .4, .3, .2, and .1. The weights of all attributes except price decrease. Depending on the subutility functions and on the attribute values of actual alternatives, the total utilities of alternatives may also go down. A visual depiction of this decrease can be seen in the last step in Figure 1 for the "standard normalization" method of estimation.

A defensive estimate of the utility function requires that we alter the standard method of normalizing the weights. In our proposed solution, we assume that until we know

otherwise, each attribute is capable of attaining the maximal rating of 100 points. When we then normalize the attribute weights—that is, divide the ratings by the normalizing factor—we use the sum of the *maximal* ratings rather than the sum of the actual ratings as the normalizing factor. In the case above, the ratings of 10, 30, 20, and 10 result in weights of .025, .075, .05, and .025, and with the increase of the price rating to 40, only the price weight changes, from .025 to .1. Again, this change is depicted as the last step in Figure 1, this time for the "defensive estimation" method.

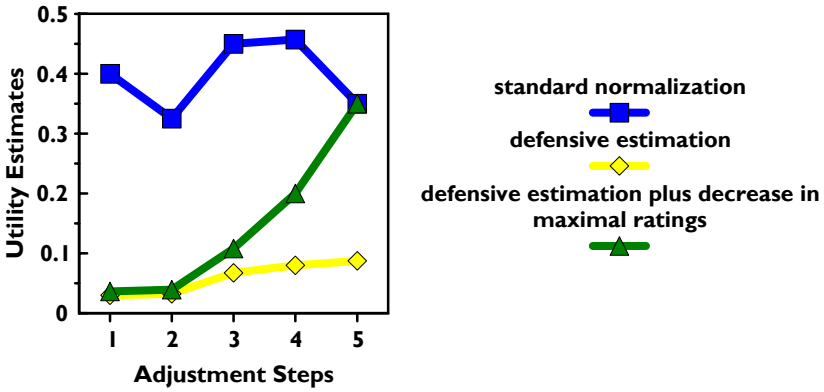

Figure 1 An example of the progression of utility estimates using three different styles of estimation. The graph shows the utility at each point during a series of 5 adjustments to ratings for a sample product with values of .1 for price, .4 for hard disk size, .7 for memory, and .5 for speed. All ratings begin at 0. At Step 1, the rating for hard disk size is set to 30, at Step 2, the rating for price is set to 10, at Step 3, the rating for memory is set to 20, at Step 4, the rating for speed is set to 10, and finally, in Step 5, the rating for price is set to 40. For the "defensive estimation plus decrease in maximal ratings" method, it is assumed that each time a rating is set to its final value (40, 30, 20, and 10 for price, hard disk size, memory, and speed, respectively), it is known that the rating will not be adjusted again.

We need not always assume that each rating can attain the maximal possible number of points. When we are certain that an attribute rating will not be changed anymore, we can reduce the maximal rating to the actual rating, thereby making the numeric estimate more accurate. In our example, if we know that the adjustment of the price rating from 10 to 40 is the final adjustment to this rating, we can reduce the maximal rating for the price attribute from 100 to 40. If none of the other ratings have yet been set to their final values, the sum by which we divide ratings in order to obtain weights is reduced from 400 to 340, and the weights would change from .025, .075, .05, and .025 to approximately .12, .09, .06, and .03. These weights are higher than those obtained if the maximal rating for price were not reduced. If the other ratings are also known to have been set to their final values, then the new normalizing factor is the sum of all of these ratings, which is 100. In this case, the weights reach their final values of .4, .3, .2, and .1. This is the case depicted in the last step of Figure 1, for the adjustment method "defensive estimation plus decrease in maximal ratings". Since we are maintaining the lower bound on the weights, an increase in the weights is an increase in the accuracy of the estimate. (Another extension to increase the accuracy of the estimate is discussed in Section 6.4.)

If all attribute ratings are completely assessed, we arrive at the standard situation in which each rating is divided by the sum of all ratings; the normalized weights then add up to 1. If we have not yet or will not reach that point, however, the normalized weights will not add up to 1, as is usual in utility functions. Although this differs from the standard interpretation of utility functions, it does guarantee that the utility estimate produced by this

partial utility function is a defensive one—it is always a lower bound on the estimate that would be produced by the completely specified utility function.

We can show more formally how our method of determining weights results in a defensive estimate of numeric utilities. The formula we use for determining weights is similar to that used in swing weighting, when all ratings are assessed at once, but now we divide the attributes into two groups: those whose ratings have been set to their final value and those whose ratings have not. As in the formula for swing weighting given earlier, let s_i denote the rating of a weight, which ranges from 0 to 100. Let C denote the set of indices of the attributes whose scores have been set to their final value, let A denote the indices of the remaining attributes, and let $|A|$ represent the cardinality of the set A. Then our defensive utility function can be written as follows:

$$U(x_{1,\ldots,}x_n) = \sum_{i=1}^{n} \left(\frac{s_i}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_i(x_i)$$

This is clearly guaranteed to be less than or equal to the result of the complete utility function, which is given by $U(x_{1,\ldots,}x_n) = \sum_{i=1}^{n} \left(\frac{s_i}{\sum_{j=1}^{m} s_j} \right) U_i(x_i)$, since for all s_i, $s_i \leq 100$. Moreover, on each successive adjustment to the utility function—whether the rating of an attribute is increased or whether it is finalized, so that the index of the attribute is moved from A to C—the defensive utility approaches the complete utility. We prove this below. Without loss of generality, consider a particular attribute $X_{m,}$ $1 \leq m \leq n$, that has not yet reached its final value (that is, $m \in A$) and the amount l by which it is increased after the user answers some question. Let i range from 1 to n, as before. Then:

$$\begin{aligned}
U(x_{1,\ldots,}x_n) &= \sum_{i=1}^{n} \left(\frac{s_i}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_i(x_i) \\
&= \left(\frac{s_m}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_m(x_m) + \sum_{i \neq n} \left(\frac{s_i}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_i(x_i) \\
&\leq \left(\frac{s_m + l}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_m(x_m) + \sum_{i \neq n} \left(\frac{s_i}{\sum_{j \in C} s_j + (|A| \cdot 100)} \right) U_i(x_i) \\
&\leq \left(\frac{s_m + l}{(s_m + l) + \sum_{j \in C} s_j + ((|A| - 1) \cdot 100)} \right) U_m(x_m) + \\
&\quad \sum_{i \neq m} \left(\frac{s_i}{(s_m + l) + \sum_{j \in C} s_j + ((|A| - 1) \cdot 100)} \right) U_i(x_i)
\end{aligned}$$

The first step involves pulling X_n out from the summation, the second holds as long as l is non-negative, which we have stipulated to be the case, and the third holds as long as long as $s_m + l \leq 100$, which is true because ratings have been stipulated to range from 0 to 100. In addition, the defensive utility estimate converges to the final utility when all ratings have been set to their final value.

4.2 Suitable Question Styles

Note that our method of defensive estimation does not rely on any particular way of asking questions to assess ratings. Its only requirement is that ratings are always adjusted in an upward direction, and its performance is improved with the knowledge of when a rating has been set for the final time. The implementor of a user modeling system is free to choose

the kind, number, and order of questions that suit her needs, as long as answers do not result in decreases in ratings; some ways of ensuring this are discussed below. The questions can also be chosen by an automated interview construction algorithm with full confidence that any choice of questions will result in an improved estimate of the user's utility function. Although there is a restriction on the way in which questions can adjust weight, we trade this limitation for increased freedom in the selection of sequences of questions. Once the mapping rules for a set of questions have been developed, any of these questions can be asked.

The defensive estimation method allows the implementor the possibility of using a set of questions for which ratings can be changed by more than one question, so that weights are adjusted incrementally. In this case, the implementor is responsible for ensuring that the ratings are increased on each adjustment. This can be achieved by increasing a rating by a set number of points, increasing a rating by a percentage, or by setting the ratings to exact numbers; with the last method, care will need to be taken with respect to the order in which rules are applied, lest a later rule set a rating to a lower value. The implementor is also responsible for determining when a rating has been changed for the last possible time, if she wishes to take advantage of the increased accuracy this affords.

On the other hand, the restriction to increasing ratings can easily be achieved by using questions that adjust the rating for each attribute only once. Individual swing-weighting style questions fit this description. Alternatively, the implementor could use a one-to-one mapping of questions to attributes, for which the answer to a question sets the rating for an attribute to an exact value once and for all. For example, the user could be asked whether she finds a particular attribute important, somewhat important, or not important, and the ratings for that attribute could then be set to 100, 50, and 0, respectively. If the implementor prefers not to ask about the attributes of alternatives directly—perhaps because she feels the user will not have enough knowledge of the attributes—it is still possible to achieve a one-to-one mapping of questions to attributes. To do this, we propose the following solution: the implementor can design "needs-oriented" attributes that are affected by "needs-oriented questions". For instance, in the computer case, each computer could be assigned a value for the needs-oriented feature "suitability for multimedia", in addition to its values for physical attributes; these values could be assigned either by an expert or by a specially-constructed algorithm. Then the user could be asked how important it is to her that her computer is suitable for multimedia, and the rating for this attribute could be adjusted in the same way as are the ratings for more tangible attributes. Such questions are similar in spirit to Chin and Porage's (2001) questions that determine stereotype membership and adjust weights accordingly (discussed in Section 6.1).

5 Application: Algorithm for Interview Construction

The defensive utility estimation technique grew out of our earlier work on utility-based decision tree optimization (Stolze and Ströbel 2001), which focused on a method for choosing questions in a utility elicitation interview. According to this method, the next question asked in an interview is the one that results in the greatest increase in the expected utility of the set of remaining alternatives, or the *focus set*. The stopping criterion in the interview is the lack of such an increase. The algorithm not only asks a user questions in order to improve its estimate of the user's utility function, but also uses the partially elicited utility functions in its reasoning about the next question to ask.

An early implementation of the original algorithm used a ratings-based framework in which ratings were initially set to default values and could then be either increased or decreased; in addition, more than one question could affect a weight. In experiments with this

implementation, we realized that in some situations the algorithm chose a sequence of questions such that some weights were further from their true values at the end of questioning than they were at the beginning. An improved and complete implementation of the algorithm thus requires some way of ascertaining that the final estimates are better than the initial estimates. One way to do this is to use a utility estimation method that ensures that utilities move progressively closer to their final values. The defensive estimation technique presented in this paper is our proposed solution.

The defensive utility estimation technique has an added advantage for our question-choosing algorithm. Not only is the utility estimate guaranteed to be moving closer to the final estimate, but now the choice of a question based on utility increases acts as a heuristic for choosing the question that most reduces the uncertainty about the utility function. If the chosen question is the one whose focus sets—the alternatives remaining after each of its possible answers—has the highest expected average utility for the user, then we are likely to have chosen the question whose focus sets have an expected average utility that is closest to the final expected average utility. That is, the average deviation between the estimated utilities and the final utilities is most likely smaller for the chosen question than for any others.

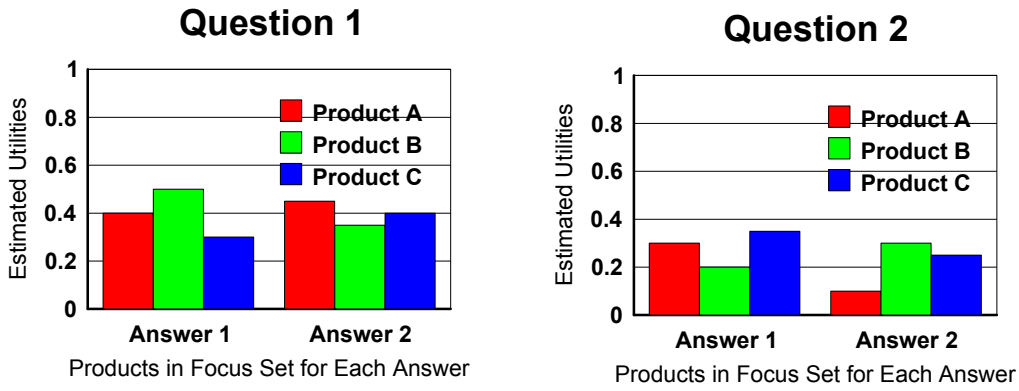

Figure 2 An example of the estimates of a user's utilities after two possible questions. According to our question-choosing algorithm, the question that is chosen is the one for which we have the greatest increase in average expected utility. To calculate this, first we calculate the average utility of the focus set for each answer, and then we sum over all possible answers to a question, weighted by the probability of each answer. In this example, Question 1 would be chosen over Question 2, no matter what the probabilities of the answers are. (Note that a question would be chosen only if it provides an increase over the current average utility of alternatives, but this is guaranteed to be the case here if our defensive utility estimation technique has been used; since the utilities cannot decrease, and since Question 2 has lower utilities than Question 1, Question 1 must provide an increase over the current average utility.)

In effect, the revised algorithm chooses the question for which it is most likely that an answer will have a strong effect on the estimated utility function; this is the question that is expected to reduce most greatly uncertainty about the average expected utility. This means that the estimates of some alternatives become more accurate, but, unfortunately, there is no way of knowing whether the estimates that become more accurate are those of the alternatives that would be most highly ranked by the user's true utility function. There is also no guarantee that those alternatives that appear most highly ranked at a given point will ultimately be the best. However, this is a problem that besets any ranking of nondominated alternatives based on partial utility models. Any method for ranking alternatives based on

partial information can only serve as a heuristic. In our case, the choice criterion acts as a heuristic for choosing a question that will be likely to result in more alternatives that can be shown to be closer to a satisfactory score. While it is hoped that this will provide a more accurate ranking of alternatives, even if the results of the ranking do not match a ranking based on the user's true utility function, the user can nonetheless rely on certain properties of defensive numeric estimates to aid her in her decision making. For example, since the estimates represent certain information about the minimal utility of alternatives, the user may be able to use this information to reason about the alternatives in a way that she can't with other methods, such as default estimates, for which the user is unaware of whether the utility estimate for an alternative is higher or lower than its actual utility.

If we use the original stopping criterion suggested in our earlier paper —the lack of an increase in the expected utility of a focus set —we run into the complication that any question that is capable of altering the utility function will, by the definition of defensive utility estimation, increase the average expected utility of the set. In this case, the question-choosing algorithm will not terminate until all such questions have been asked. If we don't wish to ask all of these questions, we can use the algorithm as a means for determining the order of questions that will most quickly improve the utility estimates; we can then either impose a limit on the number of questions to ask or allow the user to decide how many questions she wishes to answer. Alternatively, the algorithm can be terminated earlier if the expected utility of the focus set is reduced by other means, even though the utilities of all alternatives are increasing. For instance, the calculation may take into account the rate of user attrition, so that the score of a question is reduced by a certain amount (perhaps a percentage) based on how likely a user is to leave the interview upon being asked that question. In this case, the algorithm essentially balances the increase in accuracy of the utility function against the risk of losing a user. Another option is to specify a minimal increase in the average expected utility of the remaining products, such that questioning stops when no question can provide at least that increase. Intuitively, this means that questioning stops when the improvement in the accuracy of the utility function is not great enough to be worthwhile. Of course, because the algorithm is myopic (that is, it only looks at the benefit provided by asking one question at a time), the algorithm could miss a combination of questions that provides greater than the minimal increase, but since the problem of evaluating all sequences is intractable, similar risks are run by all question-choosing algorithms.

6 Discussion

Here we discuss some benefits of, possible problems with, and potential extensions to our technique. To place the discussion of our algorithm in context, we begin with an overview of related work.

6.1 Related Work

Because classical decision theory treats primarily complete utility models, it is difficult to say what the right way of interpreting partial utility models is. Different interpretations of partial utility models depend on what the unknown variables are and how the system treats them. In this section, we briefly describe five of the existing approaches to partial utility models: one that uses default weights, one that uses ranges, two that reason probabilistically, and one non-numeric approach. Like ours, all of these approaches combine a method for interpreting a partial utility model with a method for choosing the questions that further refine this model. We discuss how each of these approaches resolves the question of

unknown variables so that a partial utility model can be used to make decisions or calculations. We also discuss what ramifications each solution has for systems that automatically construct utility elicitation interviews.

Linden et al. (1997) assume an additive utility function, in which the unknown variables are attribute weights and subutility functions, and they treat unknown weights by assigning them default values. Their system has the additional feature of allowing new attributes— constraints, in their terminology—to be added to the utility function, and so their user model might be partial not just in that it has some unassessed weights, but also in that it does not contain all of the attributes that appear in complete utility function. Weights are refined based on how the user critiques alternatives presented to her. In this system, it is the user who decides when to stop the interview process, and so if default values for weights are incorrect and result in an inappropriate list of products, the user can continue to critique alternatives until the utility model results in a satisfactory list of products. However, the use of default values for unknown weights could prove problematic if an algorithm or system, not the user, determines when to stop asking questions, or if the user does not wish to continue answering questions. If such a system does not ask all possible questions, some weights may remain set to their default values, and the numeric estimate could then be worse at the end than at the beginning. For example, if the ending numeric estimate is slightly higher than the original estimate, but in fact one of the weights that remains set at its default value should be quite low, the true utility might actually far below both the original and ending estimates. Instead of using default weights, our system reasons about an unknown weight by treating its value as if it equals only as much as can be known with certainty, thus avoiding the need to revise default assumptions.

Chin and Porage (2001) also assume an additive utility functions. They deal with uncertainty over weights, and subutility functions as well, by representing ranges over them. A question is chosen based on how much it reduces uncertainty, and the interview terminates when the reduction in uncertainty cannot compensate for the user's impatience with questioning. In their algorithm, the means of the ranges are used to determine the alternative with the highest utility for the user. However, because of the nature of ranges, there is no guarantee that the estimated utility of an individual alternative will be closer to its actual utility at the end of the interview than at the beginning, and so there is no guarantee that the final solution will be better than one produced at the beginning. For example, if the true value of a weight is just below the midpoint of the initial range, and the range is then reduced to its lower half, the average of the new range is now further from the true value of the weight than the average of the initial range was. Although a case like this may be unlikely to arise in practice, nothing in the structure of the reasoning mechanisms rules it out. This seems to be an undesirable property in an interview system that aims to better serve the user by gaining a more accurate estimate of her utility function. However, the spirit of our approach is quite similar to that of Chin and Porage's. In effect, what we do is simply to take the lower end of ranges and use this endpoint as our estimate, because we can reason consistently with this endpoint in a way that we can't with a changing average.

Two approaches reason probabilistically about the utilities of alternatives. Chajewska et al. (2000) present a technique that can be used for reasoning about utility functions that do not exhibit the decomposable structure assumed by the approaches above; they also present an extension of their technique to cases in which there is correlation among outcome utilities. In both cases, they represent uncertainty directly over individual outcome utilities, rather than over attribute weights, by representing a probability distribution over these utilities, and they use the mean of utilities under this distribution in an influence diagram to recommend strategies. Jameson et al. (1995) assume a decomposable utility function and use a Bayesian

network to infer the weights of attributes, and then to infer the overall utilities of alternatives. Although there are significant differences between the systems of Jameson and Chajewska, both predict numeric utilities of alternatives based on current probability distributions, and they both use the predicted utilities in making recommendations; they also both choose questions according to the reduction in uncertainty they provide. However, we can't be sure that the average, or the result of Bayes net inference, under an updated distribution is closer to the actual utility than the result under the previous distribution. This need not be a problem for an interview construction method if some other means of testing for the improved accuracy of a final estimate is used. For example, Chajewska et al. use an additional stopping criterion that predicts the estimated regret of terminating the questioning process. However, as noted in Section 1.1, there are some situations in which we nonetheless want a consistently improving estimate[4]. For such situations, our defensive estimation technique is one way of guaranteeing the necessary increase in the accuracy of estimates. Certainty, a probability distribution could also be used to estimate the lower bound on the utilities of alternatives. However, if all that is needed is a lower bound, our simple defensive estimation technique is easier to use than a sophisticated probabilistic system, and might also be more appropriate for domains in which we do not have enough historical data to estimate accurate probabilities.

Lastly, Ha and Haddawy (1997) deal with unknown weights in an additive utility function in a non-numeric way. They use the known weights to determine which alternatives are dominated by others, and they eliminate these alternatives from consideration, without ranking or scoring the remaining alternatives. They use an iterative process to determine the ratios of particular unknown weights—by using lottery-style or "trade-off" questions that assess indifference—and then find more suboptimal alternatives. A later paper (1999) presents an approach that can eliminate dominated alternatives for multilinear utility functions. The approach of eliminating alternatives is clearly a useful one, but, as discussed in Section 1.1, we feel that it would be useful to have a numeric interpretation of partial utility functions, in addition to approaches that eliminate dominated alternatives. Our approach could be combined with an approach such as Ha's in order to obtain the benefits of both; for instance, our approach could be used to rank the nondominated alternatives as determined by Ha's algorithm.

6.2 Benefits of Defensive Utility Estimation

One of the causes of difficulties in producing interpretable numeric estimates from partial utility models is that a partial utility model is by definition one in which there is some uncertainty about a user's true utilities. However, when a partial utility model is used to produce a numeric estimate, any representation of uncertainty is lost in the translation to a single numeric value. A defensive estimate solves this problem by using only that information which is certain in calculating the defensive estimate, and, on a higher level, we are aware of the direction in which our estimate is uncertain, if not of the amount; we then know exactly what kind of information the estimate conveys about a user's true utilities.

Our defensive utility estimation technique has the properties that we found to be missing in other numeric estimates. Its meaning is easily interpretable, and the estimation is guaranteed to become more accurate as time goes on. Thus, it can be used in the situations in which, as described earlier, an interpretable and increasing numeric estimate is beneficial, such as when a user exits an interview early, when recommendations are displayed in parallel

[4] One of these situations is when a user decides to exit an interview early. Note that in the medical domain treated by Chajewska et al., it is reasonable to assume that users will answer all questions asked of them, given the seriousness of health-related decisions.

with questions, and when an implementor desires the possibility to ask any sequence of questions. Our numeric estimation technique also provides an inverse benefit to the benefits of approaches that eliminate alternatives: it cannot eliminate candidates from consideration, but it can prove that a candidate attains at least a certain utility. It can thus show that a candidate is "good enough", although perhaps not optimal.

Our defensive estimation technique also proves to be useful for another important task: it allows us to present an algorithm for constructing interviews that reasons about which question most reduces uncertainty about utilities. The amount of reduction in the uncertainty of a partial utility model is a common measure used for choosing questions, as described in the section on related work. However, in other systems, a reduction in uncertainty does not necessarily result in the immediate increased accuracy of a numeric estimate. In our algorithm for choosing questions, however, the amount by which numeric estimates are increased—and so the amount by which they become closer to their final value, and thus more accurate—also serves as way of measuring the reduction in one kind of uncertainty about the utility model (that is, uncertainty about the average utility of a set).

6.3 Difficulties

We have stated throughout the paper that we would like a numeric estimate that consistently increases in accuracy, which is not the case with approaches that use default values, averages, or probabilistic predictions. However, it is possible that the constant improvement in accuracy of our defensive estimation technique comes at the cost of having a more inaccurate estimate at the start. Indeed, since the starting estimate for each alternative is 0, the initial estimate produced by our method has a good chance of being further from the true utilities of a user than the estimate produced by the other numeric methods. However, even if that is the case, the defensive estimate still has the advantage of being clearly interpretable as a lower bound on the utility, whereas the meaning of estimates produced by the other methods is not clear.

The usefulness of the numeric defensive approach presented here seems to depend in part on a different way of viewing utility functions. The goal of multiattribute utility theory has traditionally been to find the optimal alternative in a set of alternatives, and in this sense the utilities assigned can take on a comparative meaning. Our approach is to view the utilities more as scores, which can be used to judge the alternatives independently of one another. This is what makes it possible for us to talk about a utility being "satisfactory" for a user, instead of simply using utilities to compare alternatives in order to find the best one. While this differs from the approach of classical decision theory, it may be a realistic view in many cases. It is possible that a user needn't find the optimal alternative in a set, but in fact would be satisfied with one of several alternatives; she may look for alternatives whose utility for her is above a certain threshold. It would be interesting to think further about the relation between the standard way of viewing utilities and the view that seems to be entailed by the approach presented here.

Finally, we should state the rather obvious qualification to our claims about defensive utility estimation. Our estimates are lower bounds on the user's utilities only with respect to our model. Our model could be incorrect, in that the way in which it adjusts weights might not correspond to the user's judgments. (This is a hazard with any user modeling system.) What we mean by "defensive utility estimation" is that the estimate at any given point is always less than or equal to the estimate that would be produced by a full utility function using the same model.

6.4 Possible Extensions

In the estimation technique presented here, only the minimal rating for attribute weights is adjusted by a user's answers to questions. The maximal rating is reset only once, after the final change to the minimal rating; until it is set to equal the minimal rating, it remains at its highest value. One possible extension to our estimation technique would be to have a user's answers affect the maximal rating of an attribute as well as its minimal rating. This could be used for two purposes: to compute a more accurate defensive estimate of utilities, and also to compute an "optimistic", rather than a defensive, estimate of utilities. For the first purpose, we use the maximal ratings to normalize the weights, as before. By decreasing maximal ratings, we decrease the normalizing sum, which increases the weights, thereby bringing them closer to their final value. For the second purpose, we simply invert the process of maintaining a lower bound on the weights and utilities. Instead of dividing the minimal rating by the sum of all maximal scores, we divide the maximal rating by the sum of all minimal ratings in order to compute the maximum possible utility of each alternative.[5] This would allow us to eliminate alternatives whose maximal score is not satisfactory, as well as to present the user with additional information to make her decisions. We could also use the upper and lower bounds to perform reasoning that uses ranges. Note that if we were to adjust both the minimal and maximum ratings of a weight, we would confront some issues that need to be dealt with by all approaches that use ranges, such as ensuring that our adjustment method does not allow the minimal and maximal scores to cross.

The next item in our future plans is to conduct empirical evaluations with actual users (Chin 2001). We plan to design experiments that can evaluate both the usefulness of defensive utility estimation on its own and the success of our question-choosing algorithm. One claim about defensive utility estimation that we would like to test is whether knowledge of the minimum guaranteed utility of an alternative is useful to users, and if so, whether it is more useful than other numeric estimates. With regard to the algorithm, we would like to determine whether choosing the question with the greatest decrease in uncertainty about expected utilities in fact helps the users to find a suitable product more quickly and whether it increases their confidence in the proposed selection.

7 Conclusion

In this paper, we discuss a solution to the problem of interpreting partial utility models numerically. In order to do this, we need to solve the problem posed by potential decreases in the accuracy of numeric estimates during utility elicitation interviews, such as interviews built using the utility-based interview tree optimization technique of Stolze and Ströbel (2001). Our solution is to use a defensive estimation technique that tracks the lower bound of a user's utilities. This achieves the desired result of guaranteeing increasing accuracy during utility elicitation interviews, and it also renders the meaning of a numeric estimate clearly understandable. Moreover, in the particular case of utility-based optimization, it allows the interview construction algorithm to reason about reducing uncertainty in a partial utility model.

Defensive utility estimation assumes an additive utility model and the presence of subutility functions, and it requires that the effects of questions alter attribute ratings or weights in only an upwards direction. However, it is independent of the kind of elicitation questions used, the number of questions asked, and the sequence in which questions are asked. This allows implementors of recommender systems or navigation aid systems the

[5] If the sum of all minimal ratings is 0, we can set the upper bound of each weight to 1, since it is possible for each weight to equal 0, and thus any individual weight has the possibility to carry all of the weight of the utility function.

freedom to choose questions in their preferred way or to construct their own algorithms for choosing questions, as long as the questions follow the prescribed guidelines. Implementors who wish to choose their own questions can thus use defensive utility estimation on its own as a way of reasoning consistently with partial utility models and presenting users with understandable choices. Alternatively, defensive utility estimation can be used in conjunction with the algorithm for interview construction presented here, which chooses questions that tend to most improve the partial models used for recommending alternatives. Overall, because the results of the presented techniques can be consistently interpreted, their application could allow for decision support that is both user- and implementor-friendly.

References

Bacchus, F. and A. Grove: 1995, 'Graphical Models for Preference and Utility'. *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, pp. 3-10.

Boutilier, C., R. Brafman, C. Gelb, and D. Poole: 1997, 'A Constraint-Based Approach to Preference Elicitation and Decision Making'. *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, Stanford, CA, pp. 19-28.

Chajewska, U., D. Koller and R. Parr: 2000, 'Making Rational Decisions Using Adaptive Utility Elicitation'. *Seventeenth National Conference on Artificial Intelligence*, Austin, TX, pp. 363-369.

Chin, D.: 2001, 'Empirical Evaluation of User Models and User-Adapted Systems'. *User Modeling and User-Adapted Interaction* **11**, pp. 181-194.

Chin, D. and A. Porage: 2001, 'Acquiring User Preferences for Product Customization'. *Eighth International Conference on User Modeling*, Sonthofen, Germany, pp. 95-104.

Clemen, R.: 1996, 'Making Hard Decisions: An Introduction to Decision Analysis'. Pacific Grove, CA: Duxbury Press.

Green, P. and Y. Wind: 1973, 'Multiattribute Decisions in Marketing: A Measurement Approach'. Hindale, IL: The Dryden Press.

Green, P. and V. Srinivasan: 1990, 'Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice'. *Journal of Marketing*, pp. 3-19.

Ha, V. and P. Haddawy: 1999, 'A Hybrid Approach to Reasoning with Partially Elicited Preference Models'. *Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, pp. 263-270.

Ha, V. and P. Haddawy: 1997, 'Problem-Focused Incremental Elicitation of Multi-Attribute Utility Models'. *Thirteenth Conference on Uncertainty in Artificial Intelligence*, Providence, RI, pp. 215-222.

Jameson, A., R. Schäfer, J. Simons and T. Weis: 1995, 'Adaptive Provision of Evaluation-Oriented Information: Tasks and Techniques'. *Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 1886-1895.

Keeney, R.L. And H. Raiffa: 1976, 'Decisions with Multiple Objectives: Preferences and Value Tradeoffs'. New York: Wiley.

Linden, G., S. Hanks and N. Lesh: 1997, 'Interactive Assessment of User Preference Models: The Automated Travel Assistant'. *Sixth International Conference on User Modeling*, Chia Laguna, Italy, pp. 67-78.

Lukacs, G.: 2000, 'Decision Support Under Imperfections in Electronic Commerce'. *Eleventh International Workshop on Database and Expert Systems Applications*, Greenwich, United Kingdom, pp. 538-542.

Porage, A.: 1999, 'A Framework for the Interactive Customization of Products and Services'. Dissertation submitted to the University of Hawaii. Ann Arbor, MI: UMI Dissertation Services.

Saaty, T.: 1980, 'The Analytic Hierarchy Process'. New York: McGraw Hill.

Stolze, M. and M. Ströbel: 2001, 'Utility-based Decision Tree Optimization: A Framework for Adaptive Interviewing'. *Eighth International Conference on User Modeling*, Sonthofen, Germany, pp. 105-116.

Tan, S. and J. Pearl: 1994, 'Qualitative Decision Theory'. *Twelfth National Conference on Artificial Intelligence*, Seattle, Washington, pp. 928-932.

Von Neumann, J. and O. Morgenstern: 1947, 'Theory of Games and Economic Behavior'. Princeton, NJ: Princeton University Press.

von Winterfeldt, D. and W. Edwards: 1986, 'Decision Analysis and Behavioral Research'. Cambridge: Cambridge University Press.