Artificial Intelligence 175 (2011) 1122–1152

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

The local geometry of multiattribute tradeoff preferences

Michael McGeachie [a,*], Jon Doyle [b]

[a] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA

ARTICLE INFO

Article history:
Received 2 March 2009
Received in revised form 10 August 2010
Accepted 10 August 2010
Available online 2 December 2010

Keywords:
Decision theory
Preference representation
Multiattribute tradeoffs
Ceteris paribus reasoning

ABSTRACT

Existing representations for multiattribute *ceteris paribus* preference statements have provided useful treatments and clear semantics for qualitative comparisons, but have not provided similarly clear representations or semantics for comparisons involving quantitative tradeoffs. We use directional derivatives and other concepts from elementary differential geometry to interpret conditional multiattribute *ceteris paribus* preference comparisons that state bounds on quantitative tradeoff ratios. This semantics extends the familiar economic notion of marginal rate of substitution to multiple continuous or discrete attributes. The same geometric concepts also provide means for interpreting statements about the relative importance of different attributes.

1. Building value functions from preferences and tradeoffs

Knowledge of someone's preferences can be used to make decisions on their behalf. Following the work of von Neumann and Morgenstern [35], direct and complete elicitation of preferences and their representation in the form of utility functions has enabled decision analysts to advise decision makers on how to decide specific questions. To go beyond manual construction of specific decision models, and to automate decision analysis in a way that applies in a broad range of mundane and fleeting human activities, one must find richer representations that permit making decisions with imprecise, incomplete, and accumulating information about preferences.

We pursue this end by presenting semantics for several different types of preference statements that build on earlier semantics for *ceteris paribus* preferences (preference other things being equal) [38,14]. We focus on quantitative tradeoff statements, such as "having a CPU speed of 3 GHz is at least twice as important as having 4 GB memory and a 250 GB disk in my new computer purchase." Such statements say that some outcomes that satisfy one condition (CPU speed of 3 GHz) are preferred to some outcomes that satisfy another condition (4 GB memory and 250 GB disk), and also bound below how much better the former are than the latter. We provide semantics for numerous types of statements of this character, including multiattribute tradeoffs that relate more than one attribute at a time; tradeoffs over discrete or continuous domains; conditional or unconditional tradeoffs; and quantitative or purely qualitative comparisons. We also treat related types of statements about attribute importance, such as "increasing CPU speed is at least twice as important as increasing memory and disk size in my new computer purchase." Such statements say that the weight given to some attribute or attributes in a decision should be greater than that given to other attributes.

Computing expected utility of actions requires a numerical utility or value function that represents preferences in the sense that the numerical representation assigns a greater value to one outcome than to a second outcome if the preference statements entail that the first outcome is preferred to the second. Building on earlier constructions [32], we accompany

* Corresponding author.
E-mail addresses: mmcgeach@csail.mit.edu (M. McGeachie), Jon_Doyle@ncsu.edu (J. Doyle).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2010.11.014

the semantics for *ceteris paribus* preferences statements with companion algorithms for compiling utility or value functions from collections of multiattribute tradeoff statements.

1.1. Decision-analytic methodology

Our methodology seeks to extend traditional decision analysis by relaxing assumptions and restrictions on the form and character of the preference information captured in the preference acquisition process. In particular, we aim for representations of preference information that permit automation of the process of constructing decision models starting at earlier points in the process than has been possible with traditional modeling methods.

In traditional decision analysis methodology [33,25], a human decision analyst does considerable work in understanding and analyzing a decision informally before the point at which the tools of traditional decision theory [35,34,19] are brought to bear. The decision analyst first interviews the decision maker about what dimensions or attributes of the decision are of consequence. The decision analyst then assesses utility functions on each of these dimensions by standard gambles or other means. This assessment requires the decision maker to think carefully about the upper and lower bounds of each dimension, to consider his or her attitude toward risky hypothetical choices, and to determine which attributes are utility independent of other attributes. The relative importance of each dimension must then be assessed, at which point the decision analyst can combine the results into a multiattribute utility function that models the preferences of the decision maker.

This traditional methodology has the virtue of producing considered and complete decision models appropriate to the decision at hand. It also, however, demands much effort on the part of the decision maker by requiring careful attention to complexities of the decision that might not have been considered previously, and that perhaps should not be answered immediately with only the information currently on hand. All this makes the interviewing and analysis steps lengthy and time-consuming in many cases, so that one mainly applies decision analysis in detail to repetitive decisions in which the cost of analysis can be amortized over many individual decisions, and to one-off decisions of great import, such as governmental policy or complicated life-or-death medical decisions.

We seek to begin the process of formalization earlier than with traditional decision analytic techniques. The traditional formal techniques apply once the analyst has done much of the work needed to identify the dimensions along which preferences vary. Our preference semantics allows one to formalize partial information about preferences. Such information may be stated and captured naturally without any requirement that the stated preferences involve independent or fundamental attributes, and without explicit indications of utility independence or preferential independence. In our view, such independence relations properly reflect conclusions reached during the analysis of some decision, as inferences from the whole body of stated conditions on preferences, rather than presuppositions underlying the entire analysis. We thus address the identification of dependencies and independencies among attributes in our numerical-compilation methods, which perform analyses that yield model-structuring conclusions akin to those reached by a human analyst at the point at which the human analyst begins quantitative assessment procedures. Our approach thus supports protracted incremental deliberation prior to the introduction of traditional formal decision analysis, and helps automate the initial steps previously relegated to informal reasoning that produce the formal framing of a problem.

1.2. Illustration

To illustrate these ideas, we describe a fictitious scenario in which Mike, a human, informs an automated personal shopping agent of his preferences so that it can watch for online deals on computer hardware he may find attractive. Mike will buy a new laptop if there is a good deal on one he likes. Mike does not try to tell the agent all about his preferences at the start, as without detailed knowledge of what is currently available he might not yet have developed definite preferences regarding the options. Mike instead gives his agent information about his preferences bit by bit as he learns more about what preferences are germane.

His agent retrieves a list of laptops for sale at various vendors' websites. Seeing the list, Mike decides that, with respect to price and speed, a \$1500, 3 GHz machine is preferable to a \$2000, 3.4 GHz machine, other things being equal. This preference sets up a tradeoff between price and speed, so the agent then filters out the results that are very expensive even though they are somewhat faster than average. Thinking about it a little more, Mike decides that the first machine is much better than the other one, in fact that it is at least five times better.

Looking at some of the expensive options with many attractive features, Mike then realizes that adding features and adding ounces of weight at the same time is not what he wants. Mike tells the agent that *Weight* is more important than *Price*. The agent readjusts its evaluation of options, and shows more laptops ordered by weight, with several attractive light-weight options at the top of the list.

Mike sees that there are some good machines available that are light, moderately powerful, and within his price range, but realizes that he must decide how big a screen and what resolution he needs to do his work on the road, since this adversely impacts the price and the weight. Mike decides a $12''$ screen on a 4.5 pound machine for \$1700 is better than a $14''$ screen on a 6 pound machine for \$1800. This suffices to order the remaining options in a way that satisfies Mike, and ends up earmarking a machine for later consideration and probable purchase.

Our intent in this paper is to provide meanings for the sorts of statements Mike makes and rationales for the conclusions Mike draws. We do not treat the hard problems of preference elicitation, of coaxing Mike to reveal his preferences through

introspection, interview, or observation. We instead focus on how to make use of the information that Mike does express, which differs from that required by traditional decision analysis.

For example, Mike never explicitly considers whether some attribute or other is fundamental or independent of other attributes. Indeed, at some point he might discover that computer manufacturers offer differing amounts of memory depending on which operating system is installed. Mike also does not think too carefully about the relative weights that might be assigned to different dimensions in a hypothetical value function. He knows that one product is much more attractive than the other, and makes guesses as to how much more attractive, guesses that he might later revise to produce a different set of options and final decision, as in [13].

1.3. Semantical approach

To accompany the methodology of incremental interpretation and revision of preference information, we interpret tradeoff preference statements as constraints on the shape of a utility or value function, and in particular, as constraints on the partial derivatives in different regions of the space of outcomes.

For example, a decision maker could state that "Increased longevity is five times as important as decreased price," a sentiment which one might express in the context of repainting one's home, and this is interpreted as indicating that value increases in the increasing lifetime direction five times faster than value increases in the decreasing price direction.

The tradeoff interpretation presented in this very simple example is one treated in standard economics as the notion of *marginal rate of substitution* between the two commodities longevity and price. The partial derivative of the utility function u over commodity bundles with respect to an attribute α, is known as the *marginal utility* of α, and represents the marginal impact of this commodity on utility. To compare the impact on utility due to changes in commodity α relative to commodity β, one divides the marginal utility of α by the marginal utility of β. This ratio is invariant under monotone increasing transforms of u, and is variously called the *rate of commodity substitution* of β for α and the *marginal rate of substitution* of β for α, and expresses the amounts of α and β that can be exchanged without changing desirability [24]. Thus in the formal marginal rate conception, one would restate "Increased longevity is five times as important as decreased price" as "Increasing longevity by one unit of longevity would be as good to me as decreasing price by five units of price." To stipulate that v makes the marginal rate of substitution of β for α at least r everywhere, we can choose v so as to satisfy the constraint that

$$\frac{\partial v}{\partial \alpha}(\vec{a}) \Big/ \frac{\partial v}{\partial \beta}(\vec{a}) \geqslant r \tag{1}$$

at each outcome $\vec{a}$.

In practice, however, one cannot always rely on tradeoffs to concern only two decision attributes. For example, a person building a computer might state a preference for *AMD and Overclocked* over *Intel and Cheaper*, other things being equal. To handle such preferences, our semantics extends the approach taken in inequality (1) to complex multivariate, discrete, and qualitative comparisons by using the more general notion of directional derivative.

A directional derivative $\mathrm{D}_{\vec{x}}(v)$ of a function $v : \mathbb{R}^n \to \mathbb{R}$ evaluated at a point $\vec{a} \in \mathbb{R}^n$ is the derivative along a vector with base $\vec{a}$ and direction $\vec{x}$. Furthermore, the directional derivative of a function v in the direction of vector $\vec{x}$ is equal in value to the inner product $\nabla v \cdot \vec{x}$ of the gradient of the function with $\vec{x}$. This quantity measures the increase of v in the direction of $\vec{x}$. Thus if we want to talk about constraints on the directional derivatives of the value function, or rates of increase in the directions of $\vec{x}$ and $\vec{y}$, we can state constraints of the form

$$\nabla u(\vec{a}) \cdot \vec{x} \geqslant r \nabla u(\vec{a}) \cdot \vec{y}. \tag{2}$$

In the following, we show how such constraints can, in turn, be represented as inequalities among directional derivatives of value functions over continuous or discrete attributes.

1.4. Plan of the paper

Section 2 summarizes past approaches to representation of preference information, along with traditional independence concepts used in analyzing the structure of preferences, including the notion of generalized additive independence used in our construction algorithms. Section 3 presents background concepts and definitions that will be useful throughout, namely the familiar concepts of preference orders, value functions, and what it means for a value function to represent a preference order. This section also introduces a language for partial descriptions of outcomes called "bundles" (of goods) together with useful operations on bundles and descriptions of bundles in vector terms. Section 4 introduces an extension of first-order logic for expressing specifications of preferences and tradeoffs in a way that fits the methodology of incremental specification and interpretation. The subsequent sections present the semantics of the preference and tradeoff extensions.

Sections 5 through 7 constitute the semantical core of the paper. Section 5 provides meanings for a range of types of preference and tradeoff statements including tradeoffs between concrete alternatives described partially, over many or few attributes (Section 5.2), and tradeoffs over continuous and discrete attributes (Section 5.4). Section 6 presents a semantics for qualitative *ceteris paribus* preferences that adapts the semantics of [14] to the new setting. Section 7 applies the same techniques to interpret specifications of relative importance judgments of attributes, over many or few attributes.

Section 8 presents one method for constructing value functions from the preference statements analyzed in earlier sections. This method updates one developed in earlier work [32] both to support the enhanced representation of qualitative preferences and to reflect the addition of tradeoff preferences. Section 9 shows how our tradeoff preferences can be combined with the popular CP-net representation for conditional qualitative *ceteris paribus* preferences. We provide straightforward methods for enhancing a CP-net with quantitative tradeoffs. Section 10 concludes with a review of our contributions and a discussion of some promising avenues for future work.

2. Related work

Our objective is to lessen the difficulty of decision analysis by allowing the human decision maker to express preferences in forms he or she finds natural. Our work builds on a large and varied literature attempting to make the elicitation, specification, use, and applicability of utility functions faster, simpler, easier, and broader. Some that literature concerns operational methods for elicitation of preferences, particularly in the case in which fundamental attributes for characterizing the decision have already been identified and one must formulate questions about the relative desirability of different combinations of attribute values. Other parts of the literature concern languages for expressing preferences. Our main purpose in the present work is to define a specification language for qualitative and quantitative *ceteris paribus* preferences and quantitative tradeoffs that allows direct expression of preferences and their dependence on other portions of knowledge, along with a method of compiling a utility function consistent with the set of those statements. This section briefly illustrates how our work combines ideas and harnesses benefits of prior work on preference representations.

In pursuing an approach that seeks to understand the structure of attributes that underlie preferences from expressions of preferences about these, we follow on substantial work in the fields of multiattribute decision theory and conjoint measurement theory. Multiattribute decision theory [25] takes utility function representations of the preferences of a decision-maker and studies mathematical methods for decomposing these functions into weighted combinations of subfunctions corresponding to different attributes or sets of attributes. Theorems of multiattribute decision theory allow inferences about several forms of independence of attributes from the subfunctions that make up an overall utility theorem (see, for example, [22,36,39]). We draw directly on the underlying framework of multiattribute decision theory in the following development, and focus on using differential properties to specify the utility functions being modeled. Conjoint measurement theory [27] takes orders over combinations of attributes as fundamental, and studies when one can use these to identify separable orders over attributes. Conjoint measurement theory formulates families of axioms relating orders over individual attributes to orders over combinations of attributes, with various conceptions of "cancellation" corresponding to different forms of lifting of orders, along with statistical methods for determining when observed data fits these axioms. We draw some connections to the theory of conjoint measurement in the following, but more work would be useful here, as our underlying methodology exhibits something of the same spirit in seeking to take facts about preferences among different sets of attributes, in our case assorted subsets of attributes rather than all combinations of all attributes as in conjoint measurement theory, and then relate these to preferences over individual attributes and to possible complete orders over all attribute combinations.

2.1. Logics of preference

Researchers have long considered logics of preference, whereby an entity can specify statements in some logical language encoding preference information. Logical statements of the form $p \succ q$, meaning formula p is preferred to formula q, can be given varying interpretations. One of those interpretations, termed preference *ceteris paribus*, allows preferences that apply "keeping other things equal." Such a preference might be "Other things being equal, I prefer white wine to red wine." These preferences capture the intuitive idea that other unmentioned qualities might affect the decision making process. For example, in a situation where white wine turns out to be much more expensive than red wine, and hence all else is not equal, this preference would not apply.

Ceteris paribus preferences were introduced formally by von Wright [37], and later studied in semantic and inferential terms by Hansson [23] and Doyle, Shoham, and Wellman [14]. This work describes a set of qualitative preferences that can be used to partially order outcomes described by logical variables. Other researchers have added complexities to basic *ceteris paribus* statements, including the possibility of expressing preferential independence between attributes using a specialized syntax [2].

A simple list of *ceteris paribus* preferences allows reasoning with partial preference information, complex qualitative preferences, and include information about preferential independence of various attributes or dimensions in the domain.

2.2. Conditional preference networks (CP-nets)

Also using a form of *ceteris paribus* preferences, come a series of graphical preference networks using different types of preference or utility independence conditions to define the network structure. These start with the CP-net [6,5] (here the abbreviation "CP-net" stands for "conditional preference" networks, not "*ceteris paribus*" networks). Nodes in the graph correspond to single attributes and the directed edges in the graph represent preferential dependence between attributes. In this system, one lists the preference for a particular attribute given the values of the attribute's parents in the graph, and

may then determine the resulting preferences over elements from the entire domain. The CP-net representation combines these explicit preferences with explicit preferential independence and dependence, achieving efficient reasoning in many cases. Subsequent work builds upon this base in different ways, adding representations of more complex preferences.

Brafman, Domshlak and Shimony [9] considered CP-nets with tradeoffs, or TCP-nets, an extension of CP-nets that allows expression of trade-offs between attributes. TCP-nets allow one to stipulate that one preference is *more important* than another preference, conditioned on the value of some other attributes. This allows a TCP-net to represent preferences between outcomes that vary on two attributes, while including attribute prioritization in the representation.

Wilson [40] uses a circumscription-like semantics for augmenting conditional *ceteris paribus* preference statements in CP-nets. This representation addresses approximate reasoning, stating preferences where all else is held *almost* equal, by allowing the specification of some irrelevancies. The expression $v : x \succ x'[W]$ means that given value v for attribute V, and a list of attributes W, we prefer value x to x' for attribute X, as long as all unmentioned attributes are held equal, while attributes W are allowed to be equal or unequal.

To take the CP-net idea and develop it in a different direction, Boutilier, Bacchus, and Brafman [4] introduce a quantitative extension to CP-nets called UCP-nets. A UCP-net allows a combination of utility independence information with *ceteris paribus* semantics and explicit numeric utility values assigned to each preference. As [4] observe, such an approach is warranted in cases where uncertainty is an issue, probabilities are known, and reasoning in terms of expected utility is required.

Conditional preference networks are conceptually similar to Bayesian networks in many ways. In a more direct development from Bayes nets, La Mura and Shoham have proposed combining Bayesian networks with a type of utility independence networks into what they call Expected Utility Networks (EUNs) [28]. This representation differs from CP-nets by using a type of multiplicative preference independence, rather than additive independence, which allows preference reasoning analogous to probability reasoning and results in a representation of expected utility. In [28] preference judgments are stated in multiples of the value of an arbitrary reference outcome, *i.e.* "being healthy and wealthy is three times as desirable as being merely wealthy."

In another development with roots in both Bayesian networks and preference networks, Gonzales and Perny [20,21] discuss *generalized additive independence* (GAI) networks. Generalized additive independence [19,1] is a model that decomposes a utility function into functions of independent, but overlapping, sets of attributes. Given some domain, these are structurally similar to the clique-trees used in inference algorithms of Bayesian networks, and similar message-passing algorithms are used in utility computations in GAI nets. The GAI framework, in general, allows somewhat more complicated utility models than simple additive independence allows.

Networks with other utility independence assumptions and conditions have been considered. Engel and Wellman [17] define CUI-nets for conditional utility independence networks, which have a different decomposition than CP-nets, leading to smaller representations when additive independence does not apply. This is a treatment of quantitative utility functions where each node potentially has a utility function based on the parents of that node, elicited using standard methods from multiattribute utility theory. This system handles both complementary attributes and substitution between attributes.

2.3. Other preference machinery

Other avenues of preference research have resulted from adding something like utility weights to propositional logic formulae and then reasoning about the relative utility of outcomes that obtain. For example, systems based on possibilistic logic [15] attach a "priority" to formulae in propositional logic, where priorities form a totally-ordered set. In this type of approach one infers a preference for a condition p over a condition q from the priorities assigned to p and q, with the formula given the higher priority preferred to the formula with the lesser priority. These systems are less related to the current project than systems employing explicit preference representation and tacit preferential independence assumptions. They do however share our intention to reason with partial preference information by encoding an arbitrary set of user statements and then inferring preference orders from that set. More related is the work of Lang et al. [29] in which propositions are graded with numerical "utility" values, and in which utilities assigned to situations reflects the sum of positive and negative rewards of each satisfied goal proposition. This allows comparison of situations in which multiple goals are satisfied. Adding weights together creates preferences over outcomes or choices of actions, allowing an implicit utility function to be incrementally and partially specified. In very simple cases, one might regard the ratio of utilities assigned to goals as corresponding to marginal rates of substitution. Although an exact interpretation along these lines would be complicated by the provision of the model to specify cardinal gaps in utility values for goals and the nonmonotonic provisions of the semantics, the overall result of constructing preference orders from this additive basis exhibits a similar spirit to the constructions we make from different semantical bases.

These approaches both base preference specification on comparisons to an assumed standard of comparison or "numéraire" in the language of economics. While there is nothing problematic about the notion of numéraire in standard market economics, that is the case because one can choose any market good as the standard of comparison, whether it is an ounce of gold, a dollar, or a bushel of wheat. In contrast, starting out with a given standard of comparison, especially intangible ones like units of currency, presupposes many unstated comparisons of that standard with things familiar to the modeler. Our approach attempts to avoid this problem by making it possible to state direct comparisons between sets of attributes, from which one might construct a numéraire as in market theory, from which one might identify preferential

or utility dependencies between attributes, and with which one can express overriding and context-dependent preferences. One can always still use some agreed numéraire as an attribute in phrasing comparisons, one need not coerce all comparisons into that restricted form. Eliciting preferences with respect to currency values is a common technique, but in light of the problems people have putting dollar (or mark, or franc, or yen) values on some things, empirical work would seem needed to determine whether people would always find it easier than stating direct comparisons.

Part of our goal in the current work is to compile a set of preference judgments, naturally stated, into a utility function for efficient reasoning, and one example of such a system is the proposal of Domshlak and Joachims [12]. Therein, they compute a value function from a few preference statements (of the form $x \succ y$, where x, y are formula over a space of n binary attributes) using a support vector machine (SVM). The SVM kernel implicitly translates the n attributes into a space of size 4^n, where there is one new attribute for each attribute in the input space and for each interaction between attributes in the input space. This translation both trivializes preferential independence concerns and provides tolerance to some slight errors or inconsistencies in the input preferences, while enabling reasoning with partial preference information.

2.4. Going forward

Our work builds on a base of qualitative *ceteris paribus* preferences [38,14], and augments this with reasoning about quantitative and qualitative preference and tradeoff information. As such, our preference machinery in this paper moves in a slightly different direction than CP-nets. Like the collections of logical statements augmented with preference information, we are able to reason with any number of statements; unlike CP-nets we do not require complete elicitation of preferences for each variable or according to each independence condition. Like TCP-nets, we add tradeoff and importance statements to *ceteris paribus* preferences, but we choose to add quantitative statements of importance and tradeoffs. Like UCP-nets, we endeavor to allow efficient reasoning about quantified preferences. The UCP-net has been combined with TCP-nets in recent work [7], which provide a qualitative tradeoff semantics to UCP-nets. We will show that our quantitative tradeoffs can also be combined with UCP-nets, in Section 9.

Our work also builds upon elements of the other preference systems mentioned in the preceding. Expected Utility Networks base their semantics on multiplicative judgments of value; we combine a semantics of multiplicative tradeoffs with qualitative logical preferences. Like the SVM and possibilistic logic preference compilation techniques, we take as input a list of statements of preference, in some natural logical form, and seek to compile them into an explicit utility function that enables more efficient preference reasoning. In addition to providing support for tradeoffs and importance judgments, our system may be applicable in some domains where the cardinality of some attributes makes the SVM's dimensional translation impractical, or where explicit tradeoffs provide handy shortcuts for weighted possibilistic logic statements.

In sum, our work combines elements of many existing lines of preference reasoning and representational research. The result, as presented in this article, is a formalism of qualitative preference *ceteris paribus* that combines quantified tradeoffs and importance judgments while allowing partial reasoning and assuming no particular independence forms with a compilation procedure that translates the forgoing into an explicit utility function for efficient reasoning. Such a system should be a valuable contribution to the preference research community.

3. Formal background

In this section we present the formal concepts and notation that we use throughout for outcomes, attributes, orders, representations, and independence properties of orders.

3.1. Outcomes, attributes, and partial outcomes

For the present treatment we follow common practice and identify decision outcomes with tuples of values of a set of attributes. Formally, let $A = \{A_i \mid 1 \leqslant i \leqslant n\}$ be a finite, enumerated set of attributes, and each attribute's domain be a set denoted D_i. Attribute domains can be finite or infinite, discrete or continuous. A set of outcomes is described by $\vec{A} = D_1 \times \cdots \times D_n$, a cartesian attribute space.

To simplify the presentation and discussion in the remainder, we will assume throughout that the domain of each attribute is numeric. In particular, we assume the use of a one-to-one function $\rho : \vec{A} \to \mathbb{R}^n$ that gives a numeric representation of the entire attribute space, including numeric representations of nonnumeric attribute domains. We make no assumptions regarding whether the domains of each attribute are continuous intervals or not, as this will not be of central importance in the following.

In choosing the numeric representations of attributes, we assume that the representations *conform* to the character of the underlying attribute. Numeric representations of *nominal* attributes can be chosen arbitrarily. For *ordinal* attributes, the underlying attribute domain has a natural order, and we assume that the numeric representations are chosen to have a numeric order consistent with that of the natural order on the represented nonnumeric values. Similarly, for *interval* attributes, we assume that the numeric representation preserves distances between underlying attribute values, and for *ratio* attributes, we assume that the representation preserves ratios as well.

To interpret preference statements that refer to some attributes but not to others requires means for talking about partial outcomes defined over the specific attributes mentioned by the preference statements. In fact, we employ three different

ways of talking about partial outcomes: as partial assignments of values to attributes, as vectors over attribute values extended with a "0" (meaning "unassigned") value, and as vectors over subsets of attributes.

We use *bundles* as descriptions of outcomes in terms of partial assignments of values to attributes. A bundle (of goods) is a partial function from $\vec{A}$ to $\coprod_i D_i$, the disjoint union of domains of each attribute. For a bundle b, $b(i)$ is the value assigned by b to attribute i. $b(i)$ is either undefined, in which case we write $b(i) = \perp$, or $b(i)$ is a value $w \in D_i$. If a bundle defines one value for every attribute $A_i \in A$ then we say it is *complete*. We can also write a bundle as a list of the pairs it defines: $b = \{(A_i = w_i), (A_j = w_j), \ldots\}$ where $w_i \in D_i$, $w_j \in D_j$. We call the set of attributes to which a bundle b assigns values the *support* of b, and denote it by $\sigma(b)$.

We define operations on bundles by defining component-wise operations using standard algebra, but with additional support for $\perp$. We have $\perp + \perp = \perp$, $\perp * \perp = \perp$, $\perp * 0 = 0$, and for any real $r \neq 0$, $\perp + r = \perp$ and $\perp * r = \perp$. Otherwise, bundle addition is much like vector addition, as is multiplication of a scalar and a bundle. One multiplies two bundles component-wise: for bundles b, b', we have $b * b' = b''$ where $b''(i) = b(i) * b'(i)$. Replacement of a bundle by values from another is written $b[b'] = b''$ and defined as follows: $b''(i) = b'(i)$ unless $b'(i) = \perp$, in which case $b''(i) = b(i)$. We also write $b[(i = w)]$ for the replacement of b with an implicit bundle $b' = \{(i = w)\}$.

For each bundle b, we define a corresponding vector $\phi(b) \in \mathbb{R}^n$, which we call the *value vector* for b. Writing $\phi_i(b)$ for the ith component of $\phi(b)$, we define $\phi(b)$ by $\phi_i(b) = b(i)$ whenever $b(i) \neq \perp$ and $\phi_i(b) = 0$ otherwise. Because value vectorization maps both 0 and $\perp$ to 0 elements in the vector, one cannot recover the original bundle from a vector unless 0 is not in D_i, but this degeneracy will not matter in the following, where we vectorize bundles before computing their inner product with other vectors, and the additional 0s make the development much simpler. Value vectorization allows us to use operations on vectors in place of operations on bundles. For example, we use the inner product $\phi(b) \cdot \phi(b')$ of value vectors to compute the inner product of bundles b and b'. We call a vector $\vec{x} \in \mathbb{R}^n$ the *characteristic vector for* a set of attributes $G \subseteq A$ iff $x_i = 1$ iff $A_i \in G$ and $x_i = 0$ otherwise.

For each subset $G \subseteq A$ of attributes, we define the outcomes $\vec{G}$ over G to be the cartesian product ΠG, taking the product in the enumeration order inherited from A. If $\vec{x} \in \vec{A}$, we define the projection function $\pi_G : \vec{A} \to \vec{G}$ so that $[\pi_G(\vec{x})]_a = x_a$ for each $a \in G$. If $\vec{g} \in \vec{G}$, we define the bundle $b(\vec{g})$ corresponding to $\vec{g}$ so that the bundle is undefined on every attribute not in G, and takes the same value as $\vec{g}$ on every attribute in G.

We write vectors next to each other when we refer to their combination; if $\vec{X}$ and $\vec{Y}$ are disjoint sets of attributes and $\vec{x} \in \vec{X}$ and $\vec{y} \in \vec{Y}$, then $\vec{x}\vec{y}$ refers to the vector over $\vec{X} \cup \vec{Y}$ that orders the combined values of $\vec{x}$ and $\vec{y}$ according to the attribute enumeration order.

We call $\mathcal{C} = \{C_1, \ldots, C_k\}$ a *cover* of the attributes A iff each $C_i \subseteq A$ and $\bigcup_i C_i = A$. Distinct attribute subsets in a cover need not be disjoint.

3.2. Preference orders and value functions

We model the preferences of a decision maker as an ordering over the outcomes described by $\vec{A}$. A *weak preference* ordering is a reflexive and transitive relation $\succsim$ on $\vec{A}$ where $\vec{a} \succsim \vec{a}'$ indicates that $\vec{a}$ is at least as preferable as $\vec{a}'$. We do not assume or require that $\succsim$ forms a total order. *Strict preference* $\succ$ consists of the irreflexive part of $\succsim$, that is $\vec{a} \succ \vec{a}'$ just in case $\vec{a} \succsim \vec{a}'$ but $\vec{a}' \not\succsim \vec{a}'$. When $\vec{a} \succsim \vec{a}'$ and $\vec{a}' \succsim \vec{a}$ we say $\vec{a}$ and $\vec{a}'$ are *indifferent* and write $\vec{a} \sim \vec{a}'$.

Economics and decision theory use the terms "utility function" and "value function" to name numerical representations of preference orders. We will use the terms interchangeably in this paper, but favor the term "value function." Although none of the following treats decision-making under uncertainty, for which the usual term is "utility," our intent here is definitely to provide a language and semantics for characterizing the structure of utility functions.

A value function, $v : \vec{A} \to \mathbb{R}$, allows the use of $\geqslant$ as the standard order (and therefore preorder) over the reals, and thus over the image of $\vec{A}$ under v. We write $\geqslant_v$ for the preorder on $\vec{A}$ induced by v. Complete preorders $\succsim$ over countable $\vec{A}$ can be expressed exactly by value functions, so that $v(\vec{a}) \geqslant v(\vec{a}')$ if and only if $\vec{a} \succsim \vec{a}'$. We say a value function v *represents* a complete preference order $\succsim$ when $v(\vec{a}) \geqslant v(\vec{a}')$ if and only if $\vec{a} \succsim \vec{a}'$. An incomplete preorder $\succsim$ is necessarily a subset of some preorder $\geqslant_v$. When $\succsim$ is a subset of the preorder $\geqslant_v$, we say that v *is consistent with* $\succsim$.

We call functions $\hat{v}_G : \vec{G} \to \mathbb{R}$ *partial value functions*; these assign a number to partial descriptions of an outcome. We define *subvalue functions* over G to be value functions $v_G : \vec{A} \to \mathbb{R}$ such that $v_G(\vec{a}) = \hat{v}_G(\pi_G(\vec{a}))$. Subvalue functions over G ignore all but some set of attributes, G. As a matter of notational convenience, we frequently use bundles as arguments to value functions, and would write $v(\phi(b))$ for the operation of v on the characteristic vector of a bundle b. When the context is clear, we suppress the ϕ function and just write $v(b)$.

3.3. Preferential independence

Preferential independence is a property that obtains when the contribution to value of some attributes can be determined without knowledge of other attributes. More precisely, a set of attributes X is preferentially independent of a disjoint set of attributes Y when the comparisons over attributes in X do not depend on the assignment of values to attributes in Y. We state this formally in the following definition.

Definition 3.1 *(Preferential independence).* A set of attributes $\vec{X}$ is *preferentially independent* of a disjoint set of attributes $\vec{Y}$ with $A = \vec{X} \cup \vec{Y}$, if and only if, for all $\vec{x}_1, \vec{x}_2 \in \vec{X}$, and $\vec{y}_1, \vec{y}_2 \in \vec{Y}$,

$$\vec{x}_1\vec{y}_1 \succ \vec{x}_2\vec{y}_1 \rightarrow \vec{x}_1\vec{y}_2 \succ \vec{x}_2\vec{y}_2. \tag{3}$$

Note that preferential dependence is not symmetric. It is possible for a set of attributes $\vec{X}$ to be preferentially dependent upon a disjoint set of attributes $\vec{Y}$, while $\vec{Y}$ is independent of $\vec{X}$. However, the case of symmetric preferential independence is the same as satisfying the "single cancellation axiom" of conjoint measurement theory [27].

Preferential independence generally simplifies the structure of the corresponding value functions. The simplest preference structures occur when every subset of attributes is preferentially independent of every disjoint subset, which produces a fully additive value function and a condition called *additive independence*. An *additive value function* over two attributes (or sets of attributes) is obtained when the marginal rates of substitution do not depend on the particular value [25, p. 91]. For $D = \{G, H\}$, with G, H disjoint, an additive value function has the form

$$v(\vec{a}) = g_i v_G(\vec{a}) + h_i v_H(\vec{a}). \tag{4}$$

A more general case is that seen in a *generalized additive value function* for a cover $\mathcal{C} = \{C_1, \ldots, C_k\}$ of A is a value function $v : \vec{A} \rightarrow \mathbb{R}$ formed as a weighted sum of k subvalue functions $v_i = v_{C_i} : \vec{A} \rightarrow \mathbb{R}$ [19,1,10,7], that is,

$$v(\vec{a}) = \sum_{i=1}^{k} t_i v_i(\vec{a}). \tag{5}$$

The function-construction methods presented in Section 8 assume that attributes are preferentially independent whenever there is no evidence to the contrary, and then construct value functions with generalized additive forms over subvalue functions representing subsets of attributes determined to be mutually dependent on the basis of preference statements relating them.

4. A language for preference specification

As Section 2 indicated, there are statements of preference that cannot be formally stated in existing preference reasoning systems. Preferences regarding numerical tradeoffs cannot be combined with qualitative statements of direct preference, *ceteris paribus* preference, incompletely-specified preferences, and ambivalence toward preferential independence. The traditional means for combining the import of disparate types of statements is to embed the statements in a single logical language and provide interpretations that merge the sense of the various statements. Accordingly, in this section we introduce a language and logic for preference specification called LOPAT, for Logic of Preferences and Tradeoffs that provides such a combined representation. LOPAT uses a first-order logical language to express attributes and conditions, and adds to this first-order base one or more sets of preference and tradeoff relations, with each such set representing a preference order and associated utility function over outcomes.

4.1. Base language and logic

The base of LOPAT consists of a first-order Logic of Attributes and Comparisons (LAC) obtained by choosing finite sets of relation symbols $R_1, \ldots, R_l$, function symbols $F_1, \ldots, F_m$, and a finite or infinite set of individual constant symbols $C_1, \ldots$, in which the constants and function symbols are used to name decision attributes. For example, a language describing computer-purchasing preferences might include function symbols like *speed*, *CPU*, and *GHz*, allowing reference to the numerical measurement in gigaHertz of the CPU speed of a computer X with the term $GHz(speed(CPU(X)))$. Naturally, constants amount to zero-ary functions, but we distinguish them from nonzero arity functions. We require that the relation symbols of LAC include the symbols $=, <, \leqslant, >, \geqslant$ representing familiar equality and inequality binary relations. We assume that the constants of LAC include names for any numbers in $\mathbb{R}$ needed to state conditions and values.

As usual, an interpretation $\mathcal{I} = (D, R_1^{\mathcal{I}}, \ldots, R_l^{\mathcal{I}}, F_1^{\mathcal{I}}, \ldots, F_m^{\mathcal{I}}, C_1^{\mathcal{I}}, \ldots)$ of LAC consists of an underlying domain D together with interpretations of each n-ary relation as a subset of D^n, of each n-ary function as a function from D^n to D, and of each constant as an element of D.

We define satisfaction and entailment in LAC in the usual way, so that $\mathcal{I} \models q$ just in case the meanings assigned by $\mathcal{I}$ make q true. We write $p \models q$ just in case every interpretation making p true also makes q true, and for a theory (set of sentences) S write $S \models q$ just in case $p \models q$ for each $p \in S$. We write $[\![p]\!]$ and $[\![S]\!]$ to denote the set of all models of p and S, that is, $[\![p]\!] = \{\mathcal{I} \mid \mathcal{I} \models p\}$.

4.1.1. Logical and decision-making attributes

We regard each ground term of LAC as representing a potential decision-making attribute. For example, a theory describing computer-purchasing preferences might involve terms such as $speed(CPU(X))$ and $GHz(speed(CPU(X)))$, the former representing the CPU speed of a computer X, and the latter representing the numerical measurement of that speed in gigaHertz.

In most practical situations, one restricts attention to a finite number of these attributes, denoted here by $A_1, \ldots, A_n$ to match the finite set of attributes presented in Section 3 to characterize our underlying treatment of decision making. We make no assumptions that the restricted set of attributes must be given at the beginning of decision analysis, or that it remains unchanged throughout the course of developing a decision model. We assume only that at any point in the analytic process the set of attributes of interest is finite. More general treatments of outcomes described in terms of infinitely many attributes might be useful, we do not develop such here.

With attention focused on a set of attributes $A_1, \ldots, A_n$, one can regard each interpretation $\mathcal{I}$ of Lopat as inducing the tuple of values $(A_1^{\mathcal{I}}, \ldots, A_n^{\mathcal{I}})$, which we also denote as $(A_1, \ldots, A_n)^{\mathcal{I}}$. If, as in Section 3, we associate a particular domain of values with each of these attributes, we can call an interpretation $\mathcal{I}$ *conforming* (to the assumed domains) if it interprets all attribute terms as taking values in the corresponding attribute value domains. We will assume all interpretations are conforming throughout the remainder of this paper.

In the case of conforming interpretations, we can define the *attribute meaning* $[\![p]\!]_{\mathrm{a}}$ of a statement p to be the set of value tuples possible in interpretations consistent with p, that is, $[\![p]\!]_{\mathrm{a}} = \{(A_1, \ldots, A_n)^{\mathcal{I}} \mid \mathcal{I} \in [\![p]\!]\} = \{\vec{x} \in \vec{A} \mid \vec{x} = (A_1, \ldots, A_n)^{\mathcal{I}} \wedge \mathcal{I} \models p\}$.

We do not assume that using a ground term as a decision attribute says anything about the logical or preferential dependence or independence of that attribute on other attributes or on other terms not chosen as decision attributes. Part of the work of decision analysis is to identify such dependencies and independencies. We also leave open questions about how best to treat such restrictions of attention. In some settings one might translate a theory into a Datalog-like sublanguage of Lac, that is, a sublanguage that omits all functions except for a finite set of individual constants, each of which represents one outcome attribute.

We assume that the set of constants of Lac also includes names for any elements of the attribute domains D_i needed to express preferences and tradeoffs.

4.1.2. Attribute types and comparisons

We can use the intended orderings of attribute domains to divide the set of decision attributes into *nominal* and *ordinal* attributes. Nominal attributes, such as colors and names, bear no nontrivial inequality relations among their values, so that $\alpha < \beta$ is always false, and $\alpha \geqslant \beta$ is true only if $\alpha = \beta$ is true. Ordinal attributes can carry nontrivial strict total or partial orderings over their values, orderings distinct from any preferential orderings of these values.

We have assumed a numerical encoding of all attributes for simplicity of presentation, but numerical encodings always admit order comparisons, no matter what type of attributes they represent. Our language does not forbid use of order relations in value propositions about nominal attributes, but sensible uses of the language will avoid such as making meaningless comparisons. We therefore extend the notion of conformance from interpretations to interpreted theories by requiring that a *conforming* preference theory states no order comparisons between purely nominal attributes. We assume all theories discussed in the following are conforming.

The focus of our use of Lac expressions in Lopat is in stating preferences and tradeoffs. We thus focus our attention on a sublanguage of Lac in which each statement describes bundles of attribute values.

First, we define *atomic value propositions* to be inequalities or negations of inequalities relating attribute terms to each other or to named domain values, and define *compound value propositions* to be statements formed as Boolean combinations of atomic value propositions. We use the term *value proposition* to refer to both atomic and compound value propositions.

Second, we define a *positive value proposition* to consist of a simple equality statement $(\alpha = \beta)$ relating an attribute α with a value β. Such statements are of course also atomic value propositions. A *bundle proposition* consists of a conjunction of positive value propositions in which no two conjuncts involve the same attribute. One can regard these as logical statements of bundles as defined earlier. In particular, assuming that all the attribute values have names in Lac, a bundle $b = \{(i = w_i), (j = w_j), \ldots\}$ corresponds to the proposition p_b expressed as $A_i = w_i \wedge A_j = w_j \wedge \ldots$. Finally, a proposition in *disjunctive normal form* (DNF) consists of a disjunction of conjunctions of atomic value propositions, that is, a disjunction of bundle propositions.

In addition to ordinary logical interpretations of value propositions, we also regard bundles as partial interpretations, and define the bundle meaning of value propositions accordingly.

For each bundle b, we say that b satisfies a positive value proposition $\alpha = w$, and write $b \models \alpha = w$, just in case p_b assigns the value w to attribute α. We write $b \not\models \alpha = w$ if $b \models \alpha = w$ does not hold. It follows that $b \not\models \alpha = w$ just in case either b assigns some other value to α or b does not assign a value to α. We define bundle satisfaction of the other forms of atomic value propositions similarly.

For complex value propositions p and q, we define $b \models (p \wedge q)$ to hold just in case $b \models p$ and $b \models q$, and define $b \models \neg p$ to hold just in case $b \not\models p$. We define the meanings of the other Boolean connectives similarly.

4.2. Order and utility expressions

We obtain the full language Lopat by extending Lac with a set of preference and tradeoff relations $\{\succsim_{\mathrm{cp}}, \succ_{\mathrm{cp}}, \succsim_{\mathrm{mt}}, \succ_{\mathrm{mt}}, \succsim_{\mathrm{ai}}, \succ_{\mathrm{ai}}\}$, together with a restriction connective $\Rightarrow$, and a ratio connective :. We use these new linguistic elements and statements c, p, and q of Lac to form three classes of Lopat statements as follows.

First, LOPAT includes expressions for traditional *ceteris paribus* preference statements, which we refer to here as *qualitative ceteris paribus preference statements*, similar to those presented in [14]. We restrict such statements to comparing conditions describable as value propositions p and q, for which we form the expression $p \succsim_{\text{cp}} q$, meaning that condition q is weakly preferred to condition p *ceteris paribus*, and $p \succ_{\text{cp}} q$, meaning that condition q is strictly preferred to condition p *ceteris paribus*. For example, one might write $((\alpha_1 = 3) \wedge (\alpha_2 = 1)) \succ_{\text{cp}} ((\alpha_1 = 1) \wedge (\alpha_2 = 2))$.

We form *restricted* or *conditional qualitative preference statements* by conditioning a qualitative preference statement on a value proposition that restricts the domain of quantification implicit in the *ceteris paribus* comparison. If c is a value proposition, we form such statements as $c \Rightarrow p \succ_{\text{cp}} q$ and $c \Rightarrow p \succsim_{\text{cp}} q$, each of which means that the indicated preference holds *ceteris paribus* among outcomes satisfying the condition c. For example, we can make statements $((\alpha_1 > 3) \wedge (\alpha_1 < 5)) \Rightarrow ((\alpha_2 = 3) \succsim_{\text{cp}} (\alpha_2 = 1) \wedge (\alpha_3 = 0))$.

Note that the preference conditional $\Rightarrow$ has no relation to propositional implication $\rightarrow$. It instead restricts the set of outcomes over which a preference comparison applies. Moreover, the explicit preference conditionalization indicated in $c \Rightarrow p \succ_{\text{cp}} q$ is also different from the implicit conditionalization indicated in $pc \succ_{\text{cp}} qc$, because the attributes mentioned in c play a role in the interpretation of the latter expression, but play no role in the interpretation of the former. (See Theorem 27 of [14]; $c \Rightarrow p \succ_{\text{cp}} q$ when $pc \succ_{\text{cp}} qc$ and c has no attributes in common with p or q, but that $c \Rightarrow p \succ_{\text{cp}} q$ can hold even though $pc \succ_{\text{cp}} qc$ is false.)

Second, LOPAT includes expressions that allow one to say by how much one prefers some improvement in one condition to some improvement in another condition. *Marginal tradeoff statements* take the form $r : p \succ_{\text{mt}} s : q$ or $r : p \succsim_{\text{mt}} s : q$ for value propositions p and q and numbers $r, s \in \mathbb{R}$ such that $r, s \geqslant 1$. We interpret these statements such that the pair of tradeoff factors r, s produce a meaning equivalent to the pair of tradeoff factors $1, \frac{s}{r}$. We allow omission of either or both of the numerical factors when they take the value 1. For example one could write $2 : (\alpha_2 = 3) \succ_{\text{mt}} 5 : ((\alpha_1 = 1) \wedge (\alpha_3 = 0))$, or equivalently, write $1 : (\alpha_2 = 3) \succ_{\text{mt}} 2.5 : ((\alpha_1 = 1) \wedge (\alpha_3 = 0))$. The semantics given later interprets $p \succsim_{\text{mt}} r : q$ as meaning, roughly, that increases in p are at least r times as desirable as increases in q, *ceteris paribus*. *Conditional marginal tradeoff statements*, naturally, condition a marginal tradeoff statement with a value proposition, as in $c \Rightarrow r : p \succ_{\text{mt}} s : q$. For example, we can make the statement $((\alpha_1 > 3) \wedge (\alpha_1 < 5)) \Rightarrow (\alpha_2 = 3) \succsim_{\text{mt}} 3 : ((\alpha_2 = 1) \wedge (\alpha_3 = 0))$.

Third, LOPAT includes statements that allow one to say that one set of attributes is more important than another. To do this, we extend the language to include concrete sets of attribute names, as in $\{\alpha_{i_1}, \ldots, \alpha_{i_k}\}$. More general languages might include set terms and quantification over them, but we do not do so here. There is no substitution of equals for equals in such expressions; it is the attribute names that matter, not their values. With concrete sets G and H and numeric tradeoff parameters $r, s \geqslant 1$, we form *attribute tradeoff statements* $r : G \succ_{\text{ai}} s : H$ and $r : G \succsim_{\text{ai}} s : H$. As with marginal tradeoff statements, we can simplify such statements to ones using a single tradeoff factor, for instance, $G \succ_{\text{ai}} \frac{s}{r} : H$, meaning, roughly, that the attributes in G are at least $\frac{s}{r}$ times as important as the attributes in H. We also form *conditional attribute tradeoff statements* by conditioning an attribute tradeoff statement to hold only for outcomes satisfying a value proposition, as in $c \Rightarrow r : G \succ_{\text{ai}} s : H$.

4.3. Obtaining utility functions as meanings

An interpretation $\mathcal{I} = (D, R_1^{\mathcal{I}}, \ldots, R_l^{\mathcal{I}}, F_1^{\mathcal{I}}, \ldots, F_m^{\mathcal{I}}, C_1^{\mathcal{I}}, \ldots, v)$ of LOPAT, accordingly, extends an interpretation of LAC with a utility function.

Each preference or tradeoff sentence in LOPAT expresses a condition or constraint on a value function, either directly by constraining the value function or its partial derivatives, or indirectly by constraining a preference order represented by the value function. We present the semantics for the weak preference, tradeoff, and importance statements in Sections 6–7. We interpret the strict versions in the usual way, so that

- $r : p \succ_{\text{cp}} s : q$ holds iff $r : p \succsim_{\text{cp}} s : q$ and $r : p \not\succsim_{\text{cp}} s : q$;
- $r : p \succ_{\text{mt}} s : q$ holds iff $r : p \succsim_{\text{mt}} s : q$ and $r : p \not\succsim_{\text{mt}} s : q$; and
- $r : p \succ_{\text{ai}} s : q$ holds iff $r : p \succsim_{\text{ai}} s : q$ and $r : p \not\succsim_{\text{ai}} s : q$.

In particular, because we assume conforming interpretations, we can define the *value meaning* $[\![S]\!]_{\text{v}}$ of a statement or set of statements S to be the set of value functions appearing in some interpretation of S, that is, $[\![S]\!]_{\text{v}} = \{v \mid \mathcal{I} \in [\![S]\!] \wedge \mathcal{I} = (\ldots, v)\}$. When v is a value function over $\vec{A}$, we write $v \models S$ to mean that $v \in [\![S]\!]_{\text{v}}$.

In general, then, combining a theory S phrased in LOPAT with a set of attribute terms appearing within S determines two things; a set of possible outcomes $[\![S]\!]_{\text{a}}$, and a set of possible value functions $[\![S]\!]_{\text{v}}$. We mostly ignore the restrictions on possible outcomes in this paper, and focus instead on the interpretation of the theory in terms of value functions. The value function construction method of Section 8 constitutes one way of computing a particular value function $v \in [\![S]\!]_{\text{v}}$ when given a consistent set of sentences S.

Although LOPAT offers forms of preference specifications intended to extend the range of compactly expressible utility functions, we do not yet have a characterization of just what functions are expressible or inexpressible in LOPAT. For preference orders over finite domains, one can of course construct a finite set of axioms that specify each pairwise comparison directly, and for preference orders over countable domains, one can do the same with a countable set of specification axioms. Such axiom sets can be interpreted as specifying any utility function consistent with the preference order. The real

question, however, is what utility functions over infinite domains can be specified usefully with finite sets of LOPAT axioms or axiom schemata, and the answer to this question is unknown at present.

Note that LOPAT theories do not explicitly include symbols to represent outcomes or utility functions over outcomes. The interpretations of the logic make sense no matter how one identifies decision-making attributes among the terms of the language. The ability to refer implicitly to utility functions comes directly from the inability to express utility values directly in the language. The only statements LOPAT allows about utility functions are *ceteris paribus* orderings over partial derivatives. These can be made without reference to the utility function itself or to its domain. In effect, the language presupposes that the utility function formally depends on *all* terms of the language, leaving it up to the analyst to determine on which terms it actually depends.

5. Marginal tradeoff preferences

As indicated in Section 1, we interpret marginal tradeoff statements $p \succsim_{\text{mt}} r : q$ as constraints on the directional derivatives of the value function. This section shows how to do this for a range of complex statements of conditions by associating one or more directions with each of p and q. We first dispose of what one might consider as a plausible alternative interpretation.

5.1. Avoiding a value-based tradeoff semantics

The simplest possible interpretation of a marginal tradeoff comparison between bundles x and y, $x \succsim_{\text{mt}} r : y$, is that utilities of outcomes in one class are at least r times greater than utilities of outcomes in another class, that $\phi(x)$ is preferred to $\phi(y)$ by a factor of more than r, or formally, for bundles $x, y,$ and $\vec{a}, \vec{a}' \in \vec{A}$

$$v\big(\vec{a}[x]\big) \geqslant r v\big(\vec{a}'[y]\big). \tag{6}$$

We choose not to pursue this interpretation because it does not allow additive independence (Eq. (4)) between attributes.

Theorem 5.1. *If x and y are bundles with attributes $G = \sigma(x) \cup \sigma(y)$ additively independent of the attributes $\overline{G}$, and $v(\vec{a}[x]) \geqslant r v(\vec{a}'[y])$ for all $\vec{a}, \vec{a}' \in \vec{A}$, then $r = 1$.*

Proof. Since G is additively independent of $\overline{G}$, there exists a value function involving the subvalue function $v_G(\vec{a}[x])$ for attributes in G. Then (6) must hold when k is a constant representing the value contributed by the attributes outside of G, which is the case when the assignment to attributes in $\overline{G}$ is fixed, and results in

$$v_G\big(\vec{a}[x]\big) + k \geqslant r\big(v_G\big(\vec{a}'[y]\big) + k\big). \tag{7}$$

However (7) simplifies to

$$v_G\big(\vec{a}[x]\big) \geqslant r v_G\big(\vec{a}'[y]\big) + (r-1)k.$$

This inequality can only hold independent of k when $r = 1$. □

Theorem 5.1 shows that a too-simplistic value-based tradeoff semantics would not be compatible with the simplest of independence conditions. While we will give much attention to generalized additive independence, which Theorem 5.1 does not rule out, in later sections we will be concerned to infer or assume independence conditions within a domain, and so do not wish to preclude ourselves from using a simple additive value function. We therefore conclude that to speak about quantified tradeoffs with nontrivial tradeoff ratios, we cannot use the simple value-based tradeoff semantics represented by (6).

5.2. Geometric tradeoff semantics

Our semantics for marginal tradeoff statements begins by interpreting the special case in which the tradeoff conditions represent individual bundles, and defines meanings for more complex conditions by reducing statements over complex conditions to sets of statements over bundle conditions.

To interpret nontrivial marginal tradeoff ratios, we follow the approach of preference *ceteris paribus* and interpret tradeoffs as comparisons holding other attributes constant. Specifically, we regard such tradeoff statements as constraining the partial derivatives of the utility function, as these partial derivatives explicitly hold constant attributes other than the ones being differentiated. Constraints on derivatives stating that value increases in one direction r times faster than in another direction constrain the shape of the utility function rather than its values, and this includes reference to the units of the attributes.

We interpret marginal bundle tradeoffs through the following definition.

Definition 5.1 *(Conditional bundle tradeoff).* If x and y are bundle propositions, we have $v \in [\![c \Rightarrow x \succsim_{\mathrm{mt}} r : y]\!]_{\mathrm{v}}$ iff

$$\big(\nabla v(\vec{a}) \cdot \phi(x)\big) \geqslant r\big(\nabla v(\vec{a}) \cdot \phi(y)\big) \tag{8}$$

for each $\vec{a} \in [\![c]\!]_{\mathrm{a}}$.

Inequality (8) means that the partial derivatives of each utility function $v \in [\![x \succsim_{\mathrm{mt}} r : y]\!]_{\mathrm{v}}$ must satisfy the constraint

$$\sum_{x(i)\neq\perp} \frac{\partial v}{\partial A_i}(\vec{a})\phi_i(x) \geqslant r \sum_{y(j)\neq\perp} \frac{\partial v}{\partial A_j}(\vec{a})\phi_j(y) \tag{9}$$

at all points $\vec{a}$ satisfying the conditioning proposition.

For example, suppose we are comparing different computers and wish to state that a 3.6 GHz processor with a 1-MB L2-cache and an 800 MHz front side bus is twice as good as 4 GB of ram with a 400 MHz front side bus. We can express this in Lopat as

$$\begin{aligned} &GHz\big(speed\big(processor(X)\big)\big) = 3.6 \wedge MB\big(size\big(cache(X)\big)\big) = 1 \wedge GHz\big(speed\big(bus(X)\big)\big) \\ &\qquad = 0.8 \succsim_{\mathrm{mt}} 2 : GB\big(size\big(RAM(X)\big)\big) = 4 \wedge GHz\big(speed\big(bus(X)\big)\big) = 0.4. \end{aligned}$$

This formulation follows the representation used in Section 4.1 in making explicit both the dimensions of measurement (*speed*, *size*) and the units of measurement (*GHz*, *MB*, *GB*), and distinguishing both of these from the object of measurement (*processor*, *cache*, *bus*, *RAM*). In this case, we are talking about values for four different variables, $GHz(speed(processor(X)))$, $MB(size(cache(X)))$, $GHz(speed(bus(X)))$, and $GB(size(RAM(X)))$, which measure processor speed in GHz, cache size in MB, bus speed in GHz, and RAM size in GB. For typesetting convenience, we abbreviate these variables as *processor, cache, bus*, and *RAM* in the following discussion.

The preference stated above concerns two bundles, $\{(processor = 3.6), (cache = 1)(bus = 0.8)\}$ and $\{(RAM = 4), (bus = 0.4)\}$, and by Definition 5.1 constrains the utility function to obey

$$\frac{\partial v}{\partial processor}(\vec{a})\frac{3.6}{\sqrt{14.6}} + \frac{\partial v}{\partial cache}(\vec{a})\frac{1}{\sqrt{14.6}} + \frac{\partial v}{\partial bus}(\vec{a})\frac{0.8}{\sqrt{14.6}} \geqslant 2\left(\frac{\partial v}{\partial RAM}(\vec{a})\frac{4}{\sqrt{16.16}} + \frac{\partial v}{\partial bus}(\vec{a})\frac{0.4}{\sqrt{16.16}}\right).$$

We interpret comparisons between general value propositions by reducing them to comparisons between propositions in disjunctive normal form, and interpret marginal tradeoffs between DNF conditions p and q by regarding each conjunct in p and q as a bundle and interpreting the comparison between these disjunctions of bundles as pairwise comparisons of all combinations of the disjoined bundles.

Definition 5.2. The meaning of a conditional marginal tradeoff statement $c \Rightarrow r : p \succsim_{\mathrm{mt}} s : q$ is the same as the meaning of the statement rewritten so that the compared propositions are in disjunctive normal form, and the meaning of a conditional marginal tradeoff statement $c \Rightarrow r : p \succsim_{\mathrm{mt}} s : q$ in which p and q are in disjunctive normal form is the same as the meanings of all such comparisons between conjuncts in p and conjuncts in q. That is, if the disjunctive normal form of p is p' and the disjunctive normal form of q is q', then $[\![c \Rightarrow r : p \succsim_{\mathrm{mt}} s : q]\!]_{\mathrm{v}} = [\![c \Rightarrow r : p' \succsim_{\mathrm{mt}} s : q']\!]_{\mathrm{v}}$, and if $p = \bigvee X$ and $q = \bigvee Y$, then $[\![c \Rightarrow r : p \succsim_{\mathrm{mt}} s : q]\!]_{\mathrm{v}} = [\![\{c \Rightarrow r : x \succsim_{\mathrm{mt}} s : y \mid x \in X, y \in Y\}]\!]_{\mathrm{v}}$.

To continue the example from computer configuration, suppose we see other configurations of computers and amend our preference from before. We now think that a 3.2 GHz or 3.6 GHz processor with a 1-MB or 2-MB L2-cache and an 800 MHz front side bus is twice as good as 4 GB of ram with a 400 MHz front side bus. We have compound value propositions $p = (processor = 3.6 \vee processor = 3.2) \wedge (cache = 1 \vee cache = 2) \wedge bus = 0.8$, and $q = RAM = 4 \wedge bus = 0.4$. We then convert p into disjunctive normal form, obtaining $p = (w \vee x \vee y \vee z)$ where

$$\begin{aligned} w &= \big\{(processor = 3.6), (cache = 1)\big\}, \\ x &= \big\{(processor = 3.2), (cache = 1)\big\}, \\ y &= \big\{(processor = 3.6), (cache = 2)\big\}, \\ z &= \big\{(processor = 3.2), (cache = 2)\big\} \end{aligned}$$

and letting $q' = \{(RAM = 4), (bus = 0.4)\}$, we then have

$$[\![p \succsim_{\mathrm{mt}} 2 : q]\!]_{\mathrm{v}} = \big[\!\big[\big\{w \succsim_{\mathrm{mt}} 2 : q', x \succsim_{\mathrm{mt}} 2 : q', y \succsim_{\mathrm{mt}} 2 : q', z \succsim_{\mathrm{mt}} 2 : q'\big\}\big]\!\big]_{\mathrm{v}}.$$

5.3. Properties of the semantics

Conditional bundle tradeoffs exhibit a natural form of transitivity.

Theorem 5.2 *(Transitivity). The two conditional bundle tradeoffs*

$$c_1 \quad \Rightarrow \quad x \succsim_{\text{mt}} r_1 : y, \tag{10}$$

$$c_2 \quad \Rightarrow \quad y \succsim_{\text{mt}} r_2 : z \tag{11}$$

taken together entail the tradeoff statement

$$c_1 \wedge c_2 \quad \Rightarrow \quad x \succsim_{\text{mt}} r_1 r_2 : z. \tag{12}$$

Proof. The statements (10) and (11) are interpreted, respectively, as the constraints

$$\nabla u(\vec{a}) \cdot \phi(x) \geqslant r_1 \nabla u(\vec{a}) \cdot \phi(y), \tag{13}$$

$$\nabla u(\vec{a}) \cdot \phi(y) \geqslant r_2 \nabla u(\vec{a}) \cdot \phi(z). \tag{14}$$

Both (10) and (11) apply to outcomes falling within $c_1 \wedge c_2$, the intersection of their regions of applicability. Within this region, we can multiply (14) by r_1 and then substitute back into (13), to obtain the preference part of (12). □

We now verify that the definition of quantitative tradeoffs between groups of attributes is a generalization of the standard economic notion of the marginal rate of substitution between two attributes. More precisely, the marginal rate of substitution is usually defined as the negative of the slope of the indifference curve of the value function for two commodities [24]. Specifically, in the case of unit-vector tradeoffs between two attributes, our directional derivative representation for tradeoffs between sets of attributes reduces to a condition on the marginal rate of substitution. We show this by simplifying the condition in Definition 5.1.

Theorem 5.3 *(Marginal rate of substitution). If x is the bundle proposition $A_i = 1$, and y is the bundle proposition $A_j = 1$, with $i \neq j$, then the tradeoff ratio r in $x \succsim_{\text{mt}} r : y$ forms a lower bound on the marginal rate of substitution between attributes A_i and A_j, that is, if $v \in [\![x \succsim_{\text{mt}} r : y]\!]_{\text{v}}$, then*

$$\frac{\partial v}{\partial A_i} \Big/ \frac{\partial v}{\partial A_j} \geqslant r.$$

Proof. Definition 5.1 implies $\nabla v(\vec{a}) \cdot \phi(x) / \nabla v(\vec{a}) \cdot \phi(y) \geqslant r$. Expanding the inner product gives

$$\frac{\partial v}{\partial A_i} \Big/ \frac{\partial v}{\partial A_j} \geqslant r,$$

which states the claimed bound on the marginal rate of substitution. □

The description of tradeoff preferences we have given so far is very general. In research on preference elicitation, linear utility functions or piece-wise linear functions are considered exceedingly frequently [26,16]. It is thus worth noting that simple linear utility functions can satisfy a bundle tradeoff, stated formally in the following lemma.

Lemma 5.1 *(Linear utility functions). If $\mathcal{C} = \{C_1, C_2, \ldots, C_Q\}$ is a cover of A and $v(\vec{a}) = \sum_{i=1}^{Q} t_i v_i(\vec{a})$ is a generalized additive value function with $v_i = v_{C_i}$ such that v_i is linear in each attribute $A_{i1}, \ldots, A_{iN} \in C_i$, then $v \in [\![x \succsim_{\text{mt}} r : y]\!]_{\text{v}}$ iff*

$$\sum_{j=1}^{n} k_j \phi_j(x) \geqslant r \sum_{j=1}^{n} k_i \phi_j(y), \tag{15}$$

with k_i, k_j constants.

Proof. The claim follows from Definition 5.1. □

We will make use of this result in Section 8 when we construct linear value functions from preference statements in LOPAT.

Another property of the semantics we examine is the special case of colinear bundle propositions, that is, the meaning of $x \succsim_{\text{mt}} r : y$ when x and y are bundle propositions and there is some number t such that $\phi(x) = t\phi(y)$, that is, the

comparison involves attribute values in one bundle that are the same multiple of the values in the other bundle. In this case, the semantical condition (8) reduces to $(\nabla v(\vec{a}) \cdot t\phi(y)) \geqslant r(\nabla v(\vec{a}) \cdot \phi(y))$, which is satisfiable just in case either $t \geqslant r$ or $\nabla v(\vec{a}) \cdot \phi(y) = 0$.

A final property of our semantics is that it exhibits sensitivity to the units with which one expresses bundles, in that an attribute with a range of values between 0 and 1 will typically influence comparisons less than an attribute taking values in the range between 1000 and 1,000,000. This is a common problem arising in many situations in which one must measure or compare disparate attributes. One could approach the problem by normalizing attribute values and ranges in some automatic fashion, but as has been remarked in the literature on preference elicitation, normalizing ranges can lead to bias and misunderstanding instead of to a solution to the problem [25,3,18]. We therefore do not regard the problem as one solvable in the abstract for all problems, and leave the burden on the decision analyst to formulate tradeoffs sensibly in light of the differences among attributes and units of measurement. The literature on conjoint measurement theory [27] provides some useful results addressing these problems.

It is worth mentioning, however, that at least one sort of normalization is compatible with the semantics presented here. An earlier version of this paper used definitions that compared the bundles normalized to have unit Euclidean length, so as to compare directional derivatives without reference to the magnitudes of the vectors used to identify the directions. Formally, let $\mathbf{1}(x)$ to be the normalized vector $x/|x|$, and for a bundle x let $\mathbf{1}(x) = \mathbf{1}(\phi(x))$. The normalized semantics then defined the meaning of $c \Rightarrow x \succsim_{\mathrm{mt}} r : y$ so that $v \in [\![c \Rightarrow x \succsim_{\mathrm{mt}} r : y]\!]_{\mathrm{v}}$ iff $(\nabla v(\vec{a}) \cdot \mathbf{1}(x)) \geqslant r(\nabla v(\vec{a}) \cdot \mathbf{1}(y))$ for each $\vec{a} \in [\![c]\!]_{\mathrm{a}}$. Essentially all of the theorems we state and prove in this paper hold true with the normalized semantics as well: only the examples of utility construction differ by much. Nevertheless, this simple sort of normalization does not really alter the underlying problem significantly. Instead, it introduces an additional complication, for making tradeoff comparisons by only considering the direction of different bundles but not their magnitudes means that one cannot treat comparison of colinear bundles with the normalized definition, and must reintroduce the unnormalized semantics given here for that special case, producing an unpleasant discontinuity of interpretation.

5.4. Marginal tradeoffs among discrete attributes

There is a natural extension of our directional derivatives formulation of tradeoffs to discontinuous value functions over discrete attributes. When we have two bundles x, y over discrete attributes and want to say that value is increasing in the $\phi(x)$-direction r times faster than in the $\phi(y)$-direction, we can still give meaning to this type of preference by using discrete difference analogues of the partial derivatives.

For bundles of either discrete or continuous attributes, we define a *discrete difference vector* $\Delta v(x, y)$ of complete bundles x, y, to be a vector with ith component

$$\Delta_i v(x, y) = \begin{cases} \frac{v(\phi(x)) - v(\phi(x[(i=\phi_i(y))]))}{\phi_i(x) - \phi_i(y)} & \text{if } \phi_i(x) - \phi_i(y) \neq 0, \\ 0 & \text{if } \phi_i(x) - \phi_i(y) = 0. \end{cases}$$

A discrete difference vector is a vector of slope-approximations, where the ith component of the vector is an approximation of the slope of v in the ith dimension.

We assign meanings to tradeoffs among discrete attributes using discrete difference vectors as follows.

Definition 5.3 *(Discrete bundle tradeoffs).* If x and y are bundles over discrete attributes, then $v \in [\![c \Rightarrow x \succsim_{\mathrm{mt}} r : y]\!]_{\mathrm{v}}$ iff

$$\Delta v\big(a, a'\big) \cdot \phi(x) \geqslant r \Delta v\big(a, a'\big) \cdot \phi(y). \tag{16}$$

for all complete bundles $a \neq a'$ such that $\phi(a), \phi(a') \in [\![c]\!]_{\mathrm{a}}$.

Consider a simple example of how one might use Definition 5.3 to analyze a preference stated by a certain computer scientist who in the morning thinks that a small cup of caffeinated coffee is twice a preferable as a large cup of decaffeinated coffee. The aim of this example is to illustrate our inferences about the scientist's preferences, not to attribute inferences to the scientist.

We begin by considering attributes *coffee* with domain $\{\mathit{decaf}, \mathit{regular}\}$, and *size* with domain $\{S, M, L\}$. For simplicity, suppose $\rho(\mathit{decaf}) = 1$, $\rho(\mathit{regular}) = 2$, and for *size* we let $\rho(S) = 1$, $\rho(M) = 2$, $\rho(L) = 3$. We then have bundles $x = (\mathit{coffee} = \mathit{regular})$, $(\mathit{size} = S)$, $y = (\mathit{coffee} = \mathit{decaf})$, $(\mathit{size} = L)$, and a discrete bundle tradeoff $x \succsim_{\mathrm{mt}} > 2 : y$. This tradeoff holds when $\Delta v(a, a') \cdot \phi(x) \geqslant 2\Delta v(a, a') \cdot \phi(y)$ for all (a, a'). For example, first consider $a = (1, 1), a' = (2, 3)$. We have

$$\Delta_1 v\big((1, 1), (2, 3)\big) = \big(v(1, 1) - v(2, 1)\big)/(1 - 2),$$
$$\Delta_2 v\big((1, 1), (2, 3)\big) = \big(v(1, 1) - v(1, 3)\big)/(1 - 3).$$

This lets us compute the dot-products as follows:

$$\Delta v\big((1, 1), (2, 3)\big) \cdot \phi(x) = -2\big(v(1, 1) - v(2, 1)\big) - 1/2 * \big(v(1, 1) - v(1, 3)\big),$$
$$\Delta v\big((1, 1), (2, 3)\big) \cdot \phi(y) = -1\big(v(1, 1) - v(2, 1)\big) - 3/2 * \big(v(1, 1) - v(1, 3)\big)$$

and these leave us with the constraint on the utility function:

$$-2.5v(1,1)+2v(2,1)+0.5v(1,3)\geqslant 2*\big(-2.5v(1,1)+v(2,1)+1.5v(1,3)\big)$$

which simplifies to $v(1,1)\geqslant v(1,3)$.

By considering other values of (a,a') we obtain other constraints on the utility function. Consider $\Delta v(a',a)$. After simplification we obtain $v(2,1)\geqslant v(2,3)$. Let $b=(2,1), b'=(1,3)$, then we achieve the constraint $v(2,1)\geqslant v(2,3)$. Considering $\Delta v(b',b)$ gives $v(1,1)\geqslant v(1,3)$. Suppose we define $c=(1,1), c'=(1,3)$, in this case $\Delta_1 v(c,c')=0$, but after simplifying we again get the constraint $v(1,1)\geqslant v(1,3)$, and in this case reversing the arguments to Δ gives the identical constraint. We list these and other constraints in the following table:

a	a'	Constraint
$(1,1)$	$(2,3)$	$v(1,1)\geqslant v(1,3)$
$(2,3)$	$(1,1)$	$v(2,1)\geqslant v(2,3)$
$(2,1)$	$(1,3)$	$v(2,1)\geqslant v(2,3)$
$(1,3)$	$(2,1)$	$v(1,1)\geqslant v(1,3)$
$(1,1)$	$(1,3)$	$v(1,1)\geqslant v(1,3)$
$(1,3)$	$(1,1)$	$v(1,1)\geqslant v(1,3)$
$(1,1)$	$(2,1)$	$0\geqslant 0$
$(2,1)$	$(1,1)$	$0\geqslant 0$
$(2,3)$	$(2,1)$	$v(2,1)\geqslant v(2,3)$
$(2,1)$	$(2,3)$	$v(2,1)\geqslant v(2,3)$
$(2,3)$	$(1,3)$	$0\geqslant 0$
$(1,3)$	$(2,3)$	$0\geqslant 0$

The results here suffice to order several values of the domain, such that $(2,1)$ which corresponds to $(\mathit{regular}, S)$ is the element most preferred and (decaf, L) is the least. In contrast to the continuous development in the previous section, values of $\Delta v(a,a')$ can be used to provide additional constraints, but we conjecture that only values of a, a' such that $a_i = x_i$ or $a_i = y_i$ will provide new, unentailed, constraints. A proof of this conjecture is left to future work.

The following theorem shows the meanings assigned to $c \Rightarrow x \succsim_{\mathrm{mt}} r : y$ by Definition 5.3 for discrete tradeoffs is compatible with the meaning assigned to the same statement by Definition 5.1 for continuous tradeoffs in the case in which we take continuous attributes and a differentiable value function and regard the attributes instead as discrete.

Theorem 5.4 *(Continuous tradeoffs). If x and y are bundles of continuous attributes and v is a differentiable function over $\vec{A}$ that satisfies the discrete inequality* (16) *for all complete bundles $a \neq a'$ such that $\phi(a), \phi(a') \in [\![c]\!]_{\mathrm{a}}$, then v also satisfies the continuous inequality* (8)*.*

Proof. By assumption we have

$$\Delta v\big(a,a'\big)\cdot\phi(x)\geqslant r\Delta v\big(a,a'\big)\cdot\phi(y) \tag{17}$$

for all complete bundles $a\neq a'\in[\![c]\!]_{\mathrm{a}}$. Since v is differentiable, the terms in the expansion of the discrete difference are approximations to the slopes in each dimension of the value function. Consider the term

$$v\big(\phi(a)\big)-v\big(\phi\big(a\big[\big(i=\phi_i\big(a'\big)\big)\big]\big)\big)\big/\big(\phi_i(a)-\phi_i\big(a'\big)\big), \tag{18}$$

which holds at all a, a', for $\phi_i(a)\neq\phi_i(a')$, and approximates the slope of v in the ith dimension. We can choose a' such that $\phi_i(a')=\phi_i(a)-h$ for some h. We are then assured that for differentiable v relations involving (18) hold when (18) is replaced with

$$\lim_{h\to 0} v\big(\phi(a)\big)-v\big(\phi\big(a\big[\big(i=\phi_i(a)-h\big)\big]\big)\big)/h. \tag{19}$$

Since (19) is equal to $\frac{\partial v}{\partial i}$, we can rewrite (17) as

$$\sum_i^n \frac{\partial v}{\partial i}(\vec{a})\phi_i(x)\geqslant r\sum_j^n \frac{\partial v}{\partial j}(\vec{a})\phi_j(y),$$

which is the definitional condition for the continuous inequality (8). □

6. Qualitative *ceteris paribus* preferences

Qualitative *ceteris paribus* preference statements state that one condition is preferable to another, other things being equal. The conditions are typically expressed as logical combinations of values of various attributes, so "other things being equal" means comparing only outcomes that do not differ on attributes not involved in either of the conditions being compared. The comparison, moreover, is made without implying anything about possible tradeoff ratios. Qualitative *ceteris paribus* preferences have served as the basis for numerous representations of preference information, including [14,32,5].

6.1. Reinterpreting qualitative ceteris paribus preferences

The semantics given here to the qualitative *ceteris paribus* preference statements included in LOPAT differs somewhat from that given to comparable statements treated in [14]. We explain the difference in the following. To keep the two conceptions separate, we use the comparison operators $\succ_{\text{DSW}}$ and $\succsim_{\text{DSW}}$ in this section to refer to statements and semantics of the form considered in [14]. Statements involving $\succ_{\text{DSW}}$ or $\succsim_{\text{DSW}}$ are not included in LOPAT.

The *support* of a value proposition p, denoted $\sigma(p)$, is the minimal set of attributes determining the truth of p. Bundles b and b' are *equivalent modulo* p if they take the same values outside the support of p. Formally, $b \equiv b' \bmod p$ iff $b[a] = b'[a]$ for some bundle a over $\sigma(p)$. More generally, we say that $b \equiv b' \bmod p_1, p_2, \ldots$ if $b[a] = b'[a]$ for some bundle over $\sigma(p_1) \cup \sigma(p_2) \cup \ldots$. With these notions in hand, we can restate the meaning definition from [14] as follows.

Definition 6.1 *(Qualitative* ceteris paribus *preferences, Definition 3 in [14]).* If p and q are value propositions and v is a value function, then $v \in [\![c \Rightarrow p \succsim_{\text{DSW}} q]\!]_{\text{v}}$ iff $v(b) \geqslant v(b')$ for all bundles $b, b' \in [\![c]\!]_{\text{a}}$ such that $b \models (p \wedge \neg q)$, $b' \models (q \wedge \neg p)$, and $b \equiv b' \bmod p, q$. Similarly, $v \in [\![c \Rightarrow p \succ_{\text{DSW}} q]\!]_{\text{v}}$ iff $v \in [\![c \Rightarrow p \succsim_{\text{DSW}} q]\!]_{\text{v}}$ and $v(b) > v(b')$ for some $b, b' \in [\![c]\!]_{\text{a}}$ such that $b \models (p \wedge \neg q)$, $b' \models (q \wedge \neg p)$, and $b \equiv b' \bmod p, q$.

For the comparisons expressed in LOPAT, tradeoffs between discrete attributes generalize *ceteris paribus* preferences between binary attributes. This formulation is stated most simply in terms of binary attributes. A *ceteris paribus* tradeoff between two binary attributes can be thought of as stating that "a change in P from $\neg p$ to p is preferable to a change in Q from $\neg q$ to q." For discrete attributes with larger domains, the statement becomes "a change in P from p_j to p_i is preferable to a change in Q from q_j to q_i." The definition that follows implements this intuition.

Definition 6.2 *(Qualitative* ceteris paribus *tradeoffs).* If p and q are value propositions with $\sigma(p) = \sigma(q)$ and v is a value function, then $v \in [\![c \Rightarrow p \succsim_{\text{cp}} q]\!]_{\text{v}}$ if and only if

$$\Delta v\big(a[p], a[q]\big) \cdot \phi(p) \geqslant \Delta v\big(a[q], a[p]\big) \cdot \phi(q) \tag{20}$$

for all complete bundles a such that $\phi(a) \in [\![c]\!]_{\text{a}}$.

For example, we can state preference concerning wine and food in the context of classical dining; suppose we have attributes $A = \{meal, wine\}$ and let *meal* have two values: $b_1 = meat$, $b_2 = fish$ and *wine* have two values: $w_1 = red$, $w_2 = white$. We can then define clauses $p_1 = fish \wedge white$ and $q_1 = fish \wedge red$, and state qualitative *ceteris paribus* tradeoff $p_1 \succ_{\text{cp}} q_1$. We state another preference using clauses $p_2 = meat \wedge red$ and $q_2 = meat \wedge white$, where $p_2 \succ_{\text{cp}} q_2$. We make the assumptions that $\rho(meat) = 1$, $\rho(fish) = 2$, $\rho(red) = 1$, and $\rho(white) = 2$. Using these simplifications, we proceed as we did in Section 5.4 and consider the values of $\Delta v(a[p], a[q])$. Since we have only two attributes, a is an empty bundle, and so $\Delta v(a[p_1], a[q_1]) = \Delta v((2,2),(2,1))$

$$\Delta_1 v\big((2,2),(2,1)\big) = 0, \qquad \Delta_2 v\big((2,2),(2,1)\big) = v(2,2) - v(2,1).$$

This lets us compute the dot-products as follows:

$$\Delta v\big((2,2),(2,1)\big) \cdot \phi(p_1) = 2\big(v(2,2) - v(2,1)\big), \qquad \Delta v\big((2,2),(2,1)\big) \cdot \phi(q_1) = v(2,2) - v(2,1)$$

and these leave us with the constraint on the utility function $v(2,2) > v(2,1)$. Similarly, we consider the condition from $p_2 \succ q_2$, or $\Delta v((1,1),(1,2)) \cdot \phi(p_2) > \Delta v((1,1),(1,2)) \cdot \phi(q_2)$, and through similar calculations, obtain $v(1,1) > v(1,2)$. Restating this in terms of the qualitative domains of each attribute leaves us with $v(fish, white) > v(fish, red)$, and $v(meat, red) > v(meat, white)$; note that these mirror the stated *ceteris paribus* preferences: $fish \wedge white \succ_{\text{cp}} fish \wedge red$ and $meat \wedge red \succ_{\text{cp}} meat \wedge white$.

The semantics given here for *ceteris paribus* preferences differs from that given in [14], in that [14] interprets $p \succsim_{\text{DSW}} q$ in terms of bundles satisfying $p \wedge \neg q$ and $q \wedge \neg p$. The support of these propositions can differ in some cases, and our present semantics avoids this complication.

The following theorem shows that, for linear value functions, the definition given for qualitative *ceteris paribus* comparisons obeys a useful property of the value function: that the outcomes satisfying the left-hand side of the preference relation have greater value than those on the right-hand side.

Theorem 6.1. *If x, y are bundles and v is a linear function in $[\![c \Rightarrow x \succsim_{\text{cp}} y]\!]_{\text{v}}$, then*

$$v\big(a[x]\big) \geqslant v\big(a[y]\big) \tag{21}$$

for each complete bundle a such that $\phi(a) \in [\![c]\!]_{\text{a}}$.

Proof. For $v \in [\![c \Rightarrow x \succsim_{\text{cp}} y]\!]_{\text{v}}$, the definition of qualitative tradeoffs expands to

$$\sum_{i \in \sigma(x)} \frac{v(a[x_i]) - v(a[y_i])}{x_i - y_i} x_i \geqslant \sum_{j \in \sigma(y)} \frac{v(a[y_j]) - v(a[x_j])}{y_j - x_j} y_j.$$

Multiply those terms that have negative denominators by $\frac{-1}{-1}$, let z^+ be the indices of $\phi(x)$ and $\phi(y)$ such that $x_i > y_i$, and let z^- be the indices where $x_j < y_j$. We then can write

$$\sum_{i \in z^+} \frac{v(a[x_i]) - v(a[y_i])}{x_i - y_i} (x_i - y_i) \geqslant \sum_{j \in z^-} \frac{v(a[y_j]) - v(a[x_j])}{y_j - x_j} (y_j - x_j).$$

The differences in the numerators of the above summations are simply the increase of the value function in a particular dimension. When v is linear, this slope is constant, and without loss of generality let t_i be the slope in the i-dimension, and t_j be the slope in the j-dimension. The preceding inequality simplifies to

$$\sum_{i \in z^+} t_i (x_i - y_i) \geqslant \sum_{j \in z^-} t_j (y_j - x_j),$$

which we rearrange to get

$$\sum_{i \in \sigma(x)} t_i x_i - \sum_{j \in \sigma(y)} t_j y_j \geqslant 0.$$

This inequality is just

$$v\big(z[x]\big) - v\big(z[y]\big) \geqslant 0,$$

as required. □

The preceding theorem can be used to draw a correspondence between the definitions of *ceteris paribus* preference given in Definitions 6.1 and 6.2, showing that a comparison $x' \succsim_{\text{DSW}} y'$ of two literals in the language of [14] can be represented by the same value function as represents the LOPAT bundle comparison $b[x', \neg y'] \succsim_{\text{cp}} b[\neg x', y']$ for an empty bundle b.

Theorem 6.2. *Let x' and y' be two binary attributes with $x' \neq y'$, let b be an empty bundle, and let x and y be bundles such that $x = b[x', \neg y']$ and $y = b[\neg x', y']$. Then every linear value function in $[\![c \Rightarrow x \succsim_{\text{cp}} y]\!]_{\text{v}}$ is also in $[\![c \Rightarrow x' \succsim_{\text{DSW}} y']\!]_{\text{v}}$.*

Proof. Suppose $v \in [\![c \Rightarrow x \succsim_{\text{cp}} y]\!]_{\text{v}}$ is linear. By Theorem 6.1, we have $v(a[x]) \geqslant v(a[y])$ for every complete bundle $a \in [\![c]\!]_{\text{a}}$. Now we clearly have $a[x] \equiv a[y] \bmod x', y'$, so by Definition 6.1, v is in $[\![c \Rightarrow x' \succsim_{\text{DSW}} y']\!]_{\text{v}}$. □

This result shows that some statements of weak preference between binary attributes expressed using $\succsim_{\text{DSW}}$ can be transformed into statements of weak preference expressed using $\succsim_{\text{cp}}$ such that they are satisfied by the same linear value functions, if any exist.

7. Marginal attribute importance

We have so far considered tradeoffs between particular instances of attributes: an assignment to some attributes G is preferred to an assignment to some attributes H, *ceteris paribus*, or by some factor r. In this section we describe tradeoffs of the form $G \succ_{\text{ai}} r : H$ between groups of attributes, which can also be considered a statement about the *importance* of the groups of attributes, namely that "attributes G are more important than attributes H by a factor of r."

By saying that one set of attributes is more important than another, we mean to say that the first set has a larger influence on the value function than does the second. The influence of an attribute on value does not depend on any direction in the space of outcomes, on the influence being a positive or negative contribution to value, or on the particular instantiations of attributes being considered. It is, instead, purely a measure of the weight assigned to an attribute itself. We expect such comparisons mainly arise during elicitation of preferences, in which one might want to talk about the relative value of attributes and sets of attributes.

The "importance" of attributes is a quantity that is frequently used in traditional decision analysis, and McGeachie [30] reviews various techniques for obtaining, assuming, or computing the relative importance of attributes in a value function. Our presentation here extends the usual decision analysis methodology in three ways. The first is that, as was stated in the introduction to this paper, the decision maker is not required to make any importance statements at all, for we merely take as many importance statements as occur and consider them alongside the other preference information we have. The second is that we do not expect decision makers to talk about importance in just one attribute. The importance statements we describe herein are between sets of attributes. Someone may decide that the combination of restaurant attributes *meal-quality*, *drink-quality*, and *atmosphere-quality* are at least twice as important as *time-to-arrive* and *time-spent-waiting* at a restaurant. The third difference is that instead of simply assigning a numeric weight to each attribute as a way of expressing attribute importance, as is often done, we instead interpret importance comparisons in the same geometric framework used for marginal tradeoff statements, in particular as a comparison between the norms of the gradients associated with different sets of attributes.

We formalize the geometric interpretation of conditional attribute importance statements as follows. Given an arbitrary subset $G \subseteq A$ and a function v_G over $\vec{G}$, the gradient $\nabla v_G(\vec{x})$ at a point $\vec{x} \in \vec{G}$ is a vector based at x pointing in the direction of maximum increase of v_G in $\vec{G}$. The length $|\nabla v_G(\vec{x})|$ of that vector is the magnitude of that increase. Thus if we interpret a tradeoff between a set of attributes G and another set of attributes H as a comparison between the maximum possible rates of increase in the subspaces defined by G and H, we can write that comparison in terms of the magnitudes of gradients in those spaces. Specifically,

$$\left|\nabla v_G(\vec{a})\right| \geqslant r\left|\nabla v_H(\vec{a})\right| \tag{22}$$

compares the increase in the G-space to the increase in the H-space. Further, if we choose the L_1 norm to measure the length of the above vectors, inequality (22) is equivalent to

$$\sum_{f\in G}\left|\frac{\partial v}{\partial f}(\vec{a})\right| \geqslant r\sum_{f\in H}\left|\frac{\partial v}{\partial f}(\vec{a})\right|. \tag{23}$$

Let $\vec{x}$ and $\vec{y}$ be the characteristic vectors for G and H, respectively, then (23) is equivalent to

$$\sum_{i=1}^{|A|}\left|\frac{\partial v}{\partial A_i}(\vec{a})x_i\right| \geqslant r\sum_{i=1}^{|A|}\left|\frac{\partial v}{\partial A_i}(\vec{a})y_i\right|. \tag{24}$$

With this in mind, we define the meaning of attribute importance tradeoffs as follows. We write *abs*() to denote the *absolute value of the gradient*, that is, the absolute value of each element in a vector, so that $abs(\vec{x}) = \langle |x_1|, |x_2|, \ldots, |x_n|\rangle$. We use this notation to distinguish the absolute value from the length of the vector, $|\vec{x}|$. For the following statement, recall that a vector $\vec{x} \in \vec{A}$ is the characteristic vector for G if $\vec{x}$ is such that $x_i = 1$ iff $A_i \in G$ and $x_i = 0$ otherwise.

Definition 7.1 *(Attribute importance tradeoffs).* If $\vec{x}$ and $\vec{y}$ are characteristic vectors for attribute sets G and H respectively, then $v \in [\![c \Rightarrow G \succsim_{\text{ai}} r : H]\!]_{\text{v}}$ iff

$$\big(abs\big(\nabla v(\vec{a})\big)\big)\cdot\vec{x} \geqslant r\big(abs\big(\nabla v(\vec{a})\big)\cdot\vec{y}\big) \tag{25}$$

holds on all points $\vec{a} \in [\![c]\!]_{\text{a}}$.

The condition (25) is clearly equivalent to (24) and so also to (23).

Consider a simple example. Suppose we are going out to eat and need to pick a restaurant. In this context, among the things we might consider are the time required: the time to get to a given restaurant, and the time spent waiting once at the restaurant. For instance, we can let *travel time (tt)* and *wait time (wt)* be continuous attributes measured in *minutes*, where less is better. We state a tradeoffs about these attributes: $minutes(waittime) \succ_{\text{ai}} 1.5 : minutes(traveltime)$, which indicates that, in this estimation, it is roughly 50% more annoying to wait at the restaurant than to travel to it. For brevity, we define $A = \{wt, tt\}$, as shorthand for the longer propositional attributes. The characteristic vectors for *wt* and *tt* are, respectively, $(1, 0)$ and $(0, 1)$. From these, we can compute the condition provided on the utility function by Eq. (25), and obtain: $|\frac{\partial v}{\partial wt}(\vec{x})| > 1.5|\frac{\partial v}{\partial tt}(\vec{x})|$.

Definition 7.1 thus relates the "maximum increase" measure of importance between attribute sets to the "directional-derivative" representation of importance comparisons. This correspondence allows us to use the intuitive characterization of importance tradeoffs as comparisons of the maximum increase in two different subspaces while extending the framework of partial derivatives presented in Section 5 for the formal semantics.

One may object that comparing the maximum increase of value in different subspaces seems an arbitrary choice, for we could compare other statistics of the spaces instead, such as the average increase, the median increase, or the increase at the origin. However, comparing the maximum increase of value in a space is appropriate in many cases, especially in what has been called "configuration problems," in which the goal is to find the configuration of some elements (a schedule, a composite product, a results set, etc.) in a domain that maximizes the utility of the configuration. These situations are

characterized by the presence of a rational actor that chooses the configuration or outcome, and a lack of uncertainty: however the choice among outcomes is made, that is the outcome that results. In these cases, a rational actor will always choose the outcome with greatest value, so the comparison of interest is between maximum increases in various constrained spaces of the configuration problem.

7.1. Properties of the semantics

We first present two simple corollaries to Definition 7.1.

Corollary 7.1. *If $G\backslash H \neq \emptyset$, then $c \Rightarrow G \gtrsim_{\text{ai}} r : H$ is satisfiable.*

Proof. If v is a linear function of the form $v(\vec{a}) = \sum_i t_i v_i(\vec{a})$, then according to (23), v satisfies $c \Rightarrow G \gtrsim_{\text{ai}} r : H$ just in case that

$$\sum_{A_i \in G} |t_i| \geqslant r \sum_{A_j \in H} |t_j|.$$

Now choose the weights for attributes in $G\backslash H$ so that the weight t of each attribute $f \in G\backslash H$ satisfies

$$t > r \sum_{A_j \in H} |t_j|,$$

and choose weights for attributes not in $G\backslash H$ arbitrarily. The function v so defined then satisfies $v \in [\![c \Rightarrow G \gtrsim_{\text{ai}} r : H]\!]_{\text{v}}$. □

Corollary 7.2. *The conditional attribute tradeoff statement $c \Rightarrow G \gtrsim_{\text{ai}} r : G$ is satisfiable only if $r \leqslant 1$ or if ∇v is zero over attributes G.*

Proof. We have $v \in [\![c \Rightarrow G \gtrsim_{\text{ai}} r : H]\!]_{\text{v}}$ only if

$$\big(abs\big(\nabla v(\vec{a})\big) \cdot \vec{x}\big) \geqslant r\big(abs\big(\nabla v(\vec{a})\big) \cdot \vec{x}\big)$$

for each $\vec{a} \in [\![c]\!]_{\text{a}}$. This is satisfied by any $r \leqslant 1$, and when the inner product of the absolute value of the gradient of v at $\vec{a}$ with $\vec{x}$ is zero. Since $\vec{x}$ is a characteristic vector for G, it contains only zeros and ones. Since we take the absolute value of $\nabla v(\vec{a})$, the elements of that vector are all greater or equal to zero. The inner product of two nonnegative vectors is zero only when each of the terms in the summation are zero. This is the case only when the gradient is zero in each attribute in G. □

Our definition of an attribute tradeoff ratio leaves open the possibility that the two sets of attributes involved are not disjoint, in which case the following result applies.

Theorem 7.1 *(Intersecting attributes). If G, H are sets of attributes with $J = G \cap H$ and $v \in [\![c \Rightarrow G \gtrsim_{\text{ai}} r : H]\!]_{\text{v}}$, then*

$$\sum_{f \in (G\backslash J)} \left|\frac{\partial v}{\partial f}(\vec{a})\right| \geqslant r \sum_{f \in (H\backslash J)} \left|\frac{\partial v}{\partial f}(\vec{a})\right| + (r-1) \sum_{f \in J} \left|\frac{\partial v}{\partial f}(\vec{a})\right|$$

for all $\vec{a} \in [\![c]\!]_{\text{a}}$.

We omit the proof, but it follows directly from Definition 7.1 by rearranging terms. An important corollary of Theorem 7.1 is the case in which one set of attributes contains the other.

Corollary 7.3. *If G and H are sets of attributes such that $H \subset G$ and $r > 1$, then*

$$[\![c \Rightarrow G \gtrsim_{\text{ai}} r : H]\!]_{\text{a}} = \big[\!\big[c \Rightarrow G\backslash H \gtrsim_{\text{ai}} (r-1) : H\big]\!\big]_{\text{a}}.$$

Proof. The proof is straightforward. By definition $G \gtrsim_{\text{ai}} r : H$ is

$$\sum_{f \in G} \left|\frac{\partial v}{\partial f}(\vec{a})\right| \geqslant r \sum_{f \in H} \left|\frac{\partial v}{\partial f}(\vec{a})\right|.$$

Since $H \subset G$, we can split the first summation, giving

$$\sum_{f \in (G\backslash H)} \left|\frac{\partial v}{\partial f}(\vec{a})\right| + \sum_{f \in H} \left|\frac{\partial v}{\partial f}(\vec{a})\right| \geqslant r \sum_{f \in H} \left|\frac{\partial v}{\partial f}(\vec{a})\right|.$$

From this it is obvious that we have

$$\sum_{f\in(G\setminus H)}\left|\frac{\partial v}{\partial f}(\vec{a})\right| \geqslant (r-1)\sum_{f\in Y}\left|\frac{\partial v}{\partial f}(\vec{a})\right|,$$

which establishes our equivalence. □

Theorem 7.1 shows that intersecting attributes imply simplified constraints on the utility function, while Corollary 7.3 shows that only in the case of one attribute set being contained within the other does this reduce to an attribute tradeoff between disjoint sets of attributes.

For an example of a different character, someone preparing for armed conflict might wish to state that "Guns and bullets together are r-times more important than guns or bullets individually." This is an example of complementary attributes that are worth more together than they are separately, and it captures the desire of the quartermaster to balance, somehow, the amount of guns acquired and the amount of ammunition for those guns. Let us assume for simplicity that in this case $A = \{\#guns, \#bullets\}$, which are continuous attributes representing the number of guns and of bullets. Then suppose that the decision maker encodes his intuition as two importance statements: firstly, $\{\#guns, \#bullets\} \succ_{\text{ai}} r : \{\#guns\}$, and secondly, $\{\#guns, \#bullets\} \succ_{\text{ai}} r : \{\#bullets\}$. By applying Corollary 7.3 these are equivalent to the pair

$$\{\#bullets\} \succ_{\text{ai}} r-1 : \{\#guns\},$$
$$\{\#guns\} \succ_{\text{ai}} r-1 : \{\#bullets\}.$$

It is clear that these cannot hold simultaneously.

We remark, however, that it is possible to express a related sentiment using conditional attribute importance statements. We could state that:

$$\big(\#guns > k(\#bullets)\big) \quad\Rightarrow\quad \{\#bullets\} \succ_{\text{ai}} r : \{\#guns\},$$
$$\left(\#bullets > \frac{1}{k}(\#guns)\right) \quad\Rightarrow\quad \{\#guns\} \succ_{\text{ai}} r : \{\#bullets\}.$$

Together, these mean that while the number of one attribute, *guns*, is k times more than the number of *bullets*, it is more important to gain additional *bullets*; while if this is not the case, then the attribute importance is reversed.

Just as with bundle tradeoffs, attribute tradeoff statements reduce to linear conditions on the parameters of linear value functions. We state here a lemma for attribute tradeoffs similar to Lemma 5.1. Recall that for a bundle b, $\sigma(b)$ is the support of b, the set of attributes assigned values other than $\perp$ in b.

Lemma 7.1. *Suppose that x and y are bundles and $v(\vec{a}) = \sum_{i=1}^{Q} t_i v_i(\vec{a})$ is a generalized additive value function for a cover $\mathcal{C} = \{C_1, C_2, \ldots, C_Q\}$ of A, with $v_i = v_{C_i}$. If v is linear in each $A_j \in A$, then $v \in [\![\sigma(x) \succsim_{\text{ai}} r : \sigma(y)]\!]_{\text{v}}$ iff*

$$\sum_{i\in\sigma(x)} |k_j| \geqslant r \sum_{i\in\sigma(y)} |k_j|. \tag{26}$$

Just as tradeoff statements are transitive over common conditions, so are statements of conditional importance.

Theorem 7.2 *(Attribute transitivity). The conditional attribute tradeoff statements*

$$c_1 \quad\Rightarrow\quad G \succsim_{\text{ai}} r_1 : G',$$
$$c_2 \quad\Rightarrow\quad G' \succsim_{\text{ai}} r_2 : G''$$

taken together entail the tradeoff statement

$$c_1 \wedge c_2 \quad\Rightarrow\quad G \succsim_{\text{ai}} r_1 r_2 : G''.$$

We omit the proof, which parallels that of Theorem 5.2 exactly.

7.2. Importance of discrete attributes

We extend attribute importance comparisons over continuous value functions to comparisons over discrete attributes, as we have done with bundle tradeoffs. Again, we use discrete difference equations as analogues to partial derivatives.

Definition 7.2 *(Discrete importance statements).* If $G, H \subseteq A$, then $v \in [\![c \Rightarrow G \succsim_{\text{ai}} r : H]\!]_{\text{v}}$ iff G is mutually preferentially independent of $\overline{G}$, H is mutually preferentially independent of $\overline{H}$, and, letting $\vec{x}$ and $\vec{y}$, denote the characteristic vectors for G and H, respectively, we have

$$abs\big(\Delta v\big(a, a'\big)\big) \cdot \vec{x} \geqslant r\big(abs\big(\Delta v\big(a, a'\big)\big) \cdot \vec{y}\big) \tag{27}$$

for all complete bundles a, a' such that $\phi_i(a) \neq \phi_i(a')$ with $\phi(a), \phi(a') \in [\![c]\!]_{\text{a}}$.

It should be clear that this definition allows a correspondence between the discrete case and the continuous case, much as using a discrete approximation of partial derivatives sufficed in the case of attribute tradeoffs. The arguments involved parallel those of Definition 5.3. Because much of what is said of discrete attribute tradeoffs can be said of discrete importance tradeoffs, we do not repeat those statements here. We note that the limitations on (a, a') in the definition parallel the differentiability constraints of the continuous case.

We will, however, revisit the case of the coffee-drinking scientist. In our previous incarnation of this example, we saw that a small cup of caffeinated coffee is twice a preferable as a large cup of decaffeinated coffee. Suppose we make the tradeoff more explicit with an attribute importance statement: *coffee* is equally or more important than *size*. Recall attribute *coffee* has domain $\{decaf, regular\}$, and *size* has domain $\{S, M, L\}$. We thus have a preference *coffee* $\succ_{\text{ai}}$ *size*. Using the translation defined before, $\rho(decaf) = 1$, $\rho(regular) = 2$, and for *size* we let $\rho(S) = 1$, $\rho(M) = 2$, $\rho(L) = 3$. The characteristic vector for *coffee* is then $x = (1, 0)$, and for *size* $y = (0, 1)$. Eq. (27) then becomes:

$$abs\big(\Delta v\big(a, a'\big)\big) \cdot (1, 0) \geqslant abs\big(\Delta v\big(a, a'\big)\big) \cdot (0, 1).$$

Consider $a = (1, 3)$, $a' = (2, 1)$, and the above becomes

$$\big|\Delta_1 v\big((1, 3), (2, 1)\big)\big| = \big|\big(v(1, 3) - v(2, 3)\big)/(1 - 2)\big|,$$
$$\big|\Delta_2 v\big((1, 3), (2, 1)\big)\big| = \big|\big(v(1, 3) - v(1, 1)\big)/(3 - 1)\big|.$$

This lets us compute the dot-products as follows:

$$abs\big(\Delta v\big((1, 3), (2, 1)\big)\big) \cdot (1, 0) = \big|-1\big(v(1, 3) - v(2, 3)\big)\big| - 0,$$
$$abs\big(\Delta v\big((1, 3), (2, 1)\big)\big) \cdot (0, 1) = 0 + \big|0.5\big(v(1, 3) - v(1, 1)\big)\big|$$

and these leave us with the constraint on the utility function: $|v(2, 3) - v(1, 3)| \geqslant 0.5|v(1, 3) - v(1, 1)|$. Similarly, $a = (2, 1)$, $a' = (1, 3)$ gives us the constraint on the utility function: $|v(2, 1) - v(1, 1)| \geqslant 0.5|v(2, 3) - v(2, 1)|$. Other values of (a, a') are shown in the table below, together with the constraint they imply on the utility function.

a	a'	Constraint
$(1, 3)$	$(2, 1)$	$\|v(2, 3) - v(1, 3)\| \geqslant 0.5\|v(1, 3) - v(1, 1)\|$
$(2, 1)$	$(1, 3)$	$\|v(2, 1) - v(1, 1)\| \geqslant 0.5\|v(2, 3) - v(2, 1)\|$
$(1, 1)$	$(2, 3)$	$\|v(2, 1) - v(1, 1)\| \geqslant 0.5\|v(1, 3) - v(1, 1)\|$
$(2, 3)$	$(1, 1)$	$\|v(2, 3) - v(1, 3)\| \geqslant 0.5\|v(2, 3) - v(2, 1)\|$

Note that including values of (a, a') such that $\phi_i(a) = \phi_i(a')$ result in either trivial constraints (*e.g.*, $|v(2, 1) - v(1, 1)| \geqslant 0$) or in unsatisfiable constraints ($0 \geqslant |v(1, 3) - v(1, 1)|$).

8. Value function construction

We have given many representations of different types of preferences. It remains now to join them together by providing value functions consistent with partial orderings over the attribute space representing a given set of preferences.

In this section, we consider questions concerning construction of value functions that represent a satisfiable set of statements, including conditional qualitative *ceteris paribus* preferences, conditional marginal tradeoffs, and conditional attribute tradeoff statements. The questions we address are the following. Can one find a value function consistent with a consistent set of preference statements? Under what circumstances can we find one efficiently? Does this provide a unified and flexible framework for the expression of various types of preferences?

In light of these aims, we clarify that we do not provide a consistency checking procedure for a set of statements in LOPAT. We are merely constructing a satisfying value function, the existence of which implies that the statements in question are consistent; but the absence of which does not imply the statements are necessarily inconsistent. Furthermore, we aim to find only one function consistent with a set of statements, and not all such functions. We regard the problems of deducing what a set of preference specifications entail as a harder problem than merely constructing an example of their satisfiability. However, this example, corresponding to test cases, can sometimes help identify missing conditions on the desired preferences, and so provide some information of the preferences' scope and refinement.

Much of the following augments techniques presented in [32] for constructing an ordinal value function for *ceteris paribus* preferences over binary attributes. The presentation given here is updated to accommodate all of the types of preferences discussed so far. There are still some sections and results that require little modification from the original, and as such are skipped here. For the modified theorems we present here as analogues to the theorems of [32], a more complete treatment is found in [30].

8.1. Value functions from qualitative ceteris paribus preferences

In previous work [32], we considered how to create value functions from qualitative *ceteris paribus* statements. For consistent sets of *ceteris paribus* preferences over small domains, it is practical to use a method based on the ordering implied by the statements over the entire domain. The idea is to look at the preference graph for a set of such statements, then use some variant of a topological sort of that graph to rank-order the nodes of the graph in approximate desirability. Such an ordering is isomorphic to a value function consistent with the input preferences. This technique is generally too inefficient to be used on an entire domain, but works well for the constituent subvalue functions of an additive value function.

We describe in [32] a method for computing possible additive value decompositions of the domain attributes based on a set of qualitative *ceteris paribus* preferences. The method produces a collection of subsets of the attributes such that an additive value function using each subset as the domain of a subvalue function can be defined. Such a function is of the form

$$v(\vec{a}) = \sum_{C_i \in C} t_i v_{C_i}(\vec{a}),$$

for a cover C of the attributes A.

This method is based on a technique using the structure of qualitative *ceteris paribus* preference statements to determine which attributes are necessarily preferentially dependent. This allows us to make intelligent assumptions about the structure of the value functions that are consistent with the input preferences. To extend this method to our current situation we need to determine which attributes are preferentially independent given a set of marginal and attribute tradeoff preferences. We will show here that no such methods are possible, but we also show that no such methods are necessary.

To examine this question we examine two concerns in this section. Firstly, does a tradeoff statement between attributes G and attributes H imply that G and H are preferentially independent? The answer is no. Secondly, is it *possible* that G and H are preferentially independent? The answer is yes whenever the tradeoffs made between G and H are satisfiable.

It is not true that every tradeoff statement means there must be preferential independence between the related attributes. We present this result by demonstrating a counterexample.

Theorem 8.1 *(Preferential dependence). There exists $v \in [\![b \succsim_{\text{mt}} r : b']\!]_{\text{v}}$ with b, b' nonempty bundles over A such that v has $\sigma(b)$ preferentially dependent on $\sigma(b')$.*

Proof. We exhibit a simple example. Let $A = \{X, Y\}$. Let bundles $b = (X = 1), b' = (Y = 1)$, and a value function $v(x, y) = (rx + y - 1)^2$ that exhibits preferential dependence of X on Y. □

The opposite concern is also interesting. Is it always possible to create a linear additive value function (and therefore one that exhibits preferential independence) given any set of tradeoff preferences? In fact, one can construct a piecewise linear value function for any set of satisfiable preferences.

Theorem 8.2 *(Preferential independence). For any satisfiable set of unconditional marginal tradeoff statements T, there exists $v \in [\![T]\!]_{\text{v}}$ such that v is linear in each attribute in A.*

Proof. We are given some set T of unconditional marginal tradeoff preferences of the form $b \succsim_{\text{mt}} r : b'$ over some set of attributes A. These tradeoff statements, in turn, require that the partial derivatives of the value function satisfy conditions C of the form

$$\nabla v(\vec{a}) \cdot \phi(b) \geqslant r \nabla u(\vec{a}) \cdot \phi\left(b'\right)$$

for all $\vec{a}, \vec{a}' \in \vec{A}$ and for some particular bundles b, b'. These constraints C hold at all points $\vec{a}$ in the preference space. A solution to C is a value for $\nabla v(\vec{a})$. If C is satisfiable, the solution to constraints C is a vector of numbers, let it be $\vec{w}$, and this vector is the vector of partial derivatives $\nabla v(\vec{a})$. In this case there exists v with $\nabla v(\vec{a}) = \vec{w}$ for all $\vec{a} \in \vec{A}$. This function v satisfies the condition, and proves the theorem. □

One can extend this result to conditional tradeoff statements by considering piecewise linear value functions, but we do not do so formally here. Each condition divides the space into two parts, so by considering the regions defined by consistent

combinations of conditions, we choose a different linear value function for each such region. This is discussed in more detail in the following section.

These two results combine to show that we can use our previous algorithm for computing a generalized additive decomposition from a set of preferences without modification. Although we have more and different types of preferences in the current case, these two results show that only the *ceteris paribus* preferences are relevant to defining the additive decomposition of the attributes.

8.2. Value function construction

The algorithm of [32] for computing a generalized additive decomposition results in a partition of $A : C' = \{C'_1, C'_2, \ldots, C'_Q\}$ and corresponding set of sets of attributes $B' = \{B'_1, B'_2, \ldots, B'_Q\}$, such that each set of attributes C'_i is preferentially dependent on the attributes B'_i and preferentially independent of $A \backslash B'_i$. We define a cover $C = \{C_1, C_2, \ldots, C_Q\}$ such that $C_i = (C'_i \cup B'_i)$. Given the cover C, we construct an additive value function that is a linear combination of subvalue functions v_i, each subvalue a separate function of a particular C_i. We associate a *scaling parameter* t_i with each v_i such that the value of a model is

$$v(m) = \sum_{i=1}^{Q} t_i v_i(m). \tag{28}$$

We will argue that this is consistent with a set of preferences M where M is a set of qualitative *ceteris paribus* preferences statements, conditional marginal tradeoff statements, and conditional attribute tradeoffs in LOPAT. Given the cover C, we have two remaining tasks: to craft the subvalue functions v_i, and to choose the scaling constants t_i. We will accomplish these two tasks in roughly the following way. We partition input preferences into two sets, *ceteris paribus* and tradeoff preferences. We use *ceteris paribus* preferences to make subvalue functions and the tradeoff preferences provide linear constraints in a linear programming problem that sets the weights t_i. There is a method in [32] that further partitions *ceteris paribus* preferences into those used to make subvalue functions and those that provide additional linear constraints on the scaling parameters. We say no more about subvalue functions here, techniques from [32] apply without modification. To assign values to scaling parameters t_i, we will define a set of linear inequalities which constrain the variables t_i. The linear inequalities can then be solved using standard methods for solving linear programming problems. The solutions to the inequalities are the values for the scaling parameters t_i.

8.2.1. Adding tradeoffs to qualitative ceteris paribus preferences

We are going to construct three lists of linear inequalities, I, I', I'', that must be satisfied by choosing appropriate subvalue function parameters t_i. The constraints in list I will come from the given *ceteris paribus* tradeoff statements in M. These are computed by the methodology of [32], along with the subutility functions. I' will represent the constraints in the bundle tradeoffs, and I'', from the attribute tradeoffs.

Let M' and M'' be sets of marginal tradeoff and importance statements, respectively. For each marginal tradeoff statement $S \in M'$, where $S = x \succsim_{\text{mt}} r : y$, by Definition 5.1, we have constraints of the form

$$\sum_{i=1}^{Q} \sum_{A_j \in \sigma(x) \cap C_i} \frac{\partial v}{\partial A_i}(\vec{a}) \phi_j(x) \geqslant r \sum_{k=1}^{Q} \sum_{A_l \in \sigma(y) \cap C_k} \frac{\partial v}{\partial A_j}(\vec{a}) \phi_l(y) \tag{29}$$

and the partials can be computed from the subutility functions. Let the set of these constraints for all $S \in M'$ be the set of linear inequalities I'.

Then if we consider all the attribute statements $S' \in M''$, we will obtain additional linear inequalities bounding the tradeoff parameters t_i of the value function. For each attribute tradeoff statement $S' \in M''$ with $S' = G \succsim_{\text{ai}} r : H$, S' where x, y are the characteristic vectors for G, H:

$$\big(abs\big(\nabla v(\vec{a})\big) \cdot \vec{x}\big) \geqslant r\big(abs\big(\nabla v(\vec{a})\big) \cdot \vec{y}\big). \tag{30}$$

Let the set of these constraints for all $S' \in M''$ be the set of linear inequalities I''.

We will discuss conditional statements for conditions other than $True \Rightarrow S$ in the following subsection.

Any value function $v \in [\![I', I'']\!]_{\text{v}}$ is consistent with the tradeoff and attribute preferences the inequalities represent. We state the theorem here, which is true by definition of the preferences.

Theorem 8.3 *(General inequalities). Let M', M'' be sets of marginal tradeoff and attribute preferences in* LOPAT, *respectively. If the system of linear inequalities, $I' \cup I''$, has a solution, this solution corresponds to a value function v such that $v \in [\![M' \cup M'']\!]_{\text{v}}$.*

Proof. Follows from the definitions of marginal and attribute tradeoff preferences. □

We can solve the system of linear inequalities $I \cup I' \cup I''$ using any linear inequality solver. We note that it is possible to phrase this as a linear programming problem, and use any of a number of popular linear programming techniques to find scaling parameters t_i.

The number of inequalities in I' is determined by the number of the statements in the tradeoff preferences. Simple marginal tradeoff preferences $\mathit{True} \Rightarrow x \succsim_{\mathrm{mt}} r : y$ and attribute tradeoff statements $\mathit{True} \Rightarrow G \succsim_{\mathrm{ai}} r : H$ contribute one linear inequality each. (Note that if statements are given in disjunctive normal form, like $\mathit{True} \Rightarrow d \succsim_{\mathrm{mt}} r : d'$ then this statement results in a number of inequalities: $|d| * |d'|$.) Consequently, preferences M', M'' add a polynomial number of inequalities to inequality set I.

8.2.2. Piecewise linear value functions

When preferences are conditional, and hold at different regions in the outcome space $\vec{A}$, these various preferences imply different constraints on the value function in separate regions of $\vec{A}$. In general these constraints are not simultaneously satisfiable, and this necessitates different value functions for different regions of the space. In this way, the value function for one region can satisfy the constraints required of the value function in that region, and a value function for another region can satisfy the constraints for that region. The value function for the whole attribute space $\vec{A}$ is then a collection of different value functions for different subsets of the attribute space. Since each of these value functions are linear, the whole becomes a *piecewise linear value function*.

Definition 8.1 *(Piecewise value function).* U is a *piecewise value function* if U is a set of pairs (V, v_V) where V is a compound value proposition and $v_V : \vec{A} \to \mathbb{R}$ is a value function.

U assigns the value $v_V(\vec{x})$ to $\vec{x}$ when $\vec{x} \in [\![V]\!]_{\mathrm{a}}$. We write $U(\vec{x})$ for the value U assigns to $\vec{x}$. In this way, a piecewise value function is like a *switch* or *case* statement in a programming language; it selects which of several value functions to use based on the input.

When the value functions are linear in each of the attributes, different constraints on the value function are the result of conditional preferences. This is straightforward; different preferences can be conditioned on different regions of the space, using the conditional preferences provided in the language LOPAT.

The conditional tradeoffs expressed in LOPAT are binary conditions. In some region of the attribute space, the preference holds, and in the remainder of the attribute space, the preference does not apply. Thus, given k conditional statements, each with independent conditions, we have as many as 2^k separate divisions of the attribute space with different preferences holding in each division.

Given k conditional tradeoffs, we can define the 2^k subsets of the attribute space by the intersection of a unique subset of the k conditions. Let W be the set of condition statements corresponding to M', then each $w \in W$ is a separate compound value statement. Any subset $V \subseteq W$ holds on a region of the attribute space defined by $\bigwedge\{w \mid w \in V\}$.

For each subset V of W, the set of tradeoff preferences that hold over the corresponding space is just that which correspond to the conditions. We state this in the following theorem.

Theorem 8.4 *(Space conditions). Given a set of conditional preferences M, with corresponding conditions W, each subset $V \subseteq W$ defines a region of the attribute space $\bigwedge\{w \mid w \in V\}$ where preferences corresponding to V in M hold.*

Proof. This theorem follows directly from the definition of conditional preference. □

Note that if condition $\bigwedge\{w \mid w \in V\}$ is unsatisfiable, then V describes no portion of the attribute space, and so requires no further consideration.

This theorem defines the regions of the space where different constraints hold. However, just because these regions have different constraints, it does not mean that the constraints are mutually exclusive or unsatisfiable. Given a set of conditions W and two subsets, $V \subset W$, $V' \subset W$, if the value function constraints holding over $V \cup V'$ are satisfiable by some value function v', then this value function can be used for $V \cup V'$.

In the presence of conditional tradeoff preferences, we proceed as follows. For a set of conditional and unconditional tradeoff and attribute preferences $\{M' \cup M''\}$, consider the set W of conditions on those preferences. For each subset V of W, let J be the set of preferences from $\{M' \cup M''\}$ conditioned by V. Then for preferences $J \cup M$, we can construct sets of linear inequalities as discussed in the preceding section; the solution to this set of linear inequalities gives us a value function. This value function, in turn, is the value function for the subset of the attribute space indicated by $\bigwedge\{w \mid w \in V\}$. In this way, we construct separate value functions for different sections of the attribute space.

The methods of dealing with different conditions on tradeoffs here can be computationally difficult. It is possible that borrowing techniques from constraint satisfaction literature would be efficacious here.

8.3. A detailed example

Let us consider an example and how it can fit into the frameworks mentioned above. Suppose we are going out to eat in Boston, and need to pick a restaurant. We consider the food, the wine, the atmosphere, the time to get there, and the time

spent waiting once at the restaurant. Usually, in Boston, restaurants are crowded, and since we do not have reservations expedience can be a serious concern. Let $\vec{A} = \langle m, w, a, tt, wt \rangle$ for *meal, wine, atmosphere, travel time*, and *wait time*. Then let *meal* have two values: $b_1 = meat$, $b_2 = fish$; *wine* have two values: $w_1 = red$, $w_2 = white$; and *atmosphere* have three values: $a_1 = bland$, $a_2 = gaudy$, $a_3 = quiet$. *Travel time* and *wait time* will be measured in minutes. We now state some simple *ceteris paribus* preferences:

	Variable	Preferences
p_1	m	$fish \succ meat$
p_2	w	$fish \wedge white \succ fish \wedge red$
p_3	w	$meat \wedge red \succ meat \wedge white$
p_4	a	$quiet \succ gaudy$
p_5	a	$gaudy \succ bland$

These preferences mean that we prefer fish to meat. Preferences p_2 and p_3 mean that our preference for wine depends on the main course. The remaining two preferences establish an order over the possible restaurant atmospheres.

Travel time and *wait time* are numeric attributes where less is better. We state tradeoffs about these attributes: $wt \succ_{\text{ai}} 1.5 : tt$, which indicates that is roughly 50% more annoying to wait at the restaurant than to travel to it. These preferences have laid the groundwork for a tradeoff between groups of attributes: $\{m, w, a\} \succ_{\text{ai}} 10 : \{tt, wt\}$. Suppose someone in the dinner party asserts that $\{(wt = 15), (tt = 10)\} \succ_{\text{mt}} \{(m = meat), (w = white)\}$, meaning that a moderate delay is preferable to having white wine with dark meat. This last preference (p_8, below) is somewhat fanciful but illustrates the tradeoff between delay and meal quality. We then have these three tradeoff preferences:

	Preferences
p_6	$wt \succ_{\text{ai}} 1.5 : tt$
p_7	$\{m, w, a\} \succ_{\text{ai}} 10 : \{tt, wt\}$
p_8	$\{(wt = 15), (tt = 10)\} \succ_{\text{mt}} \{(m = meat), (w = white)\}$

Preferences p_6–p_8 imply the following conditions on the partial derivatives of the value function:

	Conditions
s_1	$\lvert \frac{\partial v}{\partial wt}(\vec{x}) \rvert > 1.5 \lvert \frac{\partial v}{\partial tt}(\vec{x}) \rvert$
s_2	$\lvert \frac{\partial v}{\partial m}(\vec{x}) \rvert + \lvert \frac{\partial v}{\partial w}(\vec{x}) \rvert + \lvert \frac{\partial v}{\partial a}(\vec{x}) \rvert \geqslant 10(\lvert \frac{\partial v}{\partial tt}(\vec{x}) \rvert + \lvert \frac{\partial v}{\partial wt}(\vec{x}) \rvert)$
s_3	$15 \frac{\partial v}{\partial wt}(\vec{x}) + 10 \frac{\partial v}{\partial tt}(\vec{x}) > \frac{\partial v}{\partial m}(\vec{x}) + \frac{\partial v}{\partial w}(\vec{x})$

We return to the ordinal attributes, and can now construct subvalue functions for each of the attributes, or, in this case, for each preferentially independent set of attributes. Here attribute w is preferentially dependent on attribute m, so following the system of [32], we generate one subvalue function for $\{m, w\}$, one subvalue function for a, one for tt, and one for wt. For the qualitative attributes, we can specify their subvalue functions simply by assigning numbers to each of the qualitative alternatives of each attribute, and using these assignments as the output of the subvalue function for these attributes, respectively. To continue this example, let us assign subvalue functions as follows:

Subvalue	Value	Subvalue	Value
$v_{\{m,w\}}(fish, white)$	3	$v_a(quiet)$	3
$v_{\{m,w\}}(fish, red)$	2	$v_a(gaudy)$	2
$v_{\{m,w\}}(meat, red)$	2	$v_a(bland)$	1
$v_{\{m,w\}}(meat, white)$	1		

For numeric attributes wt and tt, we can choose a simple linear subvalue function. We take $v_{wt} = -wt$ and $v_{tt} = -tt$.

The subvalue functions are now known, and the form of the value function (additive) is known, that is, the value function is of the form: $v(\vec{a}) = \sum_i t_i v_i(\vec{a})$. But before we can use the inequalities involving the partial derivatives of the value function, we must assign value functions, ρ, that take the discrete domains to numbers. We proceed in the most straightforward way, and assign values as follows:

Value function	Value	Value function	Value
$\rho_m(fish)$	2	$\rho_a(quiet)$	3
$\rho_m(meat)$	1	$\rho_a(gaudy)$	2
$\rho_w(white)$	2	$\rho_a(bland)$	1
$\rho_w(red)$	1		

In Section 5.4 we argued that the slope of a linear subvalue function can be used as the partial derivative of that subvalue function with respect to the value of our ordinal attributes, and therefore we can compute the partial derivatives of the value function and simplify conditions s_1, s_2, and s_3. We must consider that we have different partial derivatives at different vectors in $\vec{A}$. In particular, when we evaluate the partials of v with respect to m and to w, we let x be in the domain of m, and y be in the domain of w. In these cases we have

$$\frac{\partial v}{\partial w}(x, y) = t_{\{m,w\}}\big(v_{\{m,w\}}(x, \mathit{white}) - v_{\{m,w\}}(x, \mathit{red})\big)/\big(\rho(\mathit{white}) - \rho(\mathit{red})\big),$$
$$\frac{\partial v}{\partial m}(x, y) = t_{\{m,w\}}\big(v_{\{m,w\}}(\mathit{fish}, y) - v_{\{m,w\}}(\mathit{meat}, y)\big)/\big(\rho(\mathit{fish}) - \rho(\mathit{meat})\big).$$

Note that the other partial derivatives are straightforward (following from Lemmas 5.1 and 7.1). Thus, using the above, when we fix $m = \mathit{fish}$ when computing $\frac{\partial v}{\partial w}(w)$ and $w = \mathit{white}$ when computing $\frac{\partial v}{\partial m}(m)$ we have

$$|2t_{\{m,w\}}| + |t_{\{m,w\}}| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

Similarly if we fix $m = \mathit{meat}$ and $w = \mathit{white}$ then

$$|2t_{\{m,w\}}| + |-t_{\{m,w\}}| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big),$$

and $m = \mathit{fish}$ with $w = \mathit{red}$ gives

$$0 + |t_{\{m,w\}}| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

Finally fixing $m = \mathit{meat}$ and $w = \mathit{red}$ gives this constraint

$$0 + |-t_{\{m,w\}}| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

Some of these constraints are identical because of the absolute value functions, so we can collect cases into two, and have

$$3t_{\{m,w\}} + t_a \geqslant 10(t_{tt} + t_{wt}) \quad w = \mathit{white},$$
$$t_{\{m,w\}} + t_a \geqslant 10(t_{tt} + t_{wt}) \quad w = \mathit{red}.$$

When computing the constraints implied by condition s_3, we get slightly different results for $m = \mathit{meat}$ and for $m = \mathit{fish}$. These appear with the other constraints on the parameters of the value function in the table below:

	Constraint	
c_1	$3t_{\{m,w\}} + t_a \geqslant 10(t_{tt} + t_{wt})$	$w = \mathit{white}$
c_2	$t_{\{m,w\}} + t_a \geqslant 10(t_{tt} + t_{wt})$	$w = \mathit{red}$
c_3	$t_{wt} \geqslant 1.5t_{tt}$	
c_4	$15t_{wt} + 10t_{tt} > t_{mw}$	$m = \mathit{fish}$
c_5	$15t_{wt} + 10t_{tt} > -t_{mw}$	$m = \mathit{meat}$

These systems of linear inequalities can be solved for the different cases, in principle resulting in piece-wise linear value functions. In this case, since constraint c_1 follows from constraint c_2, and c_5 from c_4 for positive t's, there is no need to have different functional forms of the value function based on different values of the w attribute. Therefore, a solution for this construction is $t_{m,w} = 1$, $t_a = 50$, $t_{wt} = 3$, and $t_{tt} = 2$.

Thus a value function for this example is

$$v(\vec{x}) = v_{m,w}(\vec{x}) + 50v_a(\vec{x}) + 3v_{wt}(\vec{x}) + 2v_{tt}(\vec{x}).$$

Such a value function can then be used to make decisions between different alternatives. Consider the three hypothetical restaurants in the following table:

	A	B	C
wt	-15	-45	-25
tt	-15	-5	-60
a	*loud*	*simple*	*elegant*
m, w	*fish, white*	*fish, white*	*meat, red*
$v()$	-22	-42	-43

In such a situation restaurant A is preferable.

8.4. A sound algorithm for value function construction

In the preceding, we have described several parts of an algorithm for computing with qualitative *ceteris paribus* preference statements, marginal tradeoff preferences, and conditional attribute tradeoffs. This has given us enough tools to accomplish our goal: generating a value function consistent with various types of input preferences. We will outline the algorithm for such here.

The algorithm takes as input a set M of qualitative *ceteris paribus* preference statements, of marginal tradeoff statements, and of attribute statements, in the language LOPAT, and parameters for building subvalue functions (from [32]). Simply speaking, a generalized additively independent cover of the attributes is computed. Then the preferences are interpreted as constraints on the partial derivatives of the value function. Next, the conditions on the preferences are considered, and a partition of the attribute space is created. Then the constraints are solved using linear programming; giving values for the parameters of the additive value function. The algorithm finally outputs a piecewise linear value function U, a set of pairs (V, v_V) such that each value functions v_V is consistent with the input preferences at a different region V of the attribute space $\vec{A}$.

Some remarks must be said about the failure conditions of this algorithm, by which we mean, the algorithm encountering error conditions that cause it to stop. First of all, the algorithm may fail because the steps concerning merely the qualitative *ceteris paribus* preferences can fail; these are heuristic methods. As we discuss in [32], consistent *ceteris paribus* preferences can always be represented by a trivial value function; one that orders each outcome according to the preorder implied by the preferences, but this gains none of the advantages of a generalized additive decomposition value function.

Secondly, a set of tradeoff preferences cannot be considered to be consistent or inconsistent without knowledge of the partial derivatives of the value function. The partial derivatives of the value function, in this case, are determined by the generalized additive decomposition of the attribute space. Thus we cannot know with certainty before the algorithm determines the additive decomposition of the attribute space if the tradeoff preferences are consistent or not.

With these shortcomings in mind, we must consider this algorithm heuristic. There is always the possibility of conflicting preferences leading to no solution. However, when the algorithm finds a solution, it is guaranteed to represent the input preferences faithfully. This algorithm, therefore, fulfills its main purpose: it illustrates that tradeoff preferences can in principle be combined with qualitative *ceteris paribus* preferences of the type presented in Section 6. Indeed, we show in the next section that tradeoff preferences can be combined with the CP-net representation of qualitative *ceteris paribus* preferences.

The soundness of this algorithm can be proven by reference to the preceding theorems of this article, and to those appearing in [30].

Theorem 8.5 *(Soundness). Given a set of* ceteris paribus*, marginal tradeoff, and attribute preferences M, if the above-outlined algorithm produces a piecewise linear value function U, then $U \in [\![M]\!]_{\mathrm{V}}$.*

A proof appears in [30].

9. Quantitative tradeoffs and CP-nets

In general, the tradeoffs and importance preference statements described in this article generate linear constraints on the parameters of additive value functions. These constraints can be easily integrated with any preference or utility estimation system that uses linear inequalities to constrain the parameters of possible value functions. And since linear models of utilities are so common in practice, the system we have proposed should be widely applicable. In the previous section we showed how to combine tradeoff preferences with the method of [32]. In the present section we show how to combine the linear inequalities generated from our preference tradeoff statements with the CP-nets system. We stress that integration with these two systems are merely representative of other possible integrations.

Methods proposed by Brafman, Domshlak, and Kogan [8] take CP-nets [5] and TCP-nets [9] and generate a value function consistent with the order implied by the CP-net or TCP-net. These methods use qualitative *ceteris paribus* preference as their input, and output a generalized additive-independent ordinal value function. When we consider the system we have presented in this article alongside the systems based on CP-nets, we find there are differences of expressiveness and tractability. CP-nets place restrictions on the form of the preference statements, and make independence relationships explicit; the methodology we have presented allows arbitrary qualitative *ceteris paribus* preferences and infers independence from the statements. The restrictions on CP-nets allow strong tractability results. Acyclic TCP-nets, for example, always allow efficient value function construction [7]. Such differences mean that both CP-nets and the preference statements presented herein may be appropriate in different situations.

We now demonstrate that the various quantitative tradeoffs we have developed fit easily together with CP-nets. And, following that exposition we will briefly outline a correspondence between our tradeoffs and the tradeoffs of TCP-nets [9].

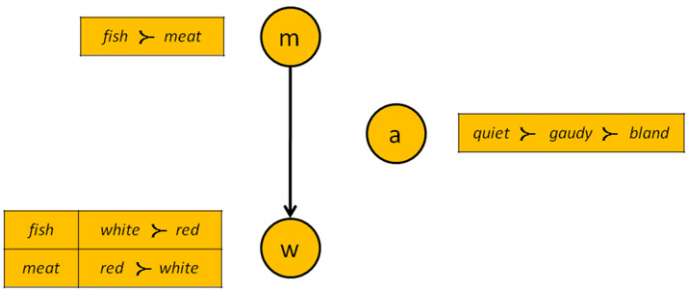

Fig. 1. CP-net for preferences p_1–p_5.

9.1. Adding quantitative tradeoffs to CP-nets

To add quantitative tradeoffs to CP-nets we require two things of the value function; one, that it should be a generalized additive value function and two, that it should have linear subvalue functions.

Using the methods of [8,7] to compile a CP-net into a value function, commits us to using a generalized additive value function. We can force the subvalue functions (termed "factors" in that source) of this value function to be linear in their input by adding additional inequalities to the system of linear equations that generates the value function. These conditions assure that our tradeoff statements in LOPAT can be easily added to the CP-net.

The system of linear inequalities constructed by [7, Section 3] has one variable per input per subvalue function, so we can add additional linearizing inequalities assuring that the output of the subvalue function for X is linear in X. The proper ordering among values of X can be found by considering the CP-Family of X [7, Section 3], and computing a different linear program for each possible ordering consistent with the CP-Family. This is a locally-exponential addition to the complexity of value function construction, so the problem remains in P when the exponents are bounded by a constant.

After assuring the subvalue functions are linear in their input, it is simple to solve an additional system of linear inequalities which constrain the tradeoff ratios between subvalue functions. This new problem has one variable for each subvalue function, representing the weight given to it in the generalized additive value function, and one or more inequalities for each tradeoff statement $S \in$ LOPAT. Each tradeoff statement results in linear constraints on the tradeoff parameters of the value function, but may result in different constraints over different areas of the domain of the value function. This is the case when the preferences over one attribute, and thus partial derivatives with respect to that attribute, switch with the values assumed by a different attribute. Such is the normal case of utility dependence between attributes. In these cases, the value function will be a piecewise linear one, having different functional forms for different parts of its domain.

9.2. A CP-net example

We previously considered an example involving choosing a restaurant in Boston. We will work through the same example here, again, but this time in a CP-net framework. This illustrates the differences between the CP-net formalism and the methods presented earlier in this article.

We again choose $\vec{A} = \langle m, w, a, tt, wt \rangle$ for *meal*, *wine*, *atmosphere*, *travel time*, and *wait time*, just as in the previous example.

The simple *ceteris paribus* preferences we used before, p_1–p_5 can be used to construct a CP-net.

In a CP-net for these preferences, we have to consider which attributes are preferentially dependent. In this case only w depends on m so we draw the CP-net as shown in Fig. 1.

We likewise use the same tradeoff preferences (p_6–p_8) from our previous example. These tradeoffs imply the same constraints as before, but here they are an addendum to the CP-net framework: we will keep them aside for now.

To compute a value function for the CP-net we must solve a system of linear inequalities, of the form of inequality 1 in [8]. In this case, it results in the following linear inequalities:

	Preferences
e_1	$v_a(quiet) > v_a(gaudy)$
e_2	$v_a(gaudy) > v_a(bland)$
e_3	$v_w(fish, white) > v_w(fish, red)$
e_4	$v_w(meat, red) > v_w(meat, white)$
e_5	$v_m(fish) + v_w(fish, white) > v_m(meat) + v_w(meat, white)$
e_6	$v_m(fish) + v_w(fish, red) > v_m(meat) + v_w(meat, red)$

We can then add linearizing inequalities to the system, forcing $3k_a v_a(quiet) \geqslant 2k_a v_a(gaudy) \geqslant k_a v_a(bland)$ and $3k_a v_a(quiet) \leqslant 2k_a v_a(gaudy) \leqslant k_a v_a(bland)$, using a new variable k_a. We make similar inequalities for v_w and v_m. We require these

additional inequalities to force the subvalue functions for each attribute in the CP-net to be linear; this simplifies our methodology. A solution is as follows:

Subvalue	Value	Subvalue	Value
$v_m(\mathit{fish})$	2	$v_w(\mathit{fish}, \mathit{white})$	4
$v_m(\mathit{meat})$	1	$v_w(\mathit{fish}, \mathit{red})$	3
$v_a(\mathit{quiet})$	3	$v_w(\mathit{meat}, \mathit{red})$	2
$v_a(\mathit{gaudy})$	2	$v_w(\mathit{meat}, \mathit{white})$	1
$v_a(\mathit{bland})$	1	k_a	1
k_m	1	k_w	1

For attributes *wt* and *tt*, we use these linear subvalue functions : $v_{wt} = -wt$ and $v_{tt} = -tt$.

The partial derivatives of the value function are different in the CP-nets example than those in our previous example. We compute the partial derivatives of v with respect to each attribute, paying special attention to the formulae for the partials of m and of w. For $x \in \{\mathit{fish}, \mathit{meat}\}$ and $y \in \{\mathit{white}, \mathit{red}\}$, we have

$$\frac{\partial v}{\partial w}(x, y) = t_w\big(v_w(x, \mathit{white}) - v_w(x, \mathit{red})\big)/\big(\rho(\mathit{white}) - \rho(\mathit{red})\big),$$
$$\frac{\partial v}{\partial m}(x, y) = t_w\big(v_w(\mathit{fish}, y) - v_w(\mathit{meat}, y)\big)/\big(\rho(\mathit{fish}) - \rho(\mathit{meat})\big) + t_m\big(v_m(\mathit{fish}) - v_m(\mathit{meat})\big)/\big(\rho(\mathit{fish}) - \rho(\mathit{meat})\big).$$

Thus, when we fix $m = \mathit{fish}$ when computing $\frac{\partial v}{\partial w}(w)$ and $w = \mathit{white}$ when computing $\frac{\partial v}{\partial m}(m)$ we have

$$|t_w| + |3t_w| + |t_m| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

Similarly if we fix $m = \mathit{meat}$ and $w = \mathit{white}$ then

$$|-t_w| + |3t_w| + |t_m| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big),$$

and $m = \mathit{fish}$ with $w = \mathit{red}$ gives

$$|t_w| + |-t_w| + |t_m| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

Finally fixing $m = \mathit{meat}$ and $w = \mathit{red}$ gives this constraint

$$|-t_w| + |-3t_w| + |t_m| + |t_a| \geqslant 10\big(|-t_{tt}| + |-t_{wt}|\big).$$

As we did with the constraints in the last example, we can again collect cases into two, and have

$$4t_w + t_m + t_a \geqslant 10(t_{tt} + t_{wt}) \quad w = \mathit{white},$$
$$2t_w + t_m + t_a \geqslant 10(t_{tt} + t_{wt}) \quad w = \mathit{red}.$$

These constraints can then be collected with all other constraints on the parameters of the value function. We then have the following constraints on the parameters of the value function:

	Constraint	
c_1	$4t_w + t_m + t_a \geqslant 10(t_{tt} + t_{wt})$	$w = \mathit{white}$
c_2	$2t_w + t_m + t_a \geqslant 10(t_{tt} + t_{wt})$	$w = \mathit{red}$
c_3	$t_{wt} \geqslant 1.5t_{tt}$	
c_4	$15t_{wt} + 10t_{tt} > t_w$	$m = \mathit{fish}$
c_5	$15t_{wt} + 10t_{tt} > -t_w$	$m = \mathit{meat}$

The only remaining step is to solve this system of linear inequalities for the tradeoff parameters t. As in the previous example, constraint c_1 follows from constraint c_2, so there is no need to have different functional forms of the value function based on different values of the w attribute. A solution to the CP-net system of inequalities is $t_m = 1$, $t_w = 1$, $t_a = 50$, $t_{wt} = 3$, and $t_{tt} = 2$.

Thus a value function for the CP-net is

$$v(\vec{a}) = v_m(\vec{a}) + 50v_a(\vec{a}) + v_w(\vec{a}) + 3v_{wt}(\vec{a}) + 2v_{tt}(\vec{a}).$$

9.3. Other types of CP-nets

It should be clear from the preceding discussion that the qualitative *ceteris paribus* preferences in a CP-net can be transposed with little difficulty to *ceteris paribus* preferences in LOPAT.

The conditional preferences of a CP-net are of the form: $c \Rightarrow x_1 \succ x_2$ for an attribute X and values $x_1, x_2 \in D(X)$ and some conditions in c such that $X \notin \sigma(c)$. In the formalism of the CP-network, the conditions c involve the parents of attribute X according to the network topology of the network; in a *ceteris paribus* preference in LOPAT c is merely a set of arbitrary conditions. We state that a conditional preference from a CP-net $c \Rightarrow x_i \succ x_j$ is equivalent to a *ceteris paribus* preference $c \Rightarrow x_i \succ_{\text{cp}} x_j$, although we leave a proof of this to future work.

The tradeoffs in TCP-net are conditional qualitative tradeoffs, wherein a *selector set* of attributes Z determine the particular tradeoff between two other attributes X, Y. In [9], this is written $\mathcal{RI}(X, Y|\mathbf{Z})$, meaning that the *Relative Importance* of X and Y is conditional on values taken by attributes in Z. When the relative importance of X is greater than that of Y, then written $X \rhd_{\mathbf{z}} Y$ any (small) increase in X is preferable to any (possibly large) increase in Y. In these cases, there is a preference order over the domain of X expressed in the CP-net portion of the TCP-net, and similarly with Y. In LOPAT, we can express a structurally similar tradeoff by $z_1 \Rightarrow X \succ_{\text{ai}} r : Y$ together with $z_2 \Rightarrow Y \succ_{\text{ai}} r : X$, where z_1 indicates values of Z for which $X \rhd Y$ and z_2 indicates values of Z where $Y \rhd X$. We have only to choose a very small value of r such that it approximates the dominance of $X \rhd Y$; we must choose r such that r times $\max i, j(v_Y(y_i) - v_Y(y_j))$ is smaller than $\min_{i,j}(v_X(x_i) - v_X(x_j))$. Again, we leave a proof of such equivalence, as well as selection methods for r, to future work.

While the forgoing may lead to possible advantages of mixing representations, it may be advantageous to convert a CP-net or TCP-net to preferences in LOPAT if the CP-net or TCP-net cannot be fully elicited or specified.

10. Conclusions

We have presented novel methods for enriching systems of qualitative *ceteris paribus* preferences with quantitative tradeoffs of various types over multiple attributes. These preference systems can then be compiled into quantitative value functions using modifications of existing techniques. Our work here has provided an important extension to both the systems of [31] and [5].

The main contribution of this article has been the representation of tradeoffs as constraints on the partial derivatives of the value function. We have demonstrated that this general approach to tradeoff semantics is broad enough to cover (1) tradeoffs between particular values of attributes, (2) importance constraints between sets of attributes, (3) multiattribute tradeoffs of each preference type considered, and (4) tradeoffs over discrete and continuous attributes.

We also obtained numerous results relating these types of constraints to each other and to earlier notions. Our semantics for multiattribute marginal tradeoffs, applied to pairs of individual attributes, reduces to the notion of marginal rate of substitution familiar in economics, and our semantics for discrete marginal tradeoffs allows for rerepresentation of the qualitative *ceteris paribus* preferences of Doyle, Wellman and Shoham [14]. Among our new concepts, we show that discrete marginal tradeoffs reduce to continuous bundle tradeoffs, that discrete attribute tradeoffs reduce to continuous attribute tradeoffs, and that discrete marginal tradeoffs reduce to discrete attribute tradeoffs when the bundles involved are equivalent to the characteristic vectors of the sets of attributes related in the attribute tradeoff.

These results show that one can combine all of these tradeoff preferences into a single methodology, together with qualitative *ceteris paribus* preferences, for computing a value function representing preference statements of all forms. Furthermore, these combination methods can function with however many or few preferences happen to be available, these preferences can be over any attributes, and there need be no explicit preferential independence or preferential dependence given with the preferences. These are all significant departures from the assumptions underlying traditional decision analysis.

Our representation of tradeoff statements as constraints on the partial derivatives of the value function is novel, and it raises many new questions for further research, both in its own right, and in interaction with *ceteris paribus* preference statements.

The basis of our interpretation of attribute importance tradeoffs is the ratio of gradients of the value function. Our use of the magnitude of the gradient of the value function to calibrate the relative utilities of two subspaces is a heuristic choice, and one could employ other measures instead, such as some kind of average-case improvement measure that tries to capture the average case outcome.

Some problems for further investigation concern the interaction of complex constraints on derivatives over nonlinear subvalue functions, especially when constructing piecewise linear value functions and when considering preferentially dependent attributes and preference reversals. When do preference reversals result in mutually incompatible constraints? Do preference reversals require negative values of the subvalue function scaling parameters? Can one characterize the types of subvalue functions that require piecewise linear solutions and those that do not?

One might investigate the integration of our partial-derivative based tradeoff preferences with other preference reasoning systems. There could be a stronger integration of our results with TCP-nets. One might seek a combination with answer-set solvers that incorporate preference representations [11]. A combination with a machine-learning system based on an SVM architecture [12] might be straightforward, but the value of this combination of techniques would require assessment.

Acknowledgements

We thank Peter Szolovits, Patrick Winston, and Howard Shrobe for comments, encouragement, and support. We also thank the anonymous reviewers of earlier versions of this manuscript for their helpful and encouraging comments, and thank one reviewer in particular for suggestions regarding the *guns* and *bullets* example. Michael McGeachie is grateful for support from DARPA, a training grant from the National Library of Medicine, and the Pfizer Corporation. Jon Doyle thanks SAS Institute for its endowment that supports his work.

References

[1] F. Bacchus, A. Grove, Graphical models for preference and utility, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1995, pp. 3–19.
[2] F. Bacchus, A. Grove, Utility independence in a qualitative decision theory, in: Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning, Morgan Kaufmann, 1996, pp. 542–552.
[3] J. Beattie, J. Baron, Investigating the effect of stimulus range on attribute weight, Journal of Experimental Psychology: Human Perception and Performance 17 (2) (1991) 571–585.
[4] C. Boutilier, F. Bacchus, R.L. Brafman, UCP-networks: A directed graphical representation of conditional utilities, in: Proceedings of Seventeenth Conference on Uncertainty in Artificial Intelligence, Seattle, 2001.
[5] C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, D. Poole, Cp-nets: A tool for representing and reasoning about conditional ceteris paribus preference statements, Journal of Artificial Intelligence Research 21 (2004) 135–191.
[6] C. Boutilier, R.I. Brafman, H.H. Hoos, D. Poole, Reasoning with conditional ceteris paribus preference statements, in: Proceedings of Uncertainty in Artificial Intelligence (UAI-99), 1999.
[7] R.I. Brafman, C. Domshlak, Graphically structured value-function compilation, Artificial Intelligence Journal 172 (2–3) (2008) 325–349.
[8] R.I. Brafman, C. Domshlak, T. Kogan, Compact value-function representations for qualitative preferences, in: Proceedings of Uncertainty in Artificial Intelligence (UAI'04), AUAI Press, Arlington, VA, 2004, pp. 51–59.
[9] R.I. Brafman, C. Domshlak, S.E. Shimony, On graphical modeling of preference and importance, Journal of Artificial Intelligence Research 25 (2006) 389–424.
[10] D. Braziunas, C. Boutilier, Local utility elicitation in GAI models, in: Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence, Edinburgh, 2005, pp. 42–49.
[11] G. Brewka, Answer sets and qualitative decision making, Synthese 146 (2004) 171–187.
[12] C. Domshlak, T. Joachims, Unstructuring user preferences: Efficient non-parametric utility revelation, in: Proceedings of the Twenty-first Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), AUAI Press, 2005, p. 169.
[13] J. Doyle, M. McGeachie, Exercising qualitative control in autonomous adaptive survivable systems, in: Revised papers from the Second International Workshop on Self-Adaptive Software (IWSAS 2), May 2001, Springer Verlag, Berlin, 2003, pp. 158–170.
[14] J. Doyle, Y. Shoham, M.P. Wellman, A logic of relative desire (preliminary report), in: Z. Ras (Ed.), Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems, in: Lecture Notes in Computer Science, Springer Verlag, Berlin, 1991, pp. 158–170.
[15] D. Dubois, H. Prade, Possibilistic logic: A retrospective and prospective view, Fuzzy Sets and Systems 144 (2004) 3–23.
[16] W. Edwards, F.H. Barron, Smarts and smarter: Improved simple methods for multiattribute utility measurement, Organizational Behavior and Human Decision Making 60 (1994) 306–325.
[17] Y. Engel, M.P. Wellman, CUI networks: A graphical representation for conditional utility independence, Journal of Artificial Intelligence Research 31 (2008) 83–112.
[18] G.W. Fischer, Range sensitivity of attribute weights in multiattribute value models, Organizational Behavior and Human Decision Processes 62 (3) (1995) 252–266.
[19] P.C. Fishburn, Decision and Value Theory, John Wiley & Sons, New York, 1964.
[20] C. Gonzales, P. Perny, GAI networks for utility elicitation, in: Proceedings of the Ninth International Conference on Knowledge Representation and Reasoning (KR'04), 2004.
[21] C. Gonzales, P. Perny, GAI networks for decision making under certainty, in: IJCAI'05 – Workshop on Advances in Preference Handling, 2005.
[22] W.M. Gorman, The structure of utility functions, Review of Economic Studies 35 (1968) 367–390.
[23] S.O. Hansson, A new semantical approach to the logic of preference, Erkenntnis 31 (1989) 1–42.
[24] J.M. Henderson, R.E. Quandt, Microeconomic Theory: A Mathematical Approach, third ed., McGraw–Hill, New York, 1980.
[25] R. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley and Sons, New York, 1976.
[26] R.L. Keeney, Value-Focused Thinking: A Path to Creative Decision Making, Harvard University Press, Cambridge, MA, 1992.
[27] D.H. Krantz, R.D. Luce, P. Suppes, A. Tversky, Foundations of Measurement, Academic Press, New York, 1971.
[28] P. La Mura, Y. Shoham, Expected utility networks, in: Proceedings of 15th Conference on Uncertainty in Artificial Intelligence, 1999, pp. 366–373.
[29] J. Lang, L. Van Der Torre, E. Weydert, Utilitarian desires, Autonomous Agents and Multi-Agent Systems 5 (3) (2002) 329–363.
[30] M. McGeachie, Local geometry of multiattribute preference tradeoffs, Ph.D. thesis, Tech. rep. MIT-CSAIL-TR-2007-029, Massachusetts Institute of Technology, Cambridge, MA, 2007.
[31] M. McGeachie, J. Doyle, Efficient utility functions for ceteris paribus preferences, in: AAAI Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, 2002.
[32] M. McGeachie, J. Doyle, Utility functions for ceteris paribus preferences, Computational Intelligence 20 (2) (2004) 158–217.
[33] H. Raiffa, Decision Analysis: Introductory Lectures on Choices Under Uncertainty, Addison–Wesley, Reading, MA, 1968.
[34] L.J. Savage, The Foundations of Statistics, John Wiley & Sons, New York, 1954.
[35] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1944.
[36] B. von Stengel, Decomposition of multiattribute expected-utility functions, Annals of Operations Research (1988).
[37] G.H. von Wright, The Logic of Preference: An Essay, Edinburgh University Press, Edinburgh, 1963.
[38] M. Wellman, J. Doyle, Preferential semantics for goals, in: T. Dean, K. McKeown (Eds.), Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1991, pp. 698–703.
[39] M.P. Wellman, J. Doyle, Modular utility representation for decision-theoretic planning, in: Proceedings of the First International Conference on AI Planning Systems, 1992, pp. 236–242.
[40] N. Wilson, Extending CP-nets with stronger conditional preference statements, in: Proceedings of Nineteenth National Conference on AI (AAAI'04), 2004, pp. 735–741.