CUI Networks: A Graphical Representation for Conditional Utility Independence*

Yagil Engel and **Michael P. Wellman**
University of Michigan, Computer Science & Engineering
2260 Hayward St, Ann Arbor, MI 48109-2121, USA
{yagil,wellman}@umich.edu

Abstract

We introduce CUI networks, a compact graphical representation of utility functions over multiple attributes. CUI networks model multiattribute utility functions using the well studied and widely applicable utility independence concept. We show how conditional utility independence leads to an effective functional decomposition that can be exhibited graphically, and how local, compact data at the graph nodes can be used to calculate joint utility. We discuss aspects of elicitation and network construction, and contrast our new representation with previous graphical preference modeling.

Introduction

Despite their equal importance to decision making, preferences and utilities have generally not received the level of attention AI researchers have devoted to beliefs and probabilities. Nor have the (increasing) efforts to develop representations and inference methods for utility achieved a degree of success comparable to the impact of graphical models on probabilistic reasoning. The representation of probability distribution functions by Markov or Bayesian networks (Pearl 1988), exploiting conditional independence to achieve compactness and computational efficiency, has led to a plethora of new techniques and applications. Recognizing that utility functions over multidimensional domains may also be amenable to factoring based on independence (Keeney & Raiffa 1976), several have aimed to develop models with analogous benefits (Bacchus & Grove 1995; Boutilier, Bacchus, & Brafman 2001; La Mura & Shoham 1999; Wellman & Doyle 1992). This is our goal as well, and we compare our approach to these and other methods in the Related Work section.

Multiattribute utility

Our utility-theoretic terminology follows the definitive text by Keeney and Raiffa (1976). In the multiattribute utility framework, an *outcome* is represented by a vector of values for n variables, called *attributes*. The decision maker's preferences are represented by a total pre-order, $\succeq$, over the set of probability distributions over outcomes, also called *lotteries*. Given a standard set of axioms, the preference order can be represented by a real-valued *utility function* over outcomes, U, such that numeric ranking of probabilistic outcomes by expected utility respects the ordering by $\succeq$. The utility function is unique up to affine transformations. A positive linear transform of U represents the same preferences, and is thus *strategically equivalent*.

The ability to represent utility over probability distributions by a function over outcomes provides some structure, but in multiattribute settings the outcome space is n-dimensional. Unless n is quite small, therefore, an explicit (e.g., tabular) representation of U will generally not be practical. Much of the research in multiattribute utility theory aims to identify structural patterns that enable more compact representations. In particular, when subsets of attributes respect various independence relationships, the utility function may be decomposed into combinations of modular subutility functions of smaller dimension.

Let $S = \{x_1, \ldots, x_n\}$ be a set of attributes. In the following definitions (and the rest of the work) capital letters denote subsets of attributes, small letters (with or without numeric subscripts) denote specific attributes, and $\overline{X}$ denotes the complement of X with respect to S. We indicate specific attribute assignments with prime signs or superscripts. To represent an instantiation of subsets X, Y at the same time we use a sequence of instantiation symbols, as in $X'Y'$.

Definition 1. Y is *Utility Independent (UI)* of $\overline{Y}$, UI$(Y, \overline{Y})$, when the preference order for lotteries over Y, holding $\overline{Y}$ fixed, do not depend on the particular instantiation chosen for $\overline{Y}$.

Given UI(Y,X), taking $X = \overline{Y}$, the *conditional utility function* over Y given X' is invariant up to affine transformations, for any value X'. This fact can be expressed by the decomposition

$$U(X, Y) = f(X) + g(X)U(X', Y), \quad g(\cdot) > 0.$$

We can in this case refer to $U(X', Y)$ as a function on Y, called a *subutility function* for Y. Y is also called a *separable* subset. Since $U(X', Y)$ is a function only of Y, we sometimes use the notation $U_{X'}(Y)$.

Definition 2. Y is *Conditionally Utility Independent (CUI)* of X given Z ($Z = \overline{XY}$) if preference order for lotteries

*This work was supported in part by NSF grant IIS-0205435, and the STIET program under NSF IGERT grant 0114368. We thank Carmel Domshlak for helpful suggestions.

involving change only in the level of Y, holding X and Z fixed, do not depend on the particular level chosen for X. We denote this relationship by CUI($Y, X \mid Z$).

CUI also supports functional decomposition. For any Z', the conditional utility function over Y given $X'Z'$ is strategically equivalent to this function given a different instantiation of Y. However, the transformation depends not only on X, but also on Z'. Hence we can write:

$$U(X,Y,Z) = f(X,Z) + g(X,Z)U(X',Y,Z), \quad g(\cdot) > 0. \tag{1}$$

That is, we can fix X on some arbitrary level X' and use two transformation functions f, g to get the value of U for other levels of X. Note that here $\{Y, Z\}$ is not considered separable, and therefore $U(X', Y, Z)$ is not considered a subutility function but a conditional utility function. Yet we can use the notation $U_{X'}(Y, Z)$ in this case as well.

A stronger, symmetric form of independence that leads to additive decomposition of the utility function is called *Additive Independence*. We provide the definition for its conditional version.

Definition 3. X and Y are *Conditionally Additive Independent* given Z, CAI($X, Y \mid Z$), if for any instantiation Z', preferences over lotteries on X, Y given Z' depend only on their marginal conditional probability distributions.

This condition leads to the following decomposition:

$$U(X,Y,Z) = f(X,Z) + g(Y,Z).$$

Additive independence and its resulting additive decomposition can be generalized to multiple subsets that are not necessarily disjoint. This condition is called *generalized additive independence* (GAI). If GAI holds, U decomposes to a sum of independent functions f_i over the GAI subsets X_i. Bacchus and Grove (1995) exploit GAI and CAI to construct graphical models representing the additive independence structure.

Graphical models of CUI

In their concluding remarks, Bacchus and Grove (1995) suggest investigating graphical models of other independence concepts, in particular utility independence. The most obvious benefit of a model based on (conditional) utility independence is the generality admitted by a weaker independence condition, in comparison to additive independence. Whereas additivity practically excludes any interaction between utility of one attribute to the value of another, utility independence allows substitutivity and complementarity relationships, as long as the risk attitude towards one variable is not affected by the value of another. One could also argue that UI is particularly intuitive, based as it is on an invariance condition on the preference order. In contrast, (generalized) additive independence requires a judgment about the effects of joint versus marginal probability distributions.

However, there are some major obstacles to achieving such modeling. Foremost, utility independence does not lead to as effective a decomposition as does additive independence. In particular, the condition $UI(Y, X)$ ensures that Y has a subutility function, but since X does not have one it is harder to carry on the decomposition into X. Hence in the case that X is large the dimensionality of the representation may remain too high. Our approach therefore employs CUI conditions on large subsets, in which case the decomposition can be driven further by decomposing the conditional utility function of the larger subset using more CUI conditions.

In the sequel we show how serial application of CUI leads to functional decomposition. Next we show how its graphical model, a CUI network, provides a lower-dimension representation of the utility function in which the function for any vertex depends only on the node and its parents. We then demonstrate the use of CUI networks by constructing an example for a relatively complex domain. Next we elaborate on the technical and semantic properties of the model and knowledge required to construct it, and conclude following a comparative discussion of related work.

CUI networks

CUI decomposition

Suppose that we obtain a set σ of CUI conditions on the variable set $S = (x_1, ..., x_n)$, such that for each $x \in S$, σ contains a condition of the form

$$\text{CUI}(S \setminus (\{x\} \cup P(x)) \ , \ x \mid P(x)).$$

In other words, there exist a set $P(x)$ that separates the rest of the variables from x. According to this condition as applied to x_1 and using (1):

$$U(S) = f_1(x_1, P(x_1)) + g_1(x_1, P(x_1))U_{x_1^0}(S \setminus \{x_1\}). \tag{2}$$

$U_{x_1^0}$ on the right hand side is the conditional utility function of S given that x_1 is fixed at a reference point x_1^0. For convenience we omit the fixed attributes from the list of arguments. In addition, we assume that we have specified a reference point for each variable x, denoted by x^0.

By applying the decomposition based on the CUI condition of x_2 on the conditional utility function $U_{x_1^0}$ we get

$$U_{x_1^0}(S \setminus \{x_1\}) = f_2 + g_2 U_{x_1^0, x_2^0}(S \setminus \{x_1, x_2\}). \tag{3}$$

We omit the list of arguments to the functions f_2, g_2 since the arguments to f_i, g_i are always $(x_i, P(x_i))$.

Substituting $U_{x_1^0}$ in (2) according to (3) yields:

$$\begin{aligned} U(S) &= f_1 + g_1(f_2 + g_2 U_{x_1^0, x_2^0}(S \setminus \{x_1, x_2\})) \\ &= f_1 + f_2 g_1 + g_1 g_2 U_{x_1^0, x_2^0}(S \setminus \{x_1, x_2\}) \end{aligned}$$

Continuing in this fashion we get for x_i:

$$U(S) = \sum_{k=1}^{i} (f_k \prod_{j=1}^{k-1} g_j) + \prod_{j=1}^{i} g_j U_{x_1^0, \ldots, x_i^0}(x_{i+1}, ..., x_n).$$

For convenience, we define the constant function $f_{n+1} \equiv U_{x_1^0, \ldots, x_n^0}()$. Ultimately we obtain

$$U(S) = \sum_{i=1}^{n+1} (f_i \prod_{j=1}^{i-1} g_j). \tag{4}$$

This term is a decomposition of the multiattribute utility function to lower dimensional functions, whose dimensions depend on the number of variables of $P(x)$ required for each CUI condition. Note that using proper bookkeeping any utility value can be recovered from the f, g representation in time $O(n)$.

This decomposition can be represented graphically as follows. First define an order on the set S (for convenience we assume the ordering $x_1, ..., x_n$). Next define the set of parents of variable x_i,

$$Pa(x_i) = P(x_i) \setminus Dn(x_i),$$

where $Dn(x_i)$ is the set of attributes in $x_1, ..., x_{i-1}$ that turned out to be descendants of x_i (in other words, those for which x_i is a parent or another descendant of x_i is a parent). This procedure defines a DAG. When associating each node x_i with the data (f_i, g_i), the utility function can be obtained from the graph using (4).

The dimensionality of the f, g functions can now be reduced further. Since f and g correspond to well-defined CUI conditions using predefined reference points, we can apply (4) to calculate utility in any variable ordering. However, if we restrict the utility calculation to agree with the reverse topological order of the graph, the descendants of x_i would always be fixed when the CUI condition of x_i is applied, and thus f_i, g_i need not depend on $Dn(x_i)$ only on $Pa(x_i)$. Formally, let $y_1, ..., y_k$ be the variables in $Dn(x_i)$. Define:

$$\begin{aligned} f_i(x_i, Pa(x_i)) &= f_i(x_i, Pa(x), y_1^0, ..., y_k^0) \\ g_i(x_i, Pa(x_i)) &= g_i(x_i, Pa(x_i), y_1^0, ..., y_k^0) \end{aligned}$$

In words, we redefine f_i, g_i for each variable x_i, as the f_i, g_i resulting from the decomposition according to (2), in which the variables in $Dn(x_i)$ are fixed according to their reference points. As a result, the dimensionality of the representation is reduced (as in Bayesian networks) to the maximal number of parents of a vertex plus one.

An example

Given are sets $S = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and

$$\begin{aligned} \sigma = \{&\text{CUI}(x_4x_5x_6, x_1 \mid x_2x_3), \text{CUI}(x_4x_3x_6, x_2 \mid x_1x_5), \\ &\text{CUI}(x_2x_4x_6, x_3 \mid x_1x_5), \text{CUI}(x_1x_3x_5, x_4 \mid x_2x_6), \\ &\text{CUI}(x_6, x_5 \mid x_1x_2x_3x_4), \text{CUI}(x_1x_2x_3x_5, x_6 \mid x_4)\}. \end{aligned}$$

Construction of the network using the order implied by the indices results in the graph illustrated in Figure 1. The minimal separating set for x_1 is $\{x_2, x_3\}$. For x_2 the only non-descendant variable that is required to separate it from the rest is x_5, and that would be its only parent. The rest of the graph is constructed in a similar way.

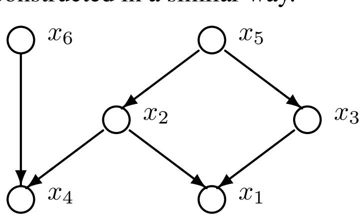

Figure 1: CUI network given σ and order x_1,…,x_6.

Next we illustrate how the utility function can be decomposed using any variable ordering that agrees with the reverse topological order of the graph. We apply the ordering x_4, x_1, x_6, x_3, x_2, x_5. To simplify notation we denote the conditional utility function in which x_i is fixed on the reference point by U_i.

$$\begin{aligned} &U(S) = f_4(x_4x_2x_6) + g_4(x_4x_2x_6)U_4(S \setminus \{x_4\}) \\ &U_4(S \setminus \{x_4\}) = f_1(x_1x_2x_3) + g_1(x_1x_2x_3)U_{1,4}(S \setminus \{x_4x_1\}) \\ &U_{1,4}(S \setminus \{x_4x_1\}) = f_6(x_6) + g_6(x_6)U_{1,4,6}(x_2x_3x_5) \\ &U_{1,4,6}(x_2x_3x_5) = f_3(x_3x_5) + g_3(x_3x_5)U_{1,3,4,6}(x_2x_5) \\ &U_{1,3,4,6}(x_2x_5) = f_2(x_2x_5) + g_2(x_2x_5)U_{1,2,3,4,6}(x_5) \\ &U_{1,2,3,4,6}(x_5) = f_5(x_5) + g_5(x_5)U_{1,2,3,4,5,6}() \end{aligned}$$

Merging the above equations, and using the definition $f_7 \equiv U_{1,2,3,4,5,6}()$ produces

$$\begin{aligned} U(S) = f_4 + g_4f_1 + g_4g_1f_6 + g_4g_1g_6f_3 + g_4g_1g_6g_3f_2 + \\ g_4g_1g_6g_3g_2f_5 + g_4g_1g_6g_3g_2g_5f_7. \end{aligned}$$

A choice of a different variable ordering would yield a different sum of products. For example if we choose x_1, x_3, x_4, x_2, x_5, x_6 we would get:

$$\begin{aligned} U(S) = f_1 + g_1f_3 + g_1g_3f_4 + g_1g_3g_4f_2 + g_1g_3g_4g_2f_5 + \\ g_1g_3g_4g_2g_5f_6 + g_1g_3g_4g_2g_5g_6f_7 \end{aligned}$$

In each case we are using the same CUI decompositions and therefore the same f, g functions.

Properties of CUI networks

Based on the construction method described above, we conclude the following.[1]

Proposition 1. *Let S be a set of attributes, and σ a set of CUI conditions on S. If σ includes a condition of the form CUI($S \setminus (x \cup Z_x), x \mid Z_x$) for each $x \in S$ then σ can be represented by a CUI network whose maximal dimensionality does not exceed* $\max_x(|Z_x| + 1)$.

Note that Z_x denotes here a minimal set of attributes (variables) that renders the rest CUI of x. This bound on the dimensionality will be obtained regardless of the variable ordering. We can expect the maximal dimension to be lower if the network is constructed using a good variable ordering. A good heuristic in determining the ordering would be to use attributes with smaller dependent sets first, so that the attributes with more dependents would have some of them as descendants. Based on such an ordering we would expect the less important attributes to be lower in the topology, while the more crucial attributes either be present higher or have a larger number of parents.

The basic property of CUI networks is a direct result of its construction method.

Proposition 2. *Let G be a CUI network for a set of attributes S. Then for any $x \in S$,*

$$\textit{CUI}(S \setminus (x \cup Pa(x) \cup Dn(x))\ ,\ x \mid Pa(x) \cup Dn(x))$$

[1]Proofs of all propositions are omitted due to space limits.

An example application

To demonstrate the potential representational advantage of CUI networks we require a domain that is difficult to simplify otherwise. The example we use is the choice of a software package by an enterprise that wishes to automate its sourcing (strategic procurement) process. We focus on the software's facilities for running auction or RFQ (request for quote) events, and tools to select winning suppliers either manually or automatically.

We identified nine key features of these kinds of software packages. In our choice scenario, the buyer evaluates each package on these nine features, graded on a discrete scale (e.g., one to five). Note that the choice of scale for the attributes does not affect the independencies among them. The features are, in brief:

Interactive Negotiations (IN) allows a separate bargaining procedure with each supplier.

Multi-Stage (MS) allows a procurement event to be comprised of separate stages of different types.

Cost formula (CF) allows the buyers to formulate their total cost of doing business with each supplier.

Supplier Tracking (ST) allows long-term tracking of supplier performance.

MultiAttribute (MA) allows bidding over multiattribute items, potentially using a scoring function.[2]

Event Monitoring (EM) provides an interface to running events and real-time graphical views.

Bundle Bidding (BB) bidding for bundles of goods.

Grid Bidding (GB) adds another bidding dimension by applying an additional aspect such as time or region.

Decision Support (DS) tools for optimization and for aiding in the choice of the best supplier(s).

We observe first that additive independence does not widely apply in this domain. For example, the multi-stage feature makes several other features more useful or important: interactive negotiations (often useful as a last stage) decision support (to choose multiple bidders to move to next stage) and event monitoring (helps keep track of how useful was each stage in reducing costs). Conversely, in some circumstances multi-stage can substitute for the functionality of other features: multiattribute (by bidding on different attributes in different stages), bundle bidding (bidding on separate items in different stages), grid bidding (bidding on different time/regions in different stages) and supplier tracking (by extracting supplier information in a "Request for Information" stage). We list all the potential dependencies for each attribute in Table 1.

When a complement or substitute relation is identified it excludes additive independence. From this fact we can identify a set of six attributes that must be mutually (additive) dependent: $\{BB, GB, DS, MA, MS, CF\}$. In consequence, the best-case dimensionality achieved by a CAI network (Bacchus & Grove 1995) for these attributes would be six (the size of the largest maximal clique, see related work).

[2] We hope the fact that the software itself may include facilities for multiattribute decision making does not cause undue confusion. Naturally, we consider this an important feature.

Var	Complements	Substitutes	CUI set
EM	CF ST MS		IN,DS,MA,GB,BB
IN	ST MS MA		EM,CF,ST,DS,GB,BB
CF	EM MS DS MA GB BB	DS	MA,GB,BB
ST	EM MS IN DS	MA	IN,CF,GB,BB
MA	IN DS CF	MS BB ST GB	GB,BB
MS	DS EM IN ST	GB BB MA CF	MA,GB,BB
DS	CF MA GB ST BB MS		IN,EM
GB	CF DS	MA MS BB	MA,BB
BB	CF DS	MA MS GB	MA,GB

Table 1: Dependent and independent sets for each attribute.

In order to constrcut a CUI network we first identify, for each attribute x, a set Y that is CUI of it. A possible approach is to first guess such a set according to the information in Table 1, then detect and verify potentially larger CUI sets. Keeney and Raiffa (1976) provide several useful results that can help in detection of UI, and those results can be generalized to CUI. In particular we can first detect a *conditional preferential independence (CPI)* condition (which is the restriction of CUI to certain outcomes) in which one element is also CUI. For example in order to verify

$$\text{CUI}(\{BB, GB, MA\}, CF \mid S \setminus \{BB, GB, MA, CF\}),$$

the following two conditions are sufficient:

$$\text{CPI}(\{BB, GB, MA\}, CF \mid S \setminus \{BB, GB, MA, CF\}), \quad (5)$$

$$\text{CUI}(BB, \{GB, MA, CF\} \mid S \setminus \{BB, GB, MA, CF\}). \quad (6)$$

The detection and verification of these conditions are also discussed by Keeney and Raiffa (1976). The intuition in our example is as follows: we observe that the features BB, GB, and MA each add a dimension to the bidding. That bidding dimension is best exploited when cost formulation is available, so each of the three complements CF, with a similar impact. This implies (5). Moreover, BB is a crucial feature and therefore the risk attitude towards it is not expected to change given a change in the level of CF, MA, and GB, and that implies (6).

In a similar fashion, we observe that the nature of the substitutivity of the three mechanisms BB, GB, MA in MS is similar: each can be simulated using multiple stages. Moreover, the dependency among the triplet is also a result of the option to substitute one by another (each of the three can be used as an alternative mechanism to the other). As a result $\{BB, GB, MA\}$ is CUI of MS (given the rest of the attributes), and each pair of the three is CUI of the third. Finally we find that the complementarity of ST and IN is marginal and does not affect the tradeoffs with other attributes. We verify that $\{ST, EM, CF, DS, GB, BB\}$ is CUI of IN, and that $\{GB, BB, CF, IN\}$ is CUI of ST. The resulting maximal CUI sets for each attribute are shown in Table 1.

Using the variable ordering EM, IN, CF, ST, MA, MS, DS, GB, BB we build a CUI network, for each defining the set of parents to be the variables that are not in the CUI set and are not descendants (Figure 2). The resulting maximal dimension is four.

The structure we obtained over the utility function in the above example is based on objective domain knowledge, and may be common to various sourcing departments.

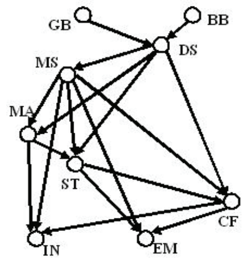

Figure 2: CUI network for the example. The maximal number of parents is 3, leading to dimension 4.

This demonstrates an important aspect of graphical modeling that is captured by CUI networks, which is the opportunity to graphically encode qualitative information about the domain, thus making the process of extracting the numeric information easier. This structure in some cases differs between specific decision makers, but in other cases (as above) it makes sense to extract such data from domain experts and reuse this structure between decision makers.

Representation and elicitation

Node data representation

Representing U by a CUI network requires that we determine the f and g functions for each CUI condition. At any node y the functions f, g represent the affine transformation of the conditional utility function $U(x^0, Y, Z)$ (here $Z = Pa(x)$) to strategically equivalent utility functions for other values of x. Like the transformation functions for UI (Keeney & Raiffa 1976), the transformation functions for CUI can be represented in terms of the conditional utility functions $U(x, Y^1, Z)$ and $U(x, Y^2, Z)$ for arbitrary values Y^1, Y^2. This can be done by solving the system of two equations below, both based on applying (1) for these specific values of Y:

$$U(x, Y^1, Z) = f(x, Z) + g(x, Z)U(x^0, Y^1, Z),$$
$$U(x, Y^2, Z) = f(x, Z) + g(x, Z)U(x^0, Y^2, Z),$$

yielding

$$g(x, Z) = \frac{U(x, Y^2, Z) - U(x, Y^1, Z)}{U(x^0, Y^2, Z) - U(x^0, Y^1, Z)}, \quad (7)$$

$$f(x, Z) = U(x, Y^1, Z) - g(x, Z)U(x^0, Y^1, Z). \quad (8)$$

The only restriction on the choice of Y^1, Y^2 is that the decision maker must not be indifferent between them given x^0 and the current assignment to Z. For example, Y^1, Y^2 may differ on any single attribute $y \in Y$ that is *strictly essential*.

Elicitation of measurable value functions

For particular applications we can point out specific attributes that can be used as a measurement for others. The most common example is preferences that are quasi-linear in a special attribute such as money. These kind of preferences can be represented by *measurable value function (MVF)*. An MVF is a cardinal utility function defined under certainty and represents *preference differences*. It has been shown (Dyer & Sarin 1979) that UI has an analogous interpretation for MVF with similar resulting decomposition. The extension to CUI is straightforward.

In the monetary example the preference difference over a pair of outcomes represents the difference in the *willingness to pay (wtp)* for each. A potential way to elicit the MVF is by asking the decision maker to provide her wtp to improve between one outcome to another, preferably when these outcomes differ over a single attribute.

Under this interpretation, we first observe that $g(x, Z)$ can be elicited as preference differences, between outcomes that possibly differ over a single attribute. Moreover, it encompasses qualitative preferential information: assume $Y^2 \succeq Y^1$ and that $x^0 \preceq x \ \forall x$. Then $g(x, Z)$ is the ratio of the preference difference between Y^1 and Y^2 given x to the same difference given x^0 (Z is fixed in all outcomes). Hence, if Y and x are complements then $g(x, Z) > 1$ and increasing in x. If Y and x are substitutes $g(x, Z) < 1$ and decreasing in x. This holds regardless of the choice for Y^1, Y^2, since by $CUI(Y, x \mid Z)$ all attributes in Y maintain the same complementarity or substitutivity relationship to x. Note also that $g(x, Z) = 1$ iff $CAI(Y, x \mid Z)$. Another important observation is that though both Y and x may depend on Z, in practice we do not expect the *level of correlation* between Y and x to depend on the particular value of Z. In that case g becomes a single dimensional function, independent of Z.

$f(x, Z)$, intuitively speaking, is a measurement of *wtp* to improve from x^0 to x. The value $U(x^0, Y^1, Z)$ is multiplied by $g(x, Z)$ to compensate for the interaction between Y and x, allowing $f(x, Z)$ to be independent of Y. If we perform the elicitation obeying the topological order of the graph, the function $U(x^0, Y^1, Z)$ can be readily calculated for each new node from data stored at its predecessors. Choose $Y^1 = Y^0$, and and let $Z = \{z_1, ..., z_k\}$, ordered such that children precede parents. Since Y, x are fixed on the reference point,

$$U(x^0, Y^0, Z) = \sum_{i=1}^{k} (f_{z_i} \prod_{j=1}^{i-1} g_{z_j}) f_{n+1}().$$

Now we can obtain $f(x, Z)$ as follows: first we elicit the preference difference function $e(x, Z) = U(x, Y^1, Z) - U(x^0, Y^1, Z)$. Then, assuming $g(x, Z)$ was already obtained, calculate:

$$f(x, Z) = e(x, Z) - (g(x, Z) - 1)U(x^0, Y^1, Z).$$

Nested representation

From (7) and (8) we conclude that node data can be represented by conditional utility functions depending on the node and its parents. But this may not be the best dimensionality that can be achieved by this network. Perhaps the set $Z = Pa(x)$ has some internal structure, and the subgraph induced by Z has maximal dimension lower than $|Z|$. In such a case we could apply the corresponding CUI decomposition on the conditional utility functions above. For example, in Figure 1 the function $U(x_1, x_4^1, x_5^1, x_6^1, x_2, x_3)$

is needed to represent node x_1. However the network shows that CUI($x_3, x_2 \mid x_1, x_4, x_5, x_6$). Hence:

$$U(x_1, x_4^1, x_5^1, x_6^1, x_2, x_3) = f'(x_1, x_2) + g'(x_1, x_2)U(x_1, x_3, x_2^0, x_4^1, x_5^1, x_6^1).$$

We use the notation f' and g' since these are not the same functions of the top level decomposition.

This can be done systematically by performing a complete CUI decomposition over the subgraph induced by Z whenever Z is not a clique, keeping in mind that all the resulting functions also depend on x.

Proposition 3. *Let G be a CUI network for utility function $U(S)$. Then $U(S)$ can be represented by a set of conditional utility functions, each depending on a set of attributes corresponding to (undirected) cliques in G.*

This result reduces the maximal dimensionality of the representation to the size of the largest maximal clique of the CUI network. An important implication is that we can somewhat relax the requirement to find very large CUI sets. If some variables end up with a large number of parents, we can reduce the dimensionality using this technique.

The proof (omitted) is constructive. The procedure may generate a complex functional form, decomposing a function multiple times before the factors become restricted to a clique. The ultimate number of factors required to represent $U(S)$ is exponential in the number of such nesting levels. However, it is difficult to construct examples for which more than just a few nesting levels are needed, and often in those cases a better variable ordering can help.

Related Work

Bacchus and Grove (1995) introduced graphical preference modeling through conditional additive independence. In the resulting networks any two sets of variables whose respective graph nodes can be separated are CAI given their complement. The utility function is decomposed to a sum over functions on subsets of variables corresponding to the maximal cliques of the graph. Since CAI implies symmetric CUI we can construct a CUI representation from a CAI network and achieve at least as low and potentially lower dimensionality (since CUI is a weaker condition), as we showed in the example application.

Boutilier et al. (2001) introduce *UCP networks*, a graphical model for cardinal utility which utilize the GAI decomposition combined with a CP-net topology (Boutilier *et al.* 1999) that requires dominance relations between parents and their children. Our main distinction from UCP is in avoiding additivity, and in particular the GAI decomposition which is harder to detect and verify. The GAI structure was also applied for graphical models by Gonzales and Perny (2004).

An earlier work by La Mura and Shoham (1999) redefines utility independence as a symmetric multiplicative condition, taking it closer to its probability analog. This way it supports a Bayes-net like representation, however using a stronger and less well-studied independence condition.

The only graphical decomposition suggested in the past for utility functions that is based on the original, asymmetric notion of utility independence is the utility tree (Von Stengel 1988) (see also Wellman & Doyle (1992) for discussion in an AI context). The utility tree is constructed by a hierarchical decomposition of attribute subsets to multilinear or multiplicative decomposition (Keeney & Raiffa 1976), ultimately decomposing U to functions of the smallest separable subsets and their complements. When an effective multilinear decomposition exists, utility trees may achieve better dimensionality than CUI nets (however bearing the price of an exponential number of constants to maintain). In general we expect CUI nets to achieve lower dimensionality since they do not require unconditional UI.

Conclusions

We have presented a graphical representation for multiattribute utility functions, based on conditional utility independence. CUI is a weaker independence condition than those previously employed as a basis for graphical utility representations. We proposed techniques to obtain and verify structural information, and use it to construct the network and elicit the numeric data. CUI networks provide a potentially compact representation of the multiattribute utility function, via functional decomposition to lower-dimensional functions that depend on a node and its parents.

References

Bacchus, F., and Grove, A. 1995. Graphical models for preference and utility. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, 3–10.

Boutilier, C.; Bacchus, F.; and Brafman, R. I. 2001. UCP-networks: A directed graphical representation of conditional utilities. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, 56–64.

Boutilier, C.; Brafman, R. I.; Hoos, H. H.; and Poole, D. 1999. Reasoning with conditional ceteris paribus preference statements. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 71–80.

Dyer, J. S., and Sarin, R. K. 1979. Measurable multiattribute value functions. *Operations Research* 27:810–822.

Gonzales, C., and Perny, P. 2004. GAI networks for utility elicitation. In *Ninth Intl. Conf. on the Principles of Knowledge Representation and Reasoning*, 224–234.

Keeney, R. L., and Raiffa, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley.

La Mura, P., and Shoham, Y. 1999. Expected utility networks. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 366–373.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

Von Stengel, B. 1988. Decomposition of multiattribute expected utility functions. *Annals of Operations Research* 16:161–184.

Wellman, M. P., and Doyle, J. 1992. Modular utility representation for decision-theoretic planning. In *First Intl. Conf. on Artificial Intelligence Planning Systems*, 236–242.