Simultaneous Elicitation of Preference Features and Utility

Craig Boutilier
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
cebly@cs.toronto.edu

Kevin Regan
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
kmregan@cs.toronto.edu

Paolo Viappiani
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
paolo@cs.toronto.edu

Abstract

Most frameworks for utility elicitation assume a predefined set of features over which user preferences are expressed. We consider utility elicitation in the presence of *subjective* or *user-defined features*, whose definitions are not known in advance. We treat the problem of learning a user's feature definition as one of *concept learning*, but whose goal is to learn only enough about the concept definition to enable a good decision to be made. This is complicated by the fact that user utility is unknown. We describe computational procedures for identifying optimal alternatives w.r.t *minimax regret* in the presence of both utility and concept uncertainty; and develop several heuristic query strategies that focus simultaneously on reduction of *relevant* concept and utility uncertainty.

Introduction

Assessing the preferences of users is a critical component in any decision support or recommender system. Preference assessment allows recommendations to be tailored to the needs and desires of a particular user (Burke 2002; Viappiani, Faltings, and Pu 2006; Boutilier et al. 2006; Salo and Hämäläinen 2001). In most work on adaptive utility elicitation, one assumes the existence of a set of *universal* or *catalog features* over which user preferences are specified. For instance, in product configuration, preferences are articulated in terms of product features and specifications (e.g., color, engine size, fuel economy, available options, etc. in the case of a car). However, users can exhibit significant variation in the features over which their preferences are most *naturally* expressed; and these may not be present among the set of catalog features. For instance, in the automotive domain, different users may be concerned about the "degree of safety" of a car, but each may have different notions of safety in mind (e.g., a driver with a young family may define safety in terms of tires, air bags, child restraints, etc., while a high-performance driver refers to the braking system, roll bars, etc.). Furthermore, the user-specific *subjectivity* of safety prevents one from adding it as a new feature to the catalog. However, if a user most naturally conceptualizes her preferences in terms of this feature, the system should allow expression of preferences using that feature.

In recent work we developed a model for *subjective feature elicitation* that queries users about the feature in question, so that utility tradeoffs can be assessed in terms of the new feature (Boutilier, Regan, and Viappiani 2009a). Assuming that the user's underlying utility function is known and can be used to render judgements of relevance, the model casts the problem as one of *concept learning* (Angluin 1987; Hellerstein et al. 1996), but with the goal of learning *just enough* about the concept definition to make a good or optimal decision.

In this work we extend this model to the more realistic case in which the utility function is not known in advance, hence requiring a model that encompasses both feature and utility elicitation. As we discuss below, separating these two forms of elicitation into different "phases" is problematic. As a consequence, we must engage in *simultaneous* feature and utility elicitation. Our contributions are three-fold. First, we define a model that allows simultaneous elicitation of user utility and user features, making appropriate tradeoffs between the two types of information. We use *minimax regret* (Boutilier et al. 2006) as our decision criterion *given concept and utility uncertainty*, allowing good decisions to be made without complete specification of either component. Second, we describe an integer program (IP) formulation for computation of minimax regret in the case of conjunctive concepts, along with a computationally effective constraint generation procedure for its solution. Third, we offer several heuristic techniques for eliciting concepts and utility that reduce minimax regret quickly. In contrast to standard concept learning, we aim to reduce "relevant" concept uncertainty w.r.t. the utility model, rather than learn an accurate concept definition for its own sake. Partially elaborated concept definitions also influence the choice of utility queries. This provides an integrated preference elicitation methodology that allows a user to dynamically (and partially) specify *their own* utility-bearing product features.

A preliminary, shorter version of this paper appeared as (Boutilier, Regan, and Viappiani 2009b).

Background

We begin with relevant background material and a review of our earlier model for regret-based feature elicitation.

Underlying Decision Problem

We assume a system is charged with the task of recommending an option to a user in some multiattribute space, for instance, the space of possible product configurations from some domain (e.g., computers, cars, apartment rental, etc.). Products are characterized by a finite set of attributes $\mathcal{X} = \{X_1, ...X_n\}$, each with finite domain $Dom(X_i)$. Let $\mathbf{X} \subseteq Dom(\mathcal{X})$ denote the set of *feasible configurations*. For instance, attributes may correspond to the features of various cars, such as color, engine size, fuel economy, etc. $\mathbf{X}$ is defined either by constraints on attribute combinations (e.g., constraints on computer components that can be put together) or by an explicit database of feasible configurations (e.g., a rental database). The user has a *utility function* $u : Dom(\mathcal{X}) \rightarrow \mathbf{R}$. For simplicity, we will assume *additive* utility (Keeney and Raiffa 1976). The precise form of u is not critical, only that $u(\mathbf{x}; w)$ is linear in the parameters (or weights) w. Thus our approach is easily generalized to more general models, such as generalized additive independent (GAI) models (Fishburn 1967; Braziunas and Boutilier 2007). A simple additive model in the car domain might be:

$$u(Car; w) = w_1 f_1(\ell/100km) + w_2 f_2(EngSz) + w_3 f_3(Color).$$

The optimal product $\mathbf{x}^*_w$ for a user with utility parameters w is the $\mathbf{x} \in \mathbf{X}$ that maximizes $u(\mathbf{x}; w)$.

Regret-based Preference Elicitation

We will not generally have direct access to the user's utility parameters w. Thus some form of preference assessment is required. We assume a regime in which the utility function is (perhaps partially) elicited from the user (Boutilier et al. 2006; Salo and Hämäläinen 2001; Viappiani, Faltings, and Pu 2006; Gelain et al. 2010). Elicitation is used to refine its knowledge of w. However, decisions will generally be made without full knowledge of w for two key reasons (Boutilier et al. 2006). First, good or optimal decisions can often be made with little utility information. Second, the value of certain utility information (w.r.t. impact on decision quality) is often not worth the (cognitive, time, or computational) cost of obtaining it. Assume a decision must be made, but the system knows only that $w \in W$, i.e., the user's utility function lies in some space W. We use *minimax regret* for making decisions in the face of such utility uncertainty (Boutilier et al. 2006). Minimax regret (Savage 1954) has been advocated as a means for robust optimization in the presence of data uncertainty (Kouvelis and Yu 1997), and has been used for decision making with utility uncertainty (Boutilier et al. 2006; Salo and Hämäläinen 2001). Given utility space W, define the *max regret* of $\mathbf{x} \in \mathbf{X}$, the *minimax regret* of W and the *minimax optimal configuration* as follows:

$$MR(\mathbf{x}; W) = \max_{w \in W} \max_{\mathbf{x}' \in \mathbf{X}} u(\mathbf{x}'; w) - u(\mathbf{x}; w)$$
$$MMR(W) = \min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}, W)$$
$$\mathbf{x}^*_W = \arg\min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}, W)$$

Intuitively, $MR(\mathbf{x}, W)$ is the worst-case loss associated with recommending $\mathbf{x}$; i.e., assuming an adversary will choose the user's utility w from W to maximize the difference in utility between the optimal configuration (under w) and $\mathbf{x}$. The minimax optimal configuration $\mathbf{x}^*_W$ minimizes this potential loss. $MR(\mathbf{x}, W)$ tightly bounds the loss associated with $\mathbf{x}$, and is zero iff $\mathbf{x}$ is optimal for all $w \in W$.

Minimax regret has been applied successfully to robust optimization given utility uncertainty in a variety of domains, for decision problems involving large-scale mixed integer programs (MIPs) and product databases (Boutilier et al. 2006; Braziunas and Boutilier 2007; Boutilier, Sandholm, and Shields 2004). While regret-optimization requires the solution of a minimax problem with a quadratic objective, the application of Benders' decomposition, constraint generation, and various reformulations renders the problem feasible, converting it to a (linear) IP. We adapt these techniques below. It also provides for effective means of utility elicitation (Boutilier et al. 2006; Braziunas and Boutilier 2007). One powerful heuristic strategy is the *current solution strategy*, where preference queries are asked that involve the current minimax-optimal product $\mathbf{x}^*_W$ and/or the adversarial configuration (or *witness*). Unlike volumetric-based approaches to elicitation (Toubia, Hauser, and Simester 2004), regret-based elicitation reduces utility uncertainty only in the relevant regions of utility space, exploiting knowledge of which products are actually feasible.

Subjective Feature Elicitation

It is often most natural to let a user specify her preferences using *subjective features* that have a "personalized" (but unknown) definition in terms of catalog features. Boutilier, Regan, and Viappiani (2009a) treat subjective features as (Boolean) concepts over the catalog features, which are partially learned by querying the user using concept queries (specifically, *membership queries* (Angluin 1987)). Unlike standard concept learning, the goal is not to learn the entire concept definition, but simply to learn enough about it to make a good recommendation. In constrast to the model we develop here, the user's utility function is given—each product has a *known value* that is independent of concept satisfaction, and a *bonus* if the concept is satisfied—and the only uncertainty lies in the concept definition.

Minimax regret is then adapted to pure concept uncertainty: a recommended product has some value, but may or may not satisfy the concept; and an adversary chooses the concept (and an alternative product) so as to maximize the "value plus bonus" advantage over the recommended product. Query strategies are investigated that refine *relevant* concept uncertainty, and it is shown that optimal recommendations can be made with partial information about the exact feature/concept definition. Since the model of Boutilier, Regan, and Viappiani (2009a) is a special case of ours, we discuss it in more depth below.

Feature and Utility Uncertainty

Attributes over which a user forms her preferences will often not coincide with catalog features. We consider *subjective features* that are objectively definable using catalog attributes, but where the definition varies from user to user.[1] For instance, the notion of a "safe" car may differ for a parent with small children, a young, single professional interested in high-performance vehicles, and a family that takes frequent trips to the mountains. The *concept* "safety" thus has *personalized definitions*. The user has *preferences* for safety, just as she does for other attributes. A utility function over this extended attribute space describes her preferences and determines the optimal vehicle. Hence, our recommender system must engage in *both preference elicitation and feature elicitation* to make a suitable recommendation.

This leads to interesting tradeoffs in elicitation. One could engage in feature elicitation using well-known concept learning techniques (Angluin 1987; Hellerstein et al. 1996) and then, with a full definition in hand, move to preference elicitation. But this could be wasteful: suppose we learn that safety requires attribute X_i to be true (e.g., have side airbags) but know nothing else about the concept. If we engaged in preference elicitation simultaneously and ascertained that no cars in the user's price range satisfy X_i—or that other more important features must be sacrificed to obtain X_i—then the full concept definition is not needed for optimal allocation. Conversely, we could engage in preference elicitation, using the subjective feature as an attribute without knowing its definition, and then engage in feature elicitation. However, without some idea of the concept definition, early termination criterion using regret, and many useful query strategies, can't be used; typically, much more preference information than needed will be elicited. This suggests that *interleaved* feature and utility elicitation can be much more effective.

In this section, we first formalize our basic model of utility and concept uncertainty. We then define the minimax regret decision criterion for this case. Finally, we develop an IP formulation for solving the computing minimax regret. We turn to the question of elicitation in the next section.

Basic Model

Assume features $\mathcal{X} = \{X_1, ...X_n\}$ which we take to be Boolean for ease of exposition (nothing critical depends on this), and a feasible product set $\mathbf{X} \subseteq Dom(\mathcal{X})$. User utility for any product $\mathbf{x} \in \mathbf{X}$ is decomposed into two components. First, the user has some utility or *reward* w.r.t. catalog features, denoted by $r(\mathbf{x}; w)$ where w are the parameters of this reward function. We assume r is additive over $\mathcal{X}$. This assumption is not critical, only that r is linear in whatever parameterization w we adopt.[2] The user also has a preference for configurations satisfying some target concept or *subjective feature* c, an unknown Boolean function over $\mathcal{X}$: $c(\mathbf{x}) = c(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.[3] Assume c is drawn from a particular function class or hypothesis space H (e.g., the set of conjunctive concepts). We treat identification of c as a problem of concept learning (Angluin 1987; Hellerstein et al. 1996), with some query set Q that can be used to refine the target concept. For instance, membership queries would be quite natural (e.g., "do you consider the following car to be safe?"). A value or *bonus* b is associated with any $\mathbf{x}$ s.t. $c(\mathbf{x})$ holds, representing user utility for concept satisfaction. Let c be the user's subjective feature or concept, w her reward vector, and b her bonus. Since b is simply another utility parameter, we incorporate it into w (using w_b to denote its value in w). Assuming utility independence for concept satisfaction relative to other preferences, we define the utility of $\mathbf{x}$ under concept c and reward/bonus weight vector (or utility parameters) w to be:

$$u(\mathbf{x}; w, c) = r(\mathbf{x}; w) + w_b c(\mathbf{x})$$

(treating $c(\mathbf{x})$ as an indicator function). The utility of $\mathbf{x}$ is its reward, plus the bonus b if $\mathbf{x}$ satisfies c. The optimal configuration is $\mathbf{x}^*_{w,c} = \arg\max u(\mathbf{x}; w, c)$.

Since c is definable in terms of catalog features, we could in principle elicit utilities using only catalog features. However, allowing a user to articulate her preferences in terms of natural composite features can dramatically reduce the burden of elicitation; furthermore, the addition of such aggregate features with suitable definitions can greatly increase the degree of (conditional) utility independence in a model. We focus our presentation assuming a single concept for ease of exposition. Naturally, a user may have multiple subjective features over which she conceives of her preferences. The extension to multiple features is conceptually straightforward, though does increase computational and query complexity. We point out briefly below how our formulation should be generalized to handle multiple concepts.

Minimax Regret

During elicitation, we are uncertain about the true utility w and the true concept c. Hence, we cannot generally identify the optimal product $\mathbf{x}^*_{w,c}$; but we can make a decision with partial utility and concept information. Let W be the set of feasible utility functions, those consistent with any prior information we have about user preferences and user query responses. W is generally a convex polytope given by linear constraints on utility parameters (as discussed below). Let *version space* $V \subseteq H$ represent our current set of consistent hypotheses w.r.t. c (Mitchell 1977), i.e., those that respect any prior knowledge about the concept and responses to queries (as discussed below). Define *minimax regret* w.r.t. utility and feature uncertainty as follows:

Definition 1 *Given utility space W and version space V, the* max regret *of $\mathbf{x} \in \mathbf{X}$, the* minimax regret *of (W, V) and the* minimax optimal configuration *are:*

$$MR(\mathbf{x}; W, V) = \max_{w \in W} \max_{c \in V} \max_{\mathbf{x}' \in \mathbf{X}} u(\mathbf{x}'; w, c) - u(\mathbf{x}; w, c) \quad (1)$$

$$MMR(W, V) = \min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}; W, V) \quad (2)$$

$$\mathbf{x}^*_{W,V} = \arg\min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}; W, V) \quad (3)$$

[1] Other subjective features may not be so definable (e.g., aesthetic, visual, or latent features); for this, data-intensive collaborative filtering techniques are more appropriate (Konstan et al. 1997).

[2] Indeed, the techniques developed below can be applied directly to, say, generalized additive models using the methods of Braziunas and Boutilier (2007).

[3] Allowing multivalued concepts is straightforward.

Should we recommend option $\mathbf{x}$, max regret $MR(\mathbf{x}; W, V)$ bounds (tightly) how far this decision could be from optimal. Intuitively, an adversary selects the user's utility function w and the intended subjective feature definition c to maximize the difference in utility between our choice $\mathbf{x}$ and the optimal choice $\mathbf{x}^*_{w,c}$ (notice that the adversary's maximizing configuration must be optimal under (w, c)). A minimax optimal choice is any product that minimizes max regret in the presence of such an adversary, and its max regret is the minimax regret given our current uncertainty.

This definition can be generalized in the obvious way if the version space V and utility space W are linked by *complicating constraints*. This can arise, for example, if the choice of $c \in V$ limits the choice of $w \in W$ (we will see how this arises for certain queries below). It is also easily generalized to the presence of multiple subjective features: we assume a version space V_i for each feature and max regret is defined by selecting a concept c_i from each.

Computing Regret: Conjunctive Concepts

We assume that the underlying configuration problem is represented as a MIP $\max_{\mathbf{x} \in \mathbf{X}} u(\mathbf{x})$. We then incorporate utility uncertainty (in the form of a bounded polytope W) into the MIP, following Boutilier et al. (2006), and feature uncertainty in the form of a version space V, following Boutilier, Regan, and Viappiani (2009a). However, in the latter case, the formulation depends critically on the form of the concept and query classes one admits. We illustrate the formulation for (nonmonotone) conjunctive concepts.

Assume target c is a conjunction of literals over variables X_j. Memberships queries ask if $\mathbf{x} \in c$ for some product $\mathbf{x}$. Let E^+ (E^-) be the set of positive (negative) examples acquired by these queries, and (nonempty) V the induced version space. Instead of representing V using most general and most specific concepts, we encode E^+ and E^- directly in our MIP (e.g., negative examples can directly represent the most general concepts in V (Hirsh 1992)).

Constraint Generation We formulate the minimax problem Eq. 2 as a semi-infinite minimization. Let $(X_1, \cdots, X_n)$ be configuration variables over our n features: their instantiation denotes the minimax optimal product. Let constant $b(\mathbf{x}, w, c) = w_b$ if $c(\mathbf{x})$ and 0 otherwise. Let indicator variable I^c, for each $c \in V$, denote that configuration $(X_1, \cdots, X_n)$ satisfies c; and write $x^j \in c$ (resp., $\overline{x}_j \in c$) to denote that variable X_j occurs positively (resp., negatively) in c. Then $MMR(W, V)$ is given by:

$$\min \quad \delta$$
$$\text{s.t. } \delta \geq r(\mathbf{x}^*_{w,c}, w) - r(X_1, \cdots X_n) + b(\mathbf{x}^*_{w,c}, w, c) - w_b I^c \quad \forall c \in V, \forall w \in W \qquad (4)$$
$$I^c \leq X_j \qquad \forall c \in V, \forall x_j \in c \qquad (5)$$
$$I^c \leq 1 - X_j \qquad \forall c \in V, \forall \overline{x}_j \in c \qquad (6)$$

For any *fixed* concept c and utility function $w \in W$, the adversary maximizes the regret of $(X_1, \cdots, X_n)$ with witness product $\mathbf{x}^*_{w,c}$. The MIP above minimizes against the "worst-case" choice of the adversary, with (4) ensuring MMR is as great as regret given any $c \in V, w \in W$; and (5, 6) encoding whether $(X_1, \cdots, X_n)$ satisfies c.

While this MIP has infinitely many constraints, regret will be maximized at vertices P of polytope W, so this can be replaced by a finite MIP with $O(|P||V|)$ constraints. However, even this gives a MIP of unreasonable size: P can grow exponentially in $|\mathcal{X}|$; and V is exponential in $|\mathcal{X}|$ with conjunctive concepts (and can have doubly exponential size for other hypothesis spaces). Fortunately, regret constraints for most $w \in W, c \in V$ will be inactive, so we use constraint generation to search through the space of adversarial utility functions and concepts. Let $Gen \subseteq W \times V$ be a (small) set of (w, c)-pairs (initially a single pair); we solve a relaxed MIP using constraints of type (4) only for those $(w, c) \in Gen$. Let δ^* and $\mathbf{x}^*$ be the solution to the relaxed MIP. We test for violated constraints by solving the max regret problem $MR(\mathbf{x}^*; W, V)$, detailed below. If $MR(\mathbf{x}^*, W, V) > \delta^*$, the utility-concept pair (w', c')—produced as a witness in the max regret computation below—offers larger regret for $\mathbf{x}^*$ than any $(w, c) \in Gen$; indeed, it corresponds to the maximally violated constraint in the relaxed MIP. So we add (w', c') to Gen and resolve. If $MR(\mathbf{x}^*; W, V) = \delta^*$, $\mathbf{x}^*$ is the optimal solution to $MMR(W, V)$.

The MIP can easily be generalized to multiple subjective features. We simply assume a different version space V_i for each feature i. The max regret subproblem generates constraints consisting of a utility w and concept c_i for each feature. Indicator variables are required for each: if we have m subjectives features, the number of such variables grows by factor m. In practice, we expect m to be small.

Generating Violated Constraints We compute the maximally violated constraint for the MIP above by solving the max regret problem $MR(\mathbf{x}^*; W, V)$ for the current relaxed solution $\mathbf{x}^*$. This too can be formulated as a MIP that, given $\mathbf{x}^*$, chooses an (adversarial) concept c, utility w and a configuration $(X^a_1, .., X^a_n)$. For the concept, let binary indicator variable $I(x_j)$ (resp., $I(\overline{x}_j)$) denote that feature X_j is positive (resp., negative) in the (adversarially selected) concept definition c (if both indicators are false, catalog feature X_j is not part of the concept). We also introduce binary variables B^x and B^a indicating that $\mathbf{x}$ and the witness allocation $(X^a_1, .., X^a_n)$, respectively, satisfy c.

Because of utility uncertainty, the components of w are variables, and the straightforward encoding of $MR(\mathbf{x}; W, V)$ in Eq. 1 gives rise to a quadratic objective: $\max \quad \sum_{j \leq n} w_j X^a_j + w_b B^a - r(\mathbf{x}; w) - w_b B^x$. We use a standard reformulation to convert the product of a continuous and a binary variable into a continuous variable, giving us the linear objective in the MIP below. Using $\mathbf{x}[j]$ to denote the jth literal of $\mathbf{x}$, this MIP gives $MR(\mathbf{x}; W, V)$:

$$\max \quad \sum_{j \leq n} Y_j + Z^a - \sum_{j \leq n} w_j \mathbf{x}[j] - Z^x$$
$$\text{s.t. } B^a + I(x_j) \leq X^a_j + 1.5 \quad \forall j \leq n \qquad (7)$$
$$B^a + I(\overline{x}_j) \leq (1 - X^a_j) + 1.5 \quad \forall j \leq n \qquad (8)$$
$$B^x \geq 1 - \sum_{j:\mathbf{x}[j] \text{ positive}} I(\overline{x}_j) - \sum_{j:\mathbf{x}[j] \text{ negative}} I(x_j) \qquad (9)$$
$$\sum_j I(\neg \mathbf{y}[j]) = 0 \quad \forall \mathbf{y} \in E^+ \qquad (10)$$

$$\sum_j I(\neg\mathbf{y}[j]) \geq 1 \quad \forall \mathbf{y} \in E^- \tag{11}$$

$$Y_j \leq X^a_j w_j\!\uparrow; \quad Y_j \leq w_j \quad \forall j \leq n \tag{12}$$

$$Z^a \leq B^a w_b\!\uparrow; \quad Z^a \leq w_b \tag{13}$$

$$B^x w_b\!\downarrow \leq Z^x; \quad B^x w_b\!\uparrow \leq Z^x + w_b\!\uparrow - w_b \tag{14}$$

$$(w_1, \cdots, w_n, w_b) \in W; \quad (X^a_1, \cdots, X^a_n) \in \mathbf{X} \tag{15}$$

Here $w_j\!\uparrow$ and $w_j\!\downarrow$ denote (constant) upper and lower bounds on w_j. Y_j represents the product $w_j X^a_j$, and takes that meaning due to constraint (12), and the fact that Y^j is maximized in the objective. Z^a represents product $w_b B^a$ (constraint 13). Z^x represents product pB^x (14 and its minimization in the objective). Constraints (7,8) ensure that the adversary does not get the concept bonus w_b (i.e., cannot set $B^a = 1$) unless $(X^a_1, \cdots, X^a_n)$ satisfies the concept dictated by the I-variables. Similarly, (9) ensures that the input configuration $\mathbf{x}$ cannot be denied the bonus (i.e., the adversary cannot set $B^x = 0$) unless $\mathbf{x}$ violates at least one conjunct in the chosen concept. Finally, the concept has to be consistent with all known positive and negative instances (10, 11).

The generalization of the subproblem MIP to the case of multiple subjective features is straightforward. The indicator variables that define concept satisfaction for $\mathbf{x}$ and the adversary's chosen configuration simply need to be replicated for each subjective feature.

Simultaneous Feature and Utility Elicitation

While minimax regret provides an appealing means for making decisions under utility and feature uncertainty, our aim is to learn enough about a user's preferences and underlying concept to make good (or even optimal) recommendations, asking as few queries as possible. In this section, we discuss the different forms of queries and develop several query strategies that can quickly reduce $MMR(W, V)$.

Queries and Constraints

With respect to explicit concept queries we restrict attention to *membership queries* of the form "does $\mathbf{x}$ satisfy concept c?" (e.g., "Do you consider car $\mathbf{x}$ to be safe?"). Such queries are quite natural in this setting, arguably much more so than equivalence, subset and other queries commonly considered in concept learning (Angluin 1987; Hellerstein et al. 1996). Each membership query gives rise to a positive or negative concept example, and the version space can be encoded in a variety of ways depending on the hypothesis class (Hirsh 1992) (e.g., see our encoding in the MIP above). There are a variety of query types that can be used to refine one's knowledge of a user's utility function (we refer to (Keeney and Raiffa 1976; Boutilier et al. 2006; Braziunas and Boutilier 2007) for further discussion). In this work, we focus on *comparison queries*: a user is asked if she prefers one product $\mathbf{x}$ to another $\mathbf{y}$.[4]

Responses to these queries impose linear constraints on W when subjective features are absent. But the situation is more complex with feature uncertainty. If a user states that she prefers $\mathbf{x}$ to $\mathbf{y}$, the greater utility of $\mathbf{x}$ could be due to its satisfaction of the feature. This cannot be captured within W alone, but requires complicating constraints that tie W and V together. One simple solution to this problem is to ask two concept queries whenever one asks a comparison query: if a user is asked whether she prefers $\mathbf{x}$ or $\mathbf{y}$, a membership query for each outcome is asked as well (e.g., "is $\mathbf{x}$ safe?"). This is reasonably natural, since preference assessment likely involves cognitive appraisal of the subjective feature in question. We call such a query a *combined comparison/membership (CCM)* query. This allows us to impose valid linear constraints on W; e.g., if $\mathbf{x}$ is preferred and satisfies the concept, while $\mathbf{y}$ does not, then we have $w\mathbf{x} + b - w\mathbf{y} > 0$.

However, if we want a pure comparison query without the corresponding membership queries, we can still impose valid, complete *conditional constraints* on W, based on the whether $\mathbf{x}, \mathbf{y}$ satisfy the concept, thus linking W and V. Intuitively, we have the following conditional constraints on W when we learn if $\mathbf{x}$ is preferred to $\mathbf{y}$:

$$w\mathbf{x} - w\mathbf{y} > 0 \;\text{ if }\; c(\mathbf{x}), c(\mathbf{y}) \tag{16}$$

$$w\mathbf{x} + b - w\mathbf{y} > 0 \;\text{ if }\; c(\mathbf{x}), \neg c(\mathbf{y}) \tag{17}$$

$$w\mathbf{x} - w\mathbf{y} - b > 0 \;\text{ if }\; \neg c(\mathbf{x}), c(\mathbf{y}) \tag{18}$$

$$w\mathbf{x} - w\mathbf{y} > 0 \;\text{ if }\; \neg c(\mathbf{x}), \neg c(\mathbf{y}) \tag{19}$$

In the case of conjunctive concepts, we linearize these conditional constraints without introducing new variables. We illustrate with constraint (17), which is encoded as:

$$w\mathbf{x} + b - w\mathbf{y} > [\sum_{j \leq n} I(\neg\mathbf{x}[j]) + (1 - I(\neg\mathbf{y}[k]))]\ \Delta\!\downarrow \quad \forall k \leq n \tag{20}$$

Here $\Delta\!\downarrow\; < 0$ is any lower bound on the (negative) difference in utility of any two outcomes; it can be computed as $l - u$, where l is any (crude) lower bound and u an upper bound on the utility of any configuration. Constraint 20 imposes $w\mathbf{x} + b - w\mathbf{y} > 0$ if the multiplier of $\Delta\!\downarrow$ is zero, and is vacuous otherwise: $\sum_{j \leq n} I(\neg\mathbf{x}[j]) = 0$ only if $c(\mathbf{x})$, and is at least 1 if $\neg c(\mathbf{x})$; hence the constraint is vacuous if $\neg c(\mathbf{x})$; and $\neg c(\mathbf{y})$ iff $I(\neg\mathbf{y}[k])$ for some $k \leq n$ iff the term $(1 - I(\neg\mathbf{y}[k]) = 0$ for some k. Thus this constraint is binding at zero iff $c(\mathbf{x})$ and $\neg c(\mathbf{y})$. The other three conditional constraints can be encoded in a similar fashion. These are imposed on the solution of the subproblem $MR(\cdot; W, V)$; they are not required for the master problem (since only valid pairs w, c are generated by the subproblem and "posted" to the master problem). Thus the response to a comparison query can be encoded using quadratic number of constraints, as opposed to a single constraint in the case with no concept uncertainty.[5]

The solution the the subproblem can become more complex in the case of multiple concepts. Specifically, the number of conditional constraints of the form (16–19) that need

[4]Such comparisons can be localized to specific attributes in our additive case, or subsets of attributes in GAI models (Braziunas and Boutilier 2007), and can be generalized to choice sets of more than two products (Viappiani and Boutilier 2009) (as is common in conjoint analysis (Toubia, Hauser, and Simester 2004)).

[5]If concept membership of $\mathbf{x}$ or $\mathbf{y}$ is certain given V and W, then only the relevant conditional constraints are posted (if both are certain, the original, unconditional constraint is used).

to be represented in the subproblem can grow exponentially in the number of subjective concepts. However, we expect that no more than a handful of subjective features will typically be required in practice.

Elicitation Strategies

We now develop elicitation strategies for simultaneous utility and feature uncertainty. To select comparison queries, we adopt the *(comparison) current solution strategy (CCSS)* (Boutilier et al. 2006): given the minimax optimal solution $\mathbf{x}^*_{W,V}$ and the adversarial witness $\mathbf{x}^a$, the user is asked which of these two products is preferred.

To select membership queries, we examine two methods explored by Boutilier, Regan, and Viappiani (2009a). The first is a simple *halving* strategy adapted from standard conjunctive concept learning (Hellerstein et al. 1996): we ask random memberships queries until a positive example is found; then queries are asked by negating literals one by one in the (unique) most specific conjunctive hypothesis. Once a positive example is found, this converges to the true conjunctive concept using a number of queries linear in the number of catalog features.[6] Of course, we need not identify the concept exactly; we terminate once minimax regret reaches an acceptable level. We also explore the *current solution strategy for membership queries (MCSS)*: this selects a query based on which of the optimal product $\mathbf{x}^*_{W,V}$ or witness $\mathbf{x}^a$ satisfy the adversary's choice of concept c^a in the current solution. If $c^a(\mathbf{x}^*_{W,V}), c^a(\mathbf{x}^a)$, then CSS asks membership query $\mathbf{x}^a$; if $\neg c^a(\mathbf{x}^*_{W,V}), c^a(\mathbf{x}^a)$, then CSS asks query $\mathbf{x}^*_{W,V}$; otherwise CSS asks a query depending on the whether $\mathbf{x}^a$ is V-consistent (see (Boutilier, Regan, and Viappiani 2009a) for further details and motivation). To avoid asking useless queries, the system only queries (W, V)-uncertain allocations.[7]

Unlike the cases of pure utility or pure feature elicitation, in the simultaneous case we must also make a decision at each stage regarding which type of query to ask, membership or comparison. In several of the strategies below, we rely on our ability to decompose max regret of the current solution into *reward regret* and *concept regret*. Let $(\mathbf{x}^*, \mathbf{x}^a, w, c)$ be the current solution. Max regret of $\mathbf{x}^*$ is $rr + cr$ (reward regret plus concept regret), where

$$rr = r(\mathbf{x}^a; w) - r(\mathbf{x}^*; w); \quad cr = w_b(c(\mathbf{x}^a) - c(\mathbf{x}^*)).$$

Intuitively, rr tells us how much utility uncertainty is contributing to the max regret of $\mathbf{x}^*$, while cr does the same for concept uncertainty. In our "interleaved" strategies below, we use these measures to determine whether to ask a comparison (utility) query or a membership (concept) query, depending on which is larger; moreover a membership query is asked only if either $\mathbf{x}^*$ or $\mathbf{x}^a$ is (W, V)-uncertain.

[6] If we are able to seed the process with an initial positive example, we can accelerate the halving process rapidly, reducing it to a linear number of queries.

[7] Configuration $\mathbf{x}$ is (W, V)-*uncertain* for version space V and utility space W iff there are $c, c' \in V$ s.t. $c(\mathbf{x})$ and $\neg c'(\mathbf{x})$, and for some $w, w' \in W$, both (c, w) and (c', w') satisfy user response constraints. In other words, $\mathbf{x}$ does not have its concept status *determined* unambiguously by the current W and V.

Given this, we examine six plausible query strategies. Two are *phased strategies* that first attempt to learn the concept and then refine the utility function. The first is dubbed *Ph(H,CCSS)* and initially uses the halving algorithm (membership queries) to determine the precise concept definition, and then uses CCSS (comparisons) to refine utility function uncertainty. The second phased strategy, *Ph(MCSS,CCSS)*, asks an MCSS query whenever $cr > 0$. If neither $\mathbf{x}^*$ nor $\mathbf{x}^a$ is (W, V)-uncertain, we ask a CCSS comparison query, so this strategy is not strictly "phased," but only biased toward membership queries. Our *interleaved strategies* ask a membership query if concept regret exceeds reward regret at the current solution, and a comparison query if reward regret is greater. They use CCSS to generate comparisons; but the first, *I(H,CCSS)*, generates membership queries via halving, while the second, *I(MCSS,CCSS)*, uses MCSS. The *CCM* strategy uses combined comparison-membership queries, with CCSS to generate the comparison, and asking membership queries of both $\mathbf{x}^*$ and $\mathbf{x}^a$.

Finally, we consider a *myopically optimal* strategy *WR* which asks the query that guarantees the greatest regret reduction over possible responses. Let (W, V) be the joint utility-version space. Given any (comparison or membership) query q, *yes* and *no* responses induce refined spaces $(W, V|q = y)$ and $(W, V|q = n)$, respectively; and these partition (W, V). The *(posterior) worst-case regret* of q is:

$$WR(q; W, V) = \max[MMR(W, V|q=y), MMR(W, V|q=n)]$$

The *myopically optimal query* q^*_{WR} minimizes this posterior worst-case regret; hence it maximizes regret reduction (given its worst-case response). Viappiani and Boutilier (2009) show how compute such myopically optimal queries without enumerating the space of queries in the case of utility uncertainty. These ideas can be adapted to our joint utility-concept uncertainty setting. Because of their different semantics, myopically optimal comparison and membership queries are determined independently (using a MIP).[8] While the optimization for the WR strategy is impractical for larger problems, its ability to determine the best (single) query provides a useful benchmark against which to compare our heuristic methods. A more scalable *hill-climbing* optimization (Viappiani and Boutilier 2009) can be used to choose a comparison query: given an initial comparison query $q^i = (\mathbf{x}, \mathbf{y})$ (*do you prefer* $\mathbf{x}$ *to* $\mathbf{y}$*?*), a new pair $q^{i+1} = (\mathbf{x}^*_{(W,V|q^i=y)}, \mathbf{x}^*_{(W,V|q^i=n)})$ is constructed, consisting of the regret-optimal configurations in each of the two response-induced partitions. This process is iterated, and can be shown that $WR(q^{i+1}; W, V) \leq WR(q^i; W, V)$.

Empirical Evaluation

We experiment with the query strategies above, comparing them on randomly generated configuration problems of two sizes. Queries are posed to simulated users, each of which has a randomly generated utility function and a subjective

[8] Roughly, we formulate a MIP where the player chooses a triplet (for membership queries): the query item $\mathbf{x}^\mu$ and the recommendations $\mathbf{x}^y$ and $\mathbf{x}^n$ associated with both answers.

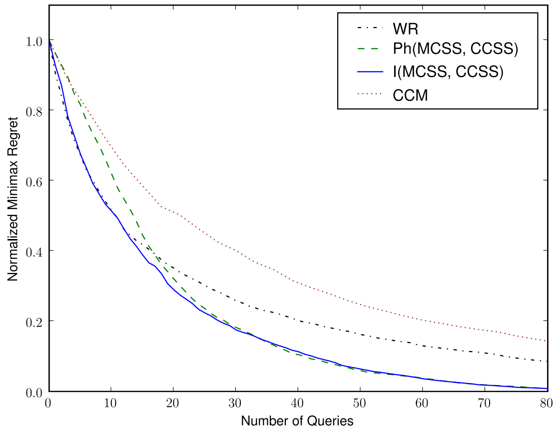

Figure 1: Normalized minimax regret, small concepts (30 runs).

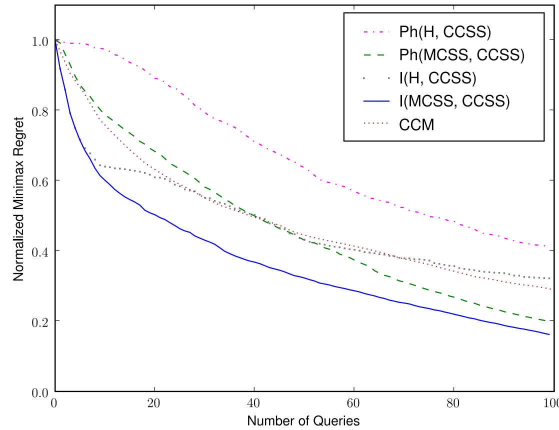

Figure 2: Normalized minimax regret, large concepts (50 runs).

feature to answer queries.[9] We measure the effectiveness of our strategies by examining regret reduction as a function of the number of queries. In our first setting, problems have 20 Boolean variables with random binary constraints to reflect the realistic assumption that the space of feasible products is relatively sparse: on average, about 2100 configurations are feasible. Conjunctive concepts are randomly drawn from a pool of 5 variables, with each variable occurring in the concept positively (probability 0.33), negatively (0.33), or not at all (0.33): as a result, the average conjunctive concept has 3.33 conjuncts. In the second setting, a larger hypothesis space is used, with conjunctive concepts defined over 10 variables, with average concept size of 6.67 conjuncts.

Fig. 1 shows reduction in MMR in the first setting. MMR is normalized w.r.t. the initial regret (i.e., prior to the first query); initial MMR or loss averages about 60% of the adversary's utility. We consider Ph(MCSS,CCSS), I(MCSS,CCSS), CCM and WR, the latter implemented using hillclimbing for comparison queries and exact MIP computation for membership queries. Given our interest in *anytime* recommendations, strategies whose recommended product has lower max regret at any point during the interaction cycle are preferred. The interleaved strategy I(MCSS,CCSS) reduces regret by half in as few as around 12 queries. It dominates CCM significantly, but is only slightly (and not significantly) better than Ph(MCSS,CCSS). Computation of the myopocally optimal WR strategy is too slow to admit real-time response, and is included only as a benchmark in the small concept setting. Interestingly, I(MCSS,CCSS) performs better than WR, and has much faster query selection time (0.5s versus 60s on average).

[9]The subjective-feature-independent *reward* component of a user utility function is assumed to be linear function of the catalog variables. Each feature X_i is randomly assigned a local value weight $r_i \sim U[0, 10]$. The system is given initial bounds on each of these weights, representing partial prior knowledge of the user's utility parameters: these bounds lie (randomly) in the same interval $[0, 10]$ and span 50% of the interval. The upper bound $b^\top$ for the bonus weight w_b was fixed at 10, with no prior information given to the system beside that the bonus must lie in $[0, 10]$.

Fig. 2 shows normalized MMR in our second setting (initial MMR averages a loss of 55% of the adversary's utility). The interleaved strategy I(MCSS,CCSS) again dominates, but Ph(MCSS,CCSS) performs reasonably well. After 100 queries, minimax regret is reduced to about a sixth of its original value. The halving-based strategies and CCM perform significantly worse that the MCSS-based strategies.

These results suggest that the current solution heuristic, which selects both membership and comparison queries in a way that refines concept or utility knowledge of the minimax-optimal $\mathbf{x}^*_{W,V}$ or the adversarial witness, is quite effective. In addition, the use of reward and concept regret to decide between comparison and membership queries is somewhat useful. Examining the behavior of I(MCSS,CCSS) reveals that user sessions tend to start with comparison queries; once reward regret is reduced sufficiently, membership queries are mostly asked until a final short period during which the query types roughly alternate. CCM does not perform that well since each interaction involves 3 queries (a comparison and two membership queries). However, they involve the same three outcomes, thus the cognitive cost might be significantly less than 3 queries. A "leftward compression" of the CCM curve would make the strategy somewhat more competitive.

While the number of queries seems large, the problems are generated randomly to test our strategies with very little prior information. Our queries and strategies also do not exploit the additive nature of utility. Additive (or GAI) structure (Fishburn 1967; Keeney and Raiffa 1976) can greatly simplify utility queries and ease elicitation burden. Moreover, the recommended product will be much closer to optimal in practice than indicated by its max regret. Indeed, we may discover the optimal product long before being able to prove its optimality for the user. Fig. 3 illustrates the *true regret* (or actual loss) associated with the recommended configuration, i.e., the difference between its true utility (given the user's utility function) and the utility of the user's true optimal configuration. We plot it normalized relative to initial regret. Notice that in as few as 10 queries with I(MCSS,CCSS) and WR, true regret drops to roughly

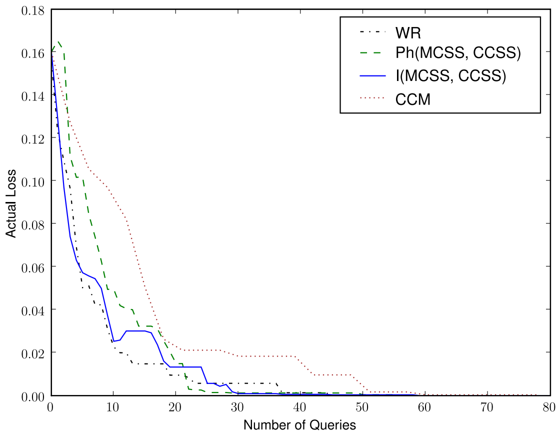

Figure 3: True regret (loss), small concepts (50 runs).

2% of initial regret, and that the optimal product is found in roughly 30 queries on average.

We note that MMR computation is initially very fast, under 1s. (resp. 2s.) in the first setting (resp. second); but it is slowed by conditional constraints: after 50 comparisons queries, MMR computation takes around 10s. in the second problem set. From this perspective, CCM offers the fastest computation. Overall these results suggest that MMR is a very effective means of determining good decisions in the face of simultaneous utility and feature uncertainty. Furthermore, it is a very effective driver of elicitation. Our interleaved, CSS-based approach seems especially effective.

Concluding Remarks

We have presented a model for utility elicitation that allows a user to define her own subjective features over which she can express her preferences. Following Boutilier, Regan, and Viappiani (2009a), we cast feature elicitation as a concept learning problem in which we elicit just enough information to make a good decision. Unlike this earlier model, we must do this under utility uncertainty. Our interleaved, CSS-based approach is especially effective at simultaneous elicitation of concepts and utilities, using regret to make appropriate choices among the different types of queries. Furthermore, optimal or near-optimal product recommendation is generally possible with little concept and utility information.

Our work has obvious connections to concept learning (Hellerstein et al. 1996), but with the critical difference that learning a full concept definition is not our aim. Our methods can also be viewed as a form of active learning; regret reduction (and termination when regret reaches some ε) is a non-Bayesian analog of the value of information criterion that underlies much work on active learning. Exploring these connections is of great interest. Further development of simultaneous elicitation strategies is one critical direction. Additional empirical, theoretical, and user study of these strategies is necessary to validate their practicality. The generalization of our computational and elicitation models to richer concept classes is also vital, as is the investigation of non-additive utility (e.g., GAI) models, multivalued features and concepts, and real-valued domains.

References

Angluin, D. 1987. Queries and concept learning. *Machine Learning* 2(4):319–342.

Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2006. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artifical Intelligence* 170(8–9):686–713.

Boutilier, C.; Regan, K.; and Viappiani, P. 2009a. Online feature elicitation in interactive optimization. *26th International Conference on Machine Learning (ICML-09)*, 73–80.

Boutilier, C.; Regan, K.; and Viappiani, P. 2009b. Preference elicitation with subjective features. *3rd ACM Conference on Recommender Systems (RecSys09)*, 341–344.

Boutilier, C.; Sandholm, T.; and Shields, R. 2004. Eliciting bid taker non-price preferences in (combinatorial) auctions. *19th National Conference on Artificial Intelligence (AAAI-04)*, 204–211.

Braziunas, D., and Boutilier, C. 2007. Minimax regret-based elicitation of generalized additive utilities. *23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, 25–32.

Burke, R. 2002. Interactive critiquing for catalog navigation in e-commerce. *Artificial Intelligence Review* 18(3-4):245–267.

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

Gelain, M.; Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2010. Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies. *Artificial Intelligence* 174(3–4):270–294.

Hellerstein, L.; Pillaipakkamnatt, K.; Raghavan, V. V.; and Wilkins, D. 1996. How many queries are needed to learn? *Journal of the ACM* 43(5):840–862.

Hirsh, H. 1992. Polynomial-time learning with version spaces. *10th National Conference on Artificial Intelligence (AAAI-92)*, 117–122.

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

Konstan, J. A.; Miller, B. N.; Maltz, D.; Herlocker, J. L.; Gordon, L. R.; and Riedl, J. 1997. Grouplens: Applying collaborative filtering to usenet news. *Comm. of the ACM* 40(3):77–87.

Kouvelis, P., and Yu, G. 1997. *Robust Discrete Optimization and Its Applications*. Dordrecht: Kluwer.

Mitchell, T. M. 1977. Version spaces: A candidate elimination approach to rule learning. *5th International Joint Conference on Artificial Intelligence (IJCAI-77)*, 305–310.

Salo, A., and Hämäläinen, R. P. 2001. Preference ratios in multiattribute evaluation (PRIME)–elicitation and decision procedures under incomplete information. *IEEE Trans. on Systems, Man and Cybernetics* 31(6):533–545.

Savage, L. J. 1954. *The Foundations of Statistics*. New York: Wiley.

Toubia, O.; Hauser, J.; and Simester, D. 2004. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research* 41:116–131.

Viappiani, P., and Boutilier, C. 2009. Regret-based optimal recommendation sets in conversational recommender systems. *3rd ACM Conference on Recommender Systems (RecSys09)*, 101–108.

Viappiani, P.; Faltings, B.; and Pu, P. 2006. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research* 27:465–503.