Preference Elicitation and Generalized Additive Utility

Darius Braziunas
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
darius@cs.toronto.edu

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
cebly@cs.toronto.edu

Abstract

Any automated decision support software must tailor its actions or recommendations to the preferences of different users. Thus it requires some *representation* of user preferences as well as a means of *eliciting* or otherwise learning the preferences of the specific user on whose behalf it is acting. While additive preference models offer a compact representation of multiattribute utility functions, and ease of elicitation, they are often overly restrictive. The more flexible generalized additive independence (GAI) model maintains much of the intuitive nature of additive models, but comes at the cost of much more complex elicitation. In this article, we summarize the key contributions of our earlier paper (UAI 2005): (a) the first elaboration of the semantic foundations of GAI models that allows one to engage in preference elicitation using *local* queries over small subsets of attributes rather than *global* queries over full outcomes; and (b) specific procedures for Bayesian preference elicitation of the parameters of a GAI model using such local queries.

1 The Preference Bottleneck

The increased emphasis on computational decision support tools in decision analysis and AI has brought into sharp focus the need for *automated preference elicitation.* Such software must have the ability to tailor its actions or recommendations to the specific needs and preferences of different users; thus it requires some means to obtain such preference information. While decision theory presumes that both the dynamics of a decision problem and preferences are known, it is often the case that the dynamics (i.e., the mapping from actions to outcomes) are fixed across a variety of users, with preferences varying widely. For example, in a travel planning scenario, the distribution over outcomes associated with choosing a specific flight from Toronto to Boston is the same for any user (e.g., the odds of a delay greater than one hour, arriving during rush hour, losing luggage), but each user's strength of preference for such outcomes can vary considerably. This *preference bottleneck*—the need to obtain individuated preference information—is one of the most formidable obstacles to the widespread deployment of computer-aided decision support systems.

Much work in decision analysis, economics, marketing and AI has dealt with the problem of preference elicitation, but challenges remain. One of the most pressing is the ability to handle large, multiattribute outcome spaces. Specifically, when outcomes in a decision problem are defined as instantiations of a set of *attributes* (e.g., the relevant factors influencing the desirability of a flight in the scenario above), the exponential size of the outcome space makes both representation and elicitation of preferences difficult. *Factored* utility models, which decompose preferences into more manageable components by making independence assumptions, can help overcome both difficulties [9; 6].

Additive models form a specific class of factored models that is used widely in practice: one assumes that the strength of preference for the values of one attribute can be expressed independently of the values of others (e.g., a delay of one hour equally despised whether or not your bags are lost). Additive models allow not only concise representation of a utility function, but can be *elicited* using almost exclusively *local queries*. These ask a user for her strength of preference for each attribute in isolation; *global queries* require comparison of full outcomes. This is significant because a user need only answer queries that relate to the underlying structure of her preferences. For example, a user can express preference for length of flight delay (20 minutes vs. one hour) independently of her preference for actual departure time, missed connection, lost baggage, and other attributes. This stands in contrast with asking a user to compare full outcomes that involve the *joint instantiation of all attributes*, a cognitively difficult task if more than a handful of attributes are involved.

The appeal of additive models is thus considerable. However, the strong independence assumptions required make their applicability suspect in many cases. The *generalized additive independence (GAI)* model [6; 1] allows for a similar additive decomposition of a utility function, but where overlapping subsets of attributes are the factors involved rather than single attributes. This representation is completely general—it can capture any utility function—yet in practice is quite intuitive and natural, and typically yields a compact decomposition of preferences. The power of this representation will be illustrated in the next section.

The GAI model, while offering the same compactness and naturalness of representation as the much more widely used additive model, had not yielded (to date) the same advantages with respect to elicitation. Specifically, no (semantically sound) elicitation process for GAI models had yet been elaborated that allowed one to ask *local queries* about preferences over small subsets of attributes. Gonzales and Perny [7], for example, recently described a sound elicitation process for GAI models involving queries over full outcomes. In Section 3 of this paper, we summarize the results of our recent UAI 2005 paper [4] which provides the first semantically sound elicitation procedure for GAI models that relies on only local queries. Unlike additive models, GAI models require much more care in calibration because of the possible overlap of factors (sharing of attributes). Our process accounts for this explicitly without requiring the user to express preferences for full outcomes. This is significant because it allows one to exploit the generality, naturalness, compactness and much wider applicability of GAI models, without losing the advantage (offered by more restrictive additive models) of elicitation based on local queries.

A second important trend in preference elicitation, especially in AI, is the recognition that eliciting complete and precise preference information comes at a cost, and that the improvement in decision quality some piece of preference information offers may not be worth the cost incurred. For example, suppose that after eliciting partial preference information from a user, a decision support system determines that the expected value (say, in dollar terms, after accounting for price) of a flight A (given the system's knowledge of odds of on-time departure, flight delays, etc.) is between \$400 and \$460, while that of flight B is between \$450 and \$560 (all other flights are less preferred). If the additional preference information needed to determine which of A or B is in fact optimal requires considerable additional interaction, it may be reasonable to terminate elicitation and simply recommend B. Specifically, we want the system to recognize that it can improve the quality of its recommendation by *at most* \$10 with further elicitation (i.e., by making decision B now, it will either have recommended the optimal decision or one that is within \$10 of optimal). If the cost of elicitation outweighs this improvement, the process should terminate.

More generally, probabilistic information about a user's utility function can be used to make this assessment. Recently, Bayesian models of preference elicitation have been proposed that do just this [5; 2; 8]. With distributions over utility functions, queries are determined based on their *expected value of information (EVOI)*: in other words, the value of a query is determined by the expected (with respect to possible responses) improvement it will offer in terms of the quality of the decision. The optimal query is that with highest EVOI (less query cost), and the elicitation process continues only as long as EVOI is positive.

The second contribution of our UAI 2005 paper is the development of a Bayesian elicitation strategy along these lines that exploits GAI structure and asks the type of local queries discussed above. Empirically, we are able to show that good or even optimal decisions can be made with very imprecise information about a user's utility function, and that our Bayesian elicitation strategy asks appropriate queries (i.e., determines good decisions with very few queries).

2 Multiattribute Utility Models

We begin by briefly summarizing key prior results on the semantic foundations of elicitation in multiattribute models.

2.1 Additive Models

We assume a set of attributes $X_1, X_2, \ldots, X_n$, each with finite domains. The set $\mathbf{X}$ of possible *outcomes* of decisions made by a system on behalf of some user correspond to instantiations of these attributes. For instance, in a real-estate setting, attributes may be house properties such as L (lot size), D (distance to a park), and P (pool). A user on whose behalf we make decisions has not only qualitative preferences over these outcomes, but also strength of preference as captured by a *utility function* $u : \mathbf{X} \mapsto \mathbb{R}$. A utility function serves as a quantitative representation of strength of preferences, and can viewed as reflecting preferences over *lotteries* (distributions over outcomes) [9]; specifically, one lottery is preferred to another if and only if its expected utility is greater. Let $\langle p, \mathbf{x}^\top; 1-p, \mathbf{x}^\perp \rangle$ denote the lottery where the best outcome $\mathbf{x}^\top$ is realized with probability p, and the worst outcome $\mathbf{x}^\perp$ with probability $1-p$; we refer to 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 $\mathbf{x}^\top$ to 1, and the utility of the worst outcome $\mathbf{x}^\perp$ to 0. In such a case, if a user is indifferent between some outcome $\mathbf{x}$ and the *standard gamble* $\langle p, \mathbf{x}^\top; 1-p, \mathbf{x}^\perp \rangle$, then $u(\mathbf{x}) = p$.

Given the exponential size of outcome space $\mathbf{X}$, simply representing u can be problematic. Fortunately, in many circumstances an *additive model* [9] can be used to compactly represent u. Under a strong independence assumption—specifically, that the user is indifferent among lotteries that have same marginals on each attribute—u can be written as a sum of single-attribute *subutility functions*:

$$u(\mathbf{x}) = \sum_{i=1}^{n} u_i(x_i) = \sum_{i=1}^{n} \lambda_i v_i(x_i).$$

This factorization exploits subutility functions $u_i(x_i) = \lambda_i v_i(x_i)$, which themselves depend on *local value functions* v_i and scaling constants λ_i. In our simple example, if the user's utility for houses is given by an additive decomposition, the user need only assess local value functions v_L (expressing strength of preference for different lot sizes), v_D and v_S, and tradeoff weights λ_L, λ_D and λ_S expressing the relative "importance" of each attribute.

Significantly, local value functions can be assessed using only local queries involving only the attribute in question [9].[1] For instance, to assess v_L, our system need only ask queries involving "anchor" levels of the attribute x_L, without requiring any consideration of the values of other attributes. One possible (but not the most practical; see below) way to determine the local value of a 10,000 sq.ft. lot is to ask a user for the probability p at which she would be indifferent between getting that lot for sure and taking a hypothetical gamble that

[1] We discuss elicitation of u_i and v_i in terms of "complete" and exact assessment for now. We consider the implications of partial assessment of preferences in Sec. 4.

offers the best lot size (e.g., 20,000 sq.ft.) with probability p and the worst possible lot (e.g., 3,500 sq.ft.) with $1-p$ (assuming fixed values of all remaining attributes).

Assessing the tradeoff weights λ_i cannot be accomplished, of course, without calibration of the v_i across attributes. This calibration requires the user to compare a small number of full outcomes. Fortunately, the number of such queries is linear in the number of attributes, and involves varying only one feature at a time from a specific fixed (or default) outcome (typically the worst instantiation of all attributes). It is this ease of assessment that makes additive utility the model of choice in almost all practical applications of multiattribute utility theory.

2.2 Generalized Additive Models

GAI models [6; 1] provide an additive decomposition of a utility function in situations where single attributes are not additively independent, but (possibly overlapping) *subsets* of attributes are. Such models are completely general and can be used in many realistic situations where simple additive models are clearly not expressive enough. In the real-estate example with attributes L, D, and P, complete additive independence may not hold, but some partial independence may. For instance, the smaller the lot the more valuable close proximity to the park; and, the value of a pool may be diminished by a smaller lot. GAI models can capture such dependencies by decomposing the utility of a full outcome into arguably natural subutilities over overlapping attribute sets $\{L\}$ (utility of a specific lot size), $\{L, D\}$ (utility of a park distance given lot size) and $\{L, P\}$ (utility of a pool given lot size).

Formally, assume a given collection $\{I_1, \ldots, I_m\}$ of possibly intersecting attribute (index) sets, or *factors*. These sets of attributes are *generalized additively independent* if and only if the user is indifferent between any two lotteries with the same marginals on each set of attributes [6]. Furthermore, if GAI holds, the utility function can be written as a sum of *subutility* functions [6]:

$$u(\mathbf{x}) = u_1(\mathbf{x}_{I_1}) + \ldots + u_m(\mathbf{x}_{I_m}).$$

In our example, $u(L, D, P) = u_1(L) + u_2(L, D) + u_3(L, P)$.

In simple additive models, we can elicit information about local value functions in isolation, and then use global queries to determine scaling parameters. With GAI utilities, the matter is less straightforward, because the values of subutility functions u_i do not directly represent the local preference relation among the attributes in factor i. Intuitively, since utility can "flow" from one subutility factor to the next through the shared attributes, the subutility values do not have an independent semantic meaning.

Partly because of such difficulties, the existing elicitation procedures for GAI models rely on full outcome queries which circumvent the problem of local elicitation and global calibration issues. Such semantically sound procedures were implicitly described by Fishburn in [6] and, more recently, by Gonzales and Perny [7]. However, by resorting to full outcome queries, we lose some of the advantages of additive models and fail to exploit the decomposition of utility functions *during the elicitation process*. The next section summarizes our contribution in addressing these issues.

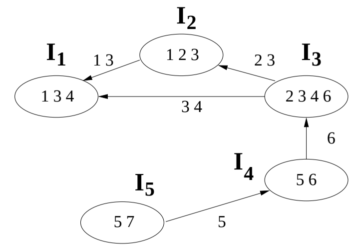

Figure 1: GAI graph. The nodes are GAI sets of attributes; edges are labeled with shared attributes. The utility function can be decomposed as $u(x_1, \ldots, x_7) = u_1(x_1, x_3, x_4) + u_2(x_1, x_2, x_3) + u_3(x_2, x_3, x_4, x_6) + u_4(x_5, x_6) + u_5(x_5, x_7)$.

3 GAI elicitation with local queries

GAI utility functions can be elicited using local queries in a manner related to local elicitation in simpler additive models. The main steps involved are: (a) eliciting *local value functions* v_i defined over the same factors as the subutility functions u_i, that (unlike the u_i) represent local preferences in a semantically sound way; and, (b) eliciting global utilities of a few "key" full outcomes to calibrate the local value functions.

Before we describe how to elicit the local value functions, we need a few more definitions. At the outset, we assume a decomposition of attributes into m factors; the attributes in factor i are indexed by the set I_i. We also designate one (arbitrary) full outcome as a *default* outcome $\mathbf{x}^0$. Attributes instantiated at their default levels will provide a reference point for consistent global scaling of locally elicited value functions. Finally, we introduce the notion of a *conditioning set* C_i of factor i as the set of all attributes that share GAI factors with attributes in I_i. For example, the conditioning set of factor 5 in Figure 1 consists of a single attribute x_6 that, once fixed, "blocks" the influence of other factors on factor 5.

After an appropriate rearrangement of indices, an outcome $\mathbf{x}$ can be written as $(\mathbf{x}_i, \mathbf{x}_{C_i}, \mathbf{y})$, where $\mathbf{y}$ are the attributes that are neither in I_i nor C_i. Once the attributes in the conditioning set are at default level, we can prove the following theorem [4]:

Theorem Under GAI conditions, if

$$(\mathbf{x}_i, \mathbf{x}^0_{C_i}, \mathbf{y}) \sim \langle p, (\mathbf{x}^\top_i, \mathbf{x}^0_{C_i}, \mathbf{y}); 1-p, (\mathbf{x}^\perp_i, \mathbf{x}^0_{C_i}, \mathbf{y})\rangle, \text{ then}$$
$$(\mathbf{x}_i, \mathbf{x}^0_{C_i}, \mathbf{y}') \sim \langle p, (\mathbf{x}^\top_i, \mathbf{x}^0_{C_i}, \mathbf{y}'); 1-p, (\mathbf{x}^\perp_i, \mathbf{x}^0_{C_i}, \mathbf{y}')\rangle,$$

for any $\mathbf{y}'$ ($\sim$ denotes the indifference relation). Therefore,

$$(\mathbf{x}_i, \mathbf{x}^0_{C_i}) \sim \langle p, (\mathbf{x}^\top_i, \mathbf{x}^0_{C_i}); 1-p, (\mathbf{x}^\perp_i, \mathbf{x}^0_{C_i})\rangle.$$

That is, as long as attributes in the conditioning set of I_i are fixed, the remaining attributes do not influence the strength of preference of local outcomes $\mathbf{x}_i$. Thus, we can perform *local* elicitation with respect to local anchors $\mathbf{x}^\top_i$ and $\mathbf{x}^\perp_i$ without specifying the levels of the $\mathbf{y}$ attributes.

A *local value function* $v_i(\cdot)$ can be defined to represent conditional local preference relations as follows: let $v_i(\mathbf{x}^\top_i) = 1$, $v_i(\mathbf{x}^\perp_i) = 0$, and $v_i(\mathbf{x}_i) = p$ iff

$$(\mathbf{x}_i, \mathbf{x}^0_{C_i}) \sim \langle p, (\mathbf{x}^\top_i, \mathbf{x}^0_{C_i}); 1-p, (\mathbf{x}^\perp_i, \mathbf{x}^0_{C_i})\rangle.$$

Such local value functions can be elicited using only *local* queries over attributes in I_i and C_i. Furthermore, we should note that standard gambles are only used to provide a semantic definition of the local value functions. In practical elicitation procedures, we use other types of local queries to elicit such functions (as we describe below); whatever the type of local queries, they involve only the comparison of alternatives, or gambles with the best and worst levels of the attributes, *in a single factor*, under the assumption that the attributes in the conditioning set are fixed at default levels.

Just as in additive case, we need to elicit utilities of a few "key" *full* outcomes to achieve the right calibration of the local value functions. In GAI models, for each factor we must know the utility of the best and the worst possible outcomes under the restriction that the attributes in *other* factors are set to their default levels. Therefore, we need to perform only $2m$ global queries—the same number as in the additive utility model (here, m would be the number of attributes).[2]

Once the utilities of key outcomes are known, the calibration can be done algebraically using an expression derived by Fishburn [6]. In our UAI 2005 paper, to which we refer for details, we introduce a tractable algorithmic procedure to perform the same task by exploiting a graphical structure (expressed by a directed GAI graph, as in Figure 1) of a given GAI model. In this way, we can provide a canonical definition of subutility functions u_i in terms of the local value functions v_i and the utilities of key outcomes.

4 Partial Elicitation with Local Queries

We now describe one possible way of performing *partial elicitation* of utility parameters. Generally speaking, good (or even optimal) decisions can be realized without complete utility information. Rather than asking for the direct assessment of utility parameters using standard gambles as in [7], we use simpler binary *comparison queries* over local gambles. Following [5; 2], we suppose some prior over the parameters of a GAI model, and use myopic expected value of information (EVOI) to determine appropriate queries.

In particular, we assume that uncertainty over utilities is quantified via independent priors over local value function parameters. In such a case, we use queries of the form "Is the local value of suboutcome $\mathbf{x}_i$ greater than l?" where l lies in the normalized range $[0, 1]$, and the attributes in the conditioning set are assumed to be fixed at default levels. (Note that this query is equivalent to a comparison of local lotteries.) Either response *yes* or *no* bounds the local values and reduces uncertainty over utility functions.

Such queries are *local* because they ask a user to focus on preferences over a (usually small) subset of attributes; the values of remaining attributes do not have to be considered. The best myopic query can be computed analytically if the prior information over local utility parameters is specified as a mixture of uniform distributions [2]. Such mixtures are closed under updates, which makes it possible to maintain an exact density over utility parameters throughout the elicitation process. Furthermore, we can compute the best query (including a continuous query point l) analytically. Experimental results on a 26-variable car-rental problem [3] illustrate that the GAI structure of this problem is sufficient to admit fast (around 1 second) EVOI computation; therefore, our approach can readily support interactive real-time preference elicitation. We also show that our myopically-optimal querying strategy allows us recommend good or even optimal decisions with very few queries.

[2] If our default outcome is the worst possible outcome, we only need to perform m global queries.

5 Next Steps

A number of directions remains to be explored. In terms of immediate extensions, methods for eliciting GAI model structure are paramount since a suitable GAI decomposition is a prerequisite for our algorithm. Other directions include incorporating noise models into user responses [2] and developing computationally tractable approximations for computing sequentially optimal querying strategies.

Critical to the success of automated preference elicitation is deeper exploration of the psychological and human-factors issues associated with framing and ordering effects, sensitivity analysis and robustness, and the reliability and acceptability of different modes of interaction. This work helps lay firm semantic foundations from a normative perspective; incorporating the insights of behavioral models is vital.

References

[1] F. Bacchus and A. Grove. Graphical models for preference and utility. *Proc. 11th Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pp.3–10, Montreal, 1995.

[2] C. Boutilier. A POMDP formulation of preference elicitation problems. *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, pp.239–246, Edmonton, 2002.

[3] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Constraint-based optimization with the minimax decision criterion. *9th Intl. Conference on Principles and Practice of Constraint Programming (CP-03)*, pp.168–182, Kinsale, Ireland, 2003.

[4] D. Braziunas and C. Boutilier. Local utility elicitation in GAI models. *Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp.42–49, Edinburgh, 2005.

[5] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pp.363–369, Austin, TX, 2000.

[6] P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Economic Review*, 8:335–342, 1967.

[7] C. Gonzales and P. Perny. GAI networks for utility elicitation. *Proc. 9th Intl. Conference on Principles of Knowledge Representation and Reasoning (KR-04)*, pp.224–234, Whistler, BC, 2004.

[8] H. A. Holloway and C. C. White, III. Question selection for multiattribute decision-aiding. *European Journal of Operational Research*, 148:525–543, 2003.

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