Incremental Utility Elicitation with Minimax Regret Decision Criterion

Tianhan Wang
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3H5, CANADA
tianhan@cs.toronto.edu

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

Abstract

Utility elicitation is a critical function of any automated decision aid, allowing decisions to be tailored to the preferences of a specific user. However, the size and complexity of utility functions often precludes full elicitation, requiring that decisions be made without full utility information. Adopting the *minimax regret* criterion for decision making with incomplete utility information, we describe and empirically compare several new procedures for incremental elicitation of utility functions that attempt to reduce minimax regret with as few questions as possible. Specifically, using the (continuous) space of standard gamble queries, we show that myopically optimal queries can be computed effectively (in polynomial time) for several different improvement criteria. One such criterion, in particular, empirically outperforms the others we examine considerably, and has provable improvement guarantees.

1 Introduction

As software for decision making under uncertainty becomes increasingly common, the process of utility elicitation takes on added importance. Tailoring decisions to a specific user requires knowledge of the user's utility function, information that generally cannot be built into software in advance. In domains as varied as travel planning, product configuration, and resource allocation, to name but a few, assessing a user's utility function is an integral part of interactive decision making. Unfortunately, as is well-known among decision analysts, utility functions are unwieldy, complex, and difficult for users to articulate [5]. To mitigate these difficulties, analysts have developed many techniques for easing the burden of elicitation. For example, structuring of multiattribute utility functions reduces the number of parameters that need to be assessed [6]; and the use *of standard gamble queries* and sensitivity analysis allows users to calibrate utilities more easily [5].

More recently, emphasis has been placed on decision making with incomplete utility information. The principle of maximum expected utility (MEU) cannot be used directly in such cases, since the utility function is unknown; thus new decision criteria are needed. In addition, methods for the automatic generation of queries have been developed that reduce uncertainty or incompleteness in the utility function with minimal effort. Within AI, probabilistic models of utility function uncertainty have been used [4; 2]. By assuming a density over possible utility functions, expectations over this density can be taken to determine the value of a decision; and standard Bayesian updating techniques can be used to account for the responses to queries. A different perspective is taken in work on imprecisely specified multiattribute utility theory (ISMAUT) [8; 7] in which linear constraints on multiattribute utility functions are refined, allowing the set of Pareto optimal decisions to be identified; these constraints arc often refined until one action can be proven optimal. Boutilier, Bacchus and Brafman [3] and Blythe [1] adopt a somewhat related perspective, also reasoning with linear constraints on utility functions.

In this work, we adopt a distribution-free model, working with linear constraints on utility functions, much like ISMAUT. Unlike ISMAUT, we allow for decisions to be made (or recommended) even when the incompleteness in utility knowledge prevents us from proving a decision is optimal. In such circumstances, we adopt the *minimax regret* decision criterion. We also propose and examine several methods for generating queries that reduce regret quickly, in contrast to work on ISMAUT (where query *generation* strategies have not been studied in depth). In this sense, our model more closely resembles probabilistic models [4; 2], which rely on the fact that decisions of good *expected* quality can be made with uncertain utility information. Using the minimax regret criterion, we generate decisions whose quality (difference from optimal) can be bounded in the face of incomplete utility information. These bounds can be traded off against query cost or minimum error requirements to guide the query process.

The paper is organized as follows. We outline relevant background in Section 2 and define the minimax regret criterion for decision making with incomplete utility information. We show how decisions of minimax regret can be computed using simple linear programs (LPs) if utility constraints are linear. We also discuss incremental elicitation, focusing on *standard gamble queries (SGQs)* [5], the responses to which impose one-dimensional, linear utility constraints that can be easily handled using LPs. Our key contribution is described in Section 3, where we develop several myopic elicitation strategies. Assuming linear constraints in each utility dimension (an assumption consistent with the use of SGQs), we show that the minimax regret improvement offered by any response to a SGQ, as a function of the (continuous) query parameter, is piecewise linear (PWL) and weakly monotonic (decreasing or increasing, depending on the response). This fact allows optimal queries under each query strategy to be computed efficiently, in time linear in the number of utility attributes, despite the fact that query space is continuous. We present empirical results comparing the different strategies in Section 4, demonstrating the effectiveness, in particular, of the *maximum expected improvement* strategy.

2 Minimax Regret with Incomplete Utility Information

We assume a system charged with making a decision on behalf of a user in a specific *decision scenario.* By a decision scenario, we refer to a setting in which a fixed set of choices (e.g., actions, policies, recommendations) are available to the system, and the (possibly stochastic) effects of these choices are known. For example, the decisions could be courses of medical treatment with known probabilities for specific outcomes [4]. The system's task is to take the optimal decision with respect to the user's utility function over outcomes, or some approximation thereof. The system may have little information about the user's utility function, so to achieve this, it must find out enough information about this utility function to enable a good decision to be made. We assume that The system has available to it a set of queries it can ask of the user that provide such information. We make these concepts more precise below.

2.1 The Minimax Criterion

Formally, a *decision scenario* consists of a finite set of possible *decisions D,* a finite set of *n* possible outcomes (or states) 5, and a distribution function $\Pr_d$ *e* A(S), for each *d 6 D*[1] The term $\Pr_d(s)$ denotes the probability of outcome *s* being realized if the system takes decision *d.* A utility function *u : S —>* [0,1] associates utility *u(s)* with each outcome *s.* We often view *u* as a n-dimensional vector *u* whose ith component *Ui* is simply *u(s{).* We assume that utilities are normalized in the range [0,1] for convenience. The *expected utility* of decision *d* with respect to utility function *u* is:

$$EU(d, \boldsymbol{u}) = \sum_{i \in S} \Pr_d(s_i) u_i.$$

Note that *EU(d,u)* is linear in *u.* The optimal decision *d** w.r.t. *u* is that with *maximum expected utility (MEU).*

In general the utility function *u* will not be known with certainty at the start of the elicitation process, nor at its end. As in ISMAUT [8; 7], we model this uncertainty by assuming a set of linear constraints *C* over the set of possibly utility functions *U* = $[0, l]^n$. More precisely, we assume that constraints over unknown utility values u_i are linear. We use $C \subsetneq U$ to denote the subspace of *U* satisfying *C.*

If a system makes a decision *d* under such conditions of incomplete utility information, some new decision criterion must be adopted to rank decisions. Following [3], we adopt the *minimax regret decision criterion.*[2] Define the optimal decision d^*_u with respect to utility vector *u* to be

$$d^*_{\boldsymbol{u}} = \arg\max_{d_i} EU(d_i, \boldsymbol{u}).$$

If the utility function were known, d_u would be the correct decision. The *regret* of decision d_i with respect to *u* is

$$R(d_i, \boldsymbol{u}) = EU(d^*_{\boldsymbol{u}}, \boldsymbol{u}) - EU(d_i, \boldsymbol{u}).$$

i.e., the loss associated with executing d_i instead of acting optimally. Let *C* C *U* be the feasible utility set. Define the *maximum regret* of decision d_i with respect to *C* to be

$$MR(d_i, C) = \max_{\boldsymbol{u} \in C} R(d_i, \boldsymbol{u})$$

and the decision d^*_c with *minimax regret* with respect to *C:*

$$d^*_C = \arg\min_{d_i} MR(d_i, C).$$

The *(minimax) regret level* of feasible utility set *C* is

$$MMR(C) = MR(d^*_C, C).$$

If the only information we have about a user's utility function is that it lies in the set C, then d^*_c is a reasonable decision. Specifically, without distributional information over the set of possible utility functions, choosing (or recommending) d^*_r minimizes the worst case loss with respect to possible realizations of the utility function (e.g., if the true *u* were chosen by an adversary).

If *C* is defined by a set *C* of linear constraints, then d^*_c as well as *MMR(C)* can be computed using a set of linear programs [3]. We can compute the *pairwise max regret,* for any pair of decisions d_i and *dj,*

$$PMR(d_i, d_j, C) = \max_{\boldsymbol{u} \in C} EU(d_j, \boldsymbol{u}) - EU(d_i, \boldsymbol{u})$$

using an LP (i.e., maximizing a linear function of the unknown outcome utilities subject to *C).* Solving *0(\D*^{2}*)* such LPs, one for each ordered pair of actions, allows us to identify the decision d^*_c that achieves minimax regret and to determine the minimax regret level *MMR(C).*

2.2 Incremental Elicitation

Given partial knowledge of a utility function in the form of constraint set C, the optimal decision d^*_c may have an unacceptable level in regret. In such a case, a user could be queried in order to reduce this level of uncertainty, thus generally improving decision quality.[3]

'The extension of our elicitation methods to a set of possible decision scenarios is straightforward.

[2]Minimax regret is often used for decision making under *strict* or unquantified uncertainty [5], but its application to imprecisely known utility functions appears not to have been considered in the decision analysis literature.

[3]Note that max regret cannot increase with additional utility information: if $C \subseteq C'$, then $MMR(C) \leq MMR(C')$.

A common type of query is a standard gamble w.r.t. outcome .s_i, where the user is asked if she prefers S_i to a gamble in which the best outcome *sT* occurs with probability / and the worst *s±* occurs with probability 1 - / [6]. We will designate this query *qi(l)* and focus our attention on such *standard gamble queries* or SGQs.[4] Given a response *yes* to query qi(/), the constraint u_t > *I* can be imposed on the user's utility function, thus (in general) refining our knowledge; similarly, a *no* response corresponds to the constraint u_i < *I.* A response to any standard gamble query imposes a one-dimensional (i.e., axis parallel) linear constraint on the utility set. Thus if our initial constraint set *C* is linear, computing the minimax optimal decision after a sequence of SGQs can be accomplished using the LP method above. Furthermore, if *C* consists of a set of bounds on utility values in each dimension—i.e., *C* forms a hyper-rectangle within $[0, l]^n$— then after any sequence of SGQs, the feasible utility set retains this form.

The interactive decision making context we consider is one in which queries are asked repeatedly until the minimax regret level falls to some acceptable value. At that point the "optimal" decision, that with minimax regret given the current constraints, is recommended. Termination can be based on simple thresholding, or can take into account the cost of a query (which can be weighed against the predicted improvement in decision quality).[5] Generally, queries will be asked that offer the greatest predicted improvement in decision quality.

Both query selection and termination rely critically on the way in which "predicted improvement in decision quality" is defined for a query. For example, when asking a query $q_i(l)$ with respect to current constraint set C, we obtain two constraint sets C_{no} and C_{yes}, respectively, given responses, *no* and *yes.* We might then define the improvement in decision quality associated with the query as some function of $MMR(C_{no})$ and $MMR(C_{yes})$. Such a method of evaluating queries is *myopic:* the query is evaluated in isolation, without consideration of its impact on the value or choice of future queries.

It is important to note that optimal query choice is inherently nonmyopic—in general, a sequence of several queries may offer much more value than the aggregate myopic values of the component queries. Unfortunately, nonmyopic methods require some form of lookahead, and thus often impose severe computational costs on the process of query selection. For this reason, we focus on the development of several myopic query selection strategies in the next section. This is analogous to the use of myopic methods for the approximation of value of information in cases where uncertainty is quantified probabilistically; while the computation of true value of information requires some form of sequential reasoning, myopic approximations tend to be used frequently in practice [5; 4].

[4]Other types of queries could be considered, though we rely on the special nature of SGQs in some of our results. SGQs are used widely in decision analysis [5], and have been the main query type studied in recent Bayesian elicitation schemes [4; 2].

[5]Of course, elicitation can continue until a zero-regret (i.e., optimal) decision is identified: this occurs whenever *C* C R_d for some decision d, the region of utility space for which *d* is optimal. The regions R_d are convex polytopes within *U.*

3 Myopic Elicitation Strategies

In this section we describe three myopic strategies for query selection under the minimax regret criterion. Throughout this section we make the following assumptions:

(a) The initial constraints have the form of upper and lower bounds in each utility dimension (these may be trivial bounds, 0, 1).

(b) SGQs are asked, assuming some known best and worst outcome. These ensure that each constraint set (after any query) has the same form as the initial set (i.e., a hyper-rectangle). We denote by uh_l and lb_i the current bounds on the utility U_i of outcome $_{it}$.

(c) For simplicity, we assume a threshold r is used to implement termination; that is, when the predicted improvement of a query falls below T, we terminate the process.

We discuss the impact of relaxing these assumptions later.

3.1 Characterization of Regret Reduction

Query selection is complicated by the fact that, in general, there are *n types* of SGQs that can be asked—one per outcome s_x—and a continuous set of instances of each type *i*— $q_i(l)$ for each $l \in [lb_i, ub_i]$. Whatever criterion is used to select queries, it must distinguish among queries in this n-dimensional continuous space.

Before describing the three strategies, we characterize the reduction in minimax regret offered by a response to a SGQ. Assume current constraint set C, with bounds ub_i and lb_i in dimension i. Given query $q_i(l)$, with $lb_i < l < ub_i$, a negative response will provide us with a refined set of constraints $C_{no} = C \cup \{u_i \le l\}$—i.e., it reduces the upper bound ub_i on u_i—with (ideally) reduced regret. A positive response gives us a similarly refined constraint set C_{yes}.

Focusing on the negative response, note that we'll end up with different C_{no} sets depending on the query point l. Naturally, the closer the query point l is to lb_i, the more informative a negative response is (since it constrains u_i to be in the interval $[lb_i, l]$). Intuitively, we'd expect minimax regret to be smaller given a tighter query. In fact, we can say more. Define $MMR_{no}(C, i, l)$ to be $MMR(C_{no})$ where the *no* is in response to query $q_i(l)$. We have:

Theorem 1 *For any $i \le n$, $MMR_{no}(C, i, l)$ is a PWL, non-decreasing function of l.*

Proof: For space reasons, we provide only an intuitive proof sketch. Define $PMR(d, d', C, i, l) = PMR(d, d', C \cup \{u_i = l\})$; that is, $PMR(d, d', C, i, l)$ is the pairwise max regret of d w.r.t. d' given the current constraints *plus* knowledge that $u_i = l$. It is not hard to see that $PMR(d, d', C, i, l)$ is a linear function of l: maximizing the regret of d w.r.t. d' is effected by setting each component u_j of the utility function to either its upper bound (if $\Pr_d(s_j) \le \Pr_{d'}(s_j)$) or lower bound (if $\Pr_d(s_j) > \Pr_{d'}(s_j)$). Since this max is achieved independently in each dimension, the contribution

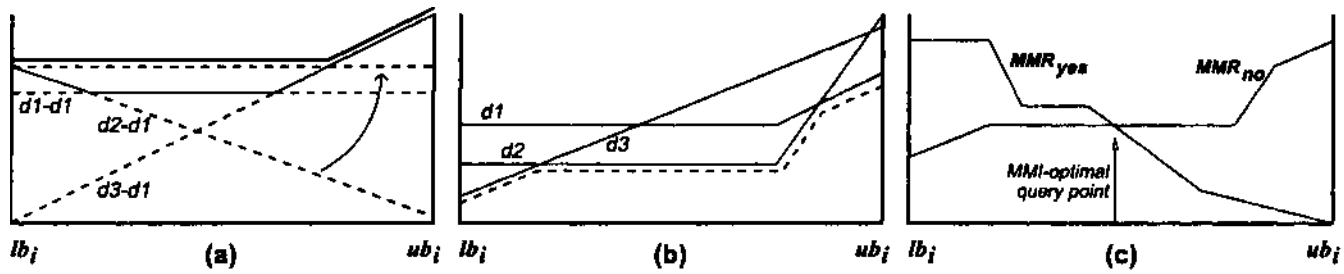

Figure 1: Structure of various functions in dimension i, as a function of/: (a) The *PMR* for *d* (w.r.t. d_1, d_2, d_3) are shown; as is *MR(d,C,,i,l) the max (thin solid line0*of pairwise max regret functions. MR_{no} (thick solid upper line) is obtained by replacing the *PMR* line d_2 — *d* (with negative slope) by the constant line, and again taking the max. (b) $MMR_{n\prime\prime}$ (dashed line) is the min of the MR_{no} functions for each decision; (c) Intersection of MMR_{no} and MMR_{yes} gives maximin improvement in MMR.

to max regret in dimensions other than *i* is constant, while the contribution to regret in dimension *i* given U_i — *I* is linear in l, with coefficient $\Pr_{d'}(s_i) - \Pr_d(s_i)$. It follows that $MR(d,C,i,l) = \max_{d'} PMR(d,d',C,i,l)$ is a PWL convex function of/, since it is the max of these pairwise regret functions (see Figure 1(a)).

Now define $MR_{no}(d,C,i,l) = MR(d, C \cup \{u_i \le l\})$, the max regret of *d* after obtaining a negative response to query $q_i(l)$. (Note that this differs from $MR(d,C,i,l)$, which is defined by assuming $u_i = l$, not $u_i \le l$.) $MR_{no}(d,C,i,l)$ is also a PWL convex function, obtained from the set of linear PMR-functions that make up $MR(d,C,i,l)$ as follows: we replace any linear components with negative slope by the constant line $MR(d,C,i,lb_i)$. Intuitively, if the regret of *d* w.r.t. *d'* after learning U_i = *l* decreases with /, then simply learning that $u_i \le l$ cannot reduce pairwise max regret, since this weaker constraint does not rule out the maximum regret at $l = lb_i$ (see Figure 1(a) for this intuitive flattening of the regret line for d_2 - d_1). This ensures that $MR_{no}(d,C,i,l)$ is also nondecreasing in /, as illustrated in Figure 1(a).

Finally, note that, by definition of minimax regret, $MMR_{no}(C,i,l) = \min_d MR_{no}(d,C,i,l)$. The minimum of a collection of PWL, convex, nondecreasing functions is also PWL and nondecreasing (though not necessarily convex) (see Figure 1(b)).

Note that this proof sketch shows how to construct a finite representation of the function $MMR_{no}(C,i,l)$ as a finite collection of linear functions and inflection points. By entirely analogous reasoning, we also have:

Theorem 2 *For any $i \le n$, $MMR_{yes}(C,i,l)$ is a PWL, nonincreasing function of l.*

3.2 Maximin Improvement

One goal of any query strategy is to determine utility information that reduces regret as quickly as possible. Unfortunately, for any given SGQ $q_i(l)$, the exact reduction in regret cannot typically be predicted in advance, since it differs depending on whether a *yes* or *no* response is obtained. The *maximin improvement (MMI)* query strategy myopically selects queries with the best *worst-case* response. More precisely, let *C* be our current constraint set. We define the minimum improvement of query $q_i(l)$:

$$MI(q_i(l),C) = MMR(C) - \max\{MMR_{yes}(C,i,l), MMR_{no}(C,i,l)\}.$$

At each stage, the query *qi(l)* is asked whose minimum improvement with respect to the current constraint set is maximum. The process stops when no query has minimum improvement greater than threshold r.

To compute the optimal MMI query point, we find the optimal query point in each dimension i, and ask the SGQ corresponding to the dimension with greatest MMI. The PWL representation of the functions $MMR_{yesi}(C,i,l)$ and $MMR_{no}(C,i,\ \ l)$ described above allows the optimal point in each dimension to be computed readily. The point / that offers MMI in dimension *i* can be determined by computing the intersection of the two functions: since one is nondecreasing and the other nonincreasing, the maximum point of the function $\max\{MMR_{yes}, MMR_{no}\}$ must lie at the intersection. Note that the intersection must exist since each has the same maximum value *MMR(C)* (see Figure 1(c)).[6] Finally, the value of the improvement in regret is the difference between the original minimax regret level and this value.

Computation of the intersection of these functions is straightforward, requiring only the computation of the intersection of the linear segments whose bounds overlap. As such, this can be accomplished in linear time in the maximum number of segments in either function. The number of segments in these functions is (very loosely) bounded by $|D|^2$? Since we must compute the optimizing point for each utility dimension, the complexity of this algorithm is $O(n|D|^2)$ The algorithm thus scales linearly in the number of outcomes and quadratically with the number of decisions.

3.3 Average and Expected Improvement

One difficulty with the MMI criterion for query selection is that, due to its worst-case nature, we can often find situations in which no query offers positive (minimum) improvement (we will see evidence of this in the next section), despite the

[6] If the intersection occurs where both functions are "flat," any query point in the intersection can be used.

[7] In practice, the number of segments appears to grow sublinearly in the number of decisions \D\.

fact that the current regret level is positive. This occurs when at least one of the responses for every query offers no improvement, thus stalling the query process. Intuitively, just because one response to a query $q_i(l)$ offers no improvement is no reason not to ask the query: the opposite response may still offer immediate improvement.[8] This suggests an alternative criterion called *maximum average improvement (MAI):* SGQs are ranked according to the average improvement offered by both positive and negative responses.

Computing the optimal query point according to MAI can also exploit the PWL nature of the functions $MMR_{yes}(C,i,l)$ and $MMR_{no}(C,i,l)$. As with MMI, we compute the optimal query point l in each dimension i independently. It is not hard to see that the point of maximum average improvement must occur at an inflection point of one of the two functions. Thus, each dimension can be optimized in time linear in the number of segments in the two functions.

The MAI criterion is not subject to stalling in the sense that MMI is: if $MMR(C)$ is positive, then there exists some query with positive MAI (this will follow from a result discussed below). However, it is subject to a different form of stalling: it may well be the case that the query $q_i(l)$ with MAI occurs at one of the boundary points ub_i or lb_i. In such a case, only one (consistent) response is possible, imposing no additional constraints on the utility function. As such, the constraint set C will remain unchanged, meaning that the same query remains MAI-optimal at the next stage.

This second type of stalling can be prevented. Suppose that a query $q_i(l)$ is optimal, where $l = ub_i$. We know that $MMR_{yes}(C,i,ub_i) < MMR_{no}(C,i,ub_i)$ (i.e., a *yes* offers greater improvement at the point ub_i)\ but since u_i cannot exceed ub_i, the probability of receiving a *yes* response is zero, so the yes-improvement cannot be realized. We can thus make the query $q_i(l)$ appear to be less desirable by accounting for the odds of receiving a specific response. The *maximum expected improvement (MEI)* criterion does just this. We define the expected improvement of a query:

$$EI(q_i(l),C) = MMR(C) - [\Pr(yes|q_i(l),C)MMR_{yes}(C,i,l) + \Pr(no|q_i(l),C)MMR_{no}(C,i,l)].$$

At each stage, we ask the query with maximum EI.

Computation of expected improvement requires some distribution over responses. For simplicity, we assume a uniform distribution over utility functions and noise-free responses: thus, $\Pr(yes|q_i(l),C) = (ub_i - l)/(ub_i - lb_i)$ (with the negative probability defined similarly). This assumption also allows for the ready computation of the MEI-optimal query. Again, we optimize each dimension separately. The optimization in dimension i can be effected by doing separate optimizations in the regions defined by the union of the inflection points in the functions $MMR_{yes}(C,i,l)$ and $MMR_{no}(C,i,l)$. We defer the details, but note that the function being optimized within each region is a simple quadratic function of l that can be solved analytically. Thus the computational complexity of this criterion is similar to that of MMI and MAI. Fortunately, MEI is not subject to stalling:

[8]Note that, since MMI is myopic, even a nonimproving response may offer information that can be exploited in the future.

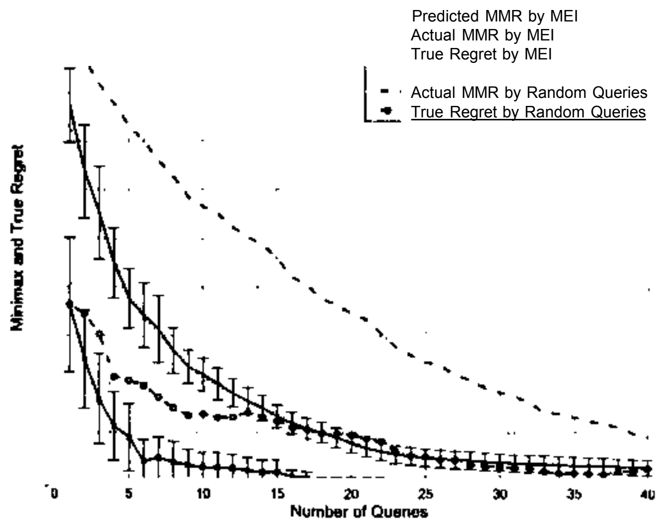

Figure 2: Performance of MEI on three-good problems (40 runs).

Proposition 1 *If $MMR(C) > 0$, then there exists some query*[1] *$q_i(l)$ with positive expected improvement; and at least one response to the MEI-optimal query reduces minimax regret.*

The MEI criterion could be adopted using other distributional assumptions, though the optimization required by query selection could become more complicated. It is worth noting that the manner in which we use distributional information is consistent with the worst-case perspective imposed by the minimax criterion. With some distribution over utility functions, we could adopt the perspective of [4; 2], and make decisions by taking expectations with respect to this distribution. However, even in this case, minimax regret allows one to offer *guarantees* on decision quality that a Bayesian approach does not address. The MEI criterion exploits distributional information *only* to guide the querying process, hoping to reach a point more quickly where acceptable guarantees can be provided; the distribution is not used to evaluate decisions *per se.*

While other prior distributions will generally require a different approach to the optimization problem for query selection, it is interesting to observe that queries associated with a *mixture of uniform distributions* can be determined in exactly the same manner. The derivation of the optimal query given such a mixture is straightforward, and these models have the desirable property (like uniform priors) that they are closed under update by query responses. Thus, if our prior beliefs can be approximated well using a mixture of uniforms with a small number of components, MEI-querying can be used directly as described here, without distribution-specific optimization. It is important to note, however, that, even if the approximate priors are used, the decision quality of the MEI strategy is unaffected—only the number of queries required may be adversely impacted.

We evaluated the MM1, MAI, and ME1 query criteria on a number of elicitation problems in two different domains. With the MEI criterion, we have also tested its robustness to different assumptions about the prior over utility functions.

We first tested our methods in two bidding scenarios involving simultaneous auctions and combinatorial preferences [2]. In the first scenario, a bidding agent must offer bids for four different goods auctioned simultaneously in four different markets. To discretize the decision space, we assume that the agent can offer three distinct bids—low, medium, and high—for each good. Each of these bid levels corresponds to a precise cost: should the bid win, the user pays the price associated with each bid (with, of course, higher prices associated with higher bid levels). To suppress the need for strategic reasoning, the agent has a fixed, known probability of winning a good associated with each of the three bid levels. The probabilities of winning each good are independent, and increasing in the bid level. With four goods and three bid levels, there are 81 possible decisions (mappings from markets to bids) and 16 outcomes (subsets of goods the user might obtain). The user's utility function need not be additive with respect to the goods obtained. For instance, the user might value goods *g* and g_2 in conjunction, but may value neither individually. Thus utility is associated with each of the 16 outcomes. We assume that the overall utility function (accounting for price paid) is quasi-linear; so the price paid is subtracted from the utility of the subset of goods obtained. A smaller scenario with three goods was also run: this has 27 decisions and 8 outcomes.

We first discuss the smaller (3-good) scenario. For each query criterion, we run elicitation using that criterion for 40 steps (or until no query has positive value). For each criterion, 40 trials using random utility functions drawn from $[0, I]^n$ were run, with elicitation simulated using responses based on that function. For each query in a run, we record: *(a) predicted MMR*—the MMR level that is predicted to hold after asking the optimal query;[9] (b) *actual MMR*—the MMR level realized once the actual query response is obtained; and (c) the *true regret*—the difference in utility between the minimax decision and the true optimal decision for the underlying utility function. While our algorithms don't have access to true regret, this measure gives an indication of true decision quality, not just the quality guarantees the algorithms provide.

Figure 2 shows the performance of the MEI criterion (std. error bars are shown on actual MMR and true regret, but are excluded from predicted MMR for legibility). We see that the algorithm quickly converges to a point where the minimax regret guarantees are quite tight: within 20 queries, the average regret guarantee falls below 0.1 (less than 10%); and within forty queries, decision quality is guaranteed to be with 4% of optimal. More interesting, we see that *true regret* falls to under 10% with 5 queries, and to near zero within 20 queries. Thus the actual decision quality associated with acting according the decision with MMR is generally far better than the MMR guarantee. The MEI criterion seems to select suitable queries, allowing optimal decisions and tight regret guarantees to be identified with few interactions. For reference, we also include the performance of randomly selected queries (where both dimension *i* and query point / are chosen uniformly at random from the feasible region). Because the problem is of relatively low dimension, random queries perform reasonably well, though they have difficulty reducing regret to zero, and do not compete with MEI queries.

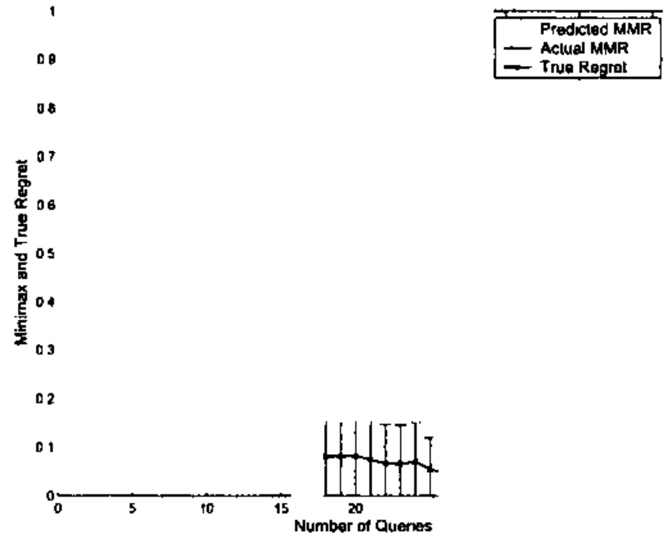

Figure 3: Performance of MAI on three-good problems (40 runs).

Figure 3 shows the same measurements for the MAI criterion. We note that this query strategy does not reduce regret bounds nearly as quickly as the MEI strategy, reaching only an average regret guarantee of 0.18 after 40 queries. True regret is generally much better, but still does not approach the performance of MEI (or even random queries). We note that the MAI criterion often stalls: in such a case, we complete the data with the last minimax regret value. Finally, it is worth noting that the MMI criterion performs extremely badly. We don't plot its performance, but note that in all runs, it stalls after a maximum of five queries; its average minimax regret bound is 0.8, and average true regret level is 0.3 when it stalls.

The MEI criterion appears to offer much better performance than MAI, MMI, or random querying. Figure 4 shows the performance of MEI, MAI and random querying strategies on the larger four-good (16-outcome, 81-decision) scenario. Again we see that MEI converges quickly and outperforms the other strategies. With the increase in dimensionality, random queries fare worse than MAI-optimal queries. We note that in all experiments, the optimal query (regardless of criterion) can be computed very quickly.

We have also tested the MEI criterion on a travel planning domain (as in the previous tests, MEI seems to dominate the other criteria, so we focus on it). In this domain, an agent must choose a collection of flight segments from a flight database to take a user from a source to a destination city [1]. To make the elicitation problem interesting, we added the following information to the DB: the probability of any flight arrival being delayed by specific amount of time;

[9]For example, MMI predicts the maximum regret over all responses, while MEI predicts the expected regret.

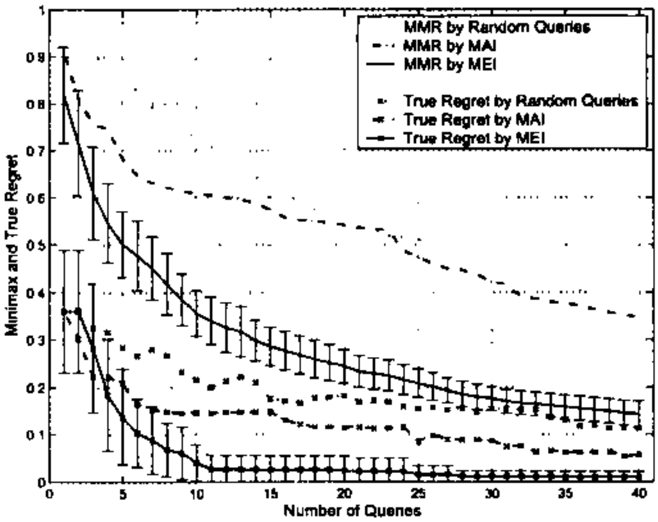

Figure 4: Performance on four-good problems (40 runs).

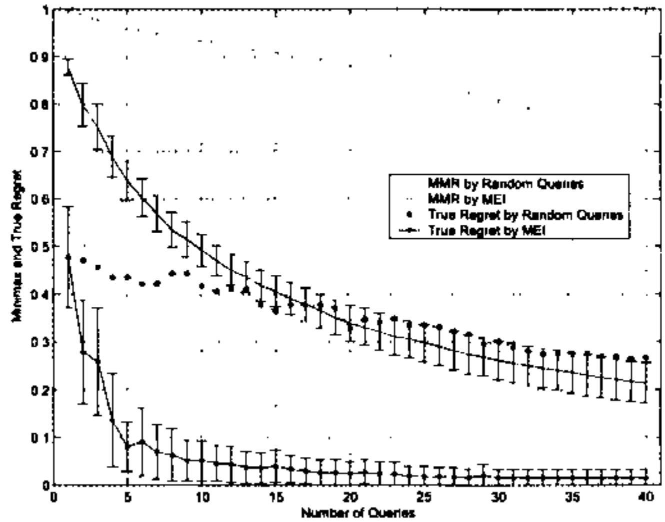

Figure 5: Performance on YYZ-SFO, with utility functions drawn from uniform (40 runs).

the probability of missing a connecting flight as a function of connection time and airport; a distribution over ground travel times to a hotel in the destination city as a function of arrival time (reflecting, e.g., arrival in rush hour or off-peak); and the probability of losing a hotel room as a function of arrival time at the hotel. As a result, for any specific flight combination (decision), a joint distribution over these variables (outcome) is obtained. A user's utility function is quasi-linear, given by her utility for a specific outcome over these four attributes less the flight price. The specific formulation discretizes these attributes, so the outcome space is of size 64. In our experiments, the flight DB was designed to allow 20 flights (both direct and indirect) between pairs of cities.

We tested the ME1 strategy using a uniform prior over utility space to select queries, with user utility functions drawn from the same uniform distribution. The results for a specific source-destination pair (Toronto-San Francisco) are shown in Figure 5. As before, we see that the ME1 strategy easily outperforms random querying, both in terms of the regret guarantees, and the true regret of the decisions it would recommend at each stage. These results are representative of those obtained in other decision scenarios.

We also explored the use of strong prior knowledge to guide the querying process. We repeated the test above, drawing user utility functions from a strongly peaked (truncated) Gaussian distribution over utility space (with diagonal covariance matrix, and variance 0.03 in each dimension). We tested the MEI-criterion using a (hand-chosen) mixture of three uniform distributions over subregions of utility space that very roughly approximated the Gaussian.[10] To test the robustness of MEI to inaccurate priors, we also used MEI using a single uniform prior over all of utility space (despite the fact that the true utility function is drawn from the Gaussian). The results illustrated in Figure 6 demonstrate that MEI can benefit considerably from strong prior information. Indeed, minimax regret is reduced very quickly when a reasonable prior is used to select queries; and true regret is reduced to zero in every instance of this scenario within four queries. Random querying does very poorly, indicating that this problem is not easy to solve without sufficient utility information. The robustness of MEI to inaccurate priors is also in evidence. We see that minimax regret and true regret are also reduced very quickly when an uniformative uniform prior is used to guide the MEI-querying process.

5 Concluding Remarks

We have presented a new procedure for decision making with incomplete utility information which recommends decisions that minimize maximum regret. We defined several different myopic query selection criteria, and showed that myopically optimal queries under each criterion can be computed effectively, in polynomial time. The empirical performance of one such criterion, maximum expected improvement, proved to be rather attractive: not only did it provide strong guarantees after few queries, but true decision quality tended to exceed these guarantees significantly.

Our work differs from existing approaches to preference elicitation in several important ways. Like recent Bayesian approaches [4; 2], our approach identifies a concrete decision in the face of utility function uncertainty. Unlike these methods, for the purposes of decision making, we assume only constraints on possible utility functions, not distributions. As a result, the minimax regret criterion is used to identify decisions with guaranteed error bounds on quality. Our use of constraints on utility functions is more closely related to work on ISMAUT [8; 7]. However, the focus in ISMAUT is the identification of Pareto optimal decisions in the face of utility function uncertainty, as opposed to the choice of a specific decision that maximizes some decision criterion. Furthermore,

[10] In principle, this mixture could have been fit to the actual pnor using, say, EM; but our goal is not accurate modeling of the prior.

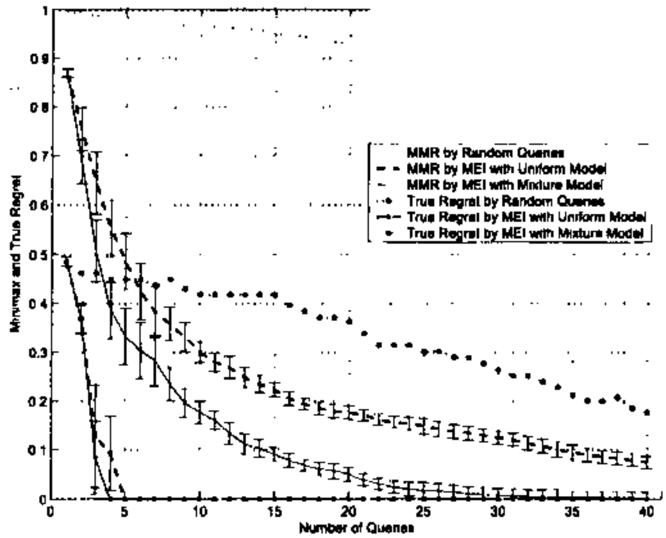

Figure 6: Performance on YYZ-SFO, with utility functions drawn from a strongly peaked Gaussian. MEI querying using both a uniform prior and a 3-component mixture are shown (40 runs).

little attention has been paid to query strategies in ISMAUT, which, in contrast, is our main focus.

There are a number of directions in which this work can be extended. Obviously, scaling issues are of paramount importance. We are currently exploring pruning techniques for removing decisions from consideration that can never be minimax optimal, thus reducing the quadratic dependence on the number of decisions. We are also exploring methods for dealing with more general linear constraints (apart from one-dimensional bounds), as well as more expressive query types. Also of interest are methods for dealing with noisy/inconsistent query responses, and visualization techniques. Finally, we are developing heuristics that simulate some of the effects of nonmyopic elicitation without explicit lookahead. One such technique involves enumerating the vertices of the regions R_d of utility space in which each decision *d* is optimal (the regions are convex polytopes). Queries at those points can quickly help rule out suboptimal actions. We hope to combine the computationally attractive methods devised in this paper with more intensive techniques like this to help reduce the number of required queries even further.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) and the Institute for Robotics and Intelligent Systems (IRIS).

References

[1] Jim Blythe. Visual exploration and incremental utility elicitation. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* pages 526-532, Edmonton, 2002.

[2] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* pages 239-246, Edmonton, 2002.

[3] Craig Boutilier, Fahiem Bacchus, and Ronen I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence,* pages 56-64, Seattle, 2001.

[4] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence,* pages 363-369, Austin, TX, 2000.

[5] Simon French. *Decision Theory.* Halsted Press, New York, 1986.

[6] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs.* Wiley, New York, 1976.

[7] Edward A. Sykes and Chelsea C. White, III. Multiobjective intelligent computer-aided design. *IEEE Transactions on Systems, Man and Cvbemetics,* 21(6): 1498—1511,1991.

[8] Chelsea C. White, III, Andrew P. Sage, and Shigeru Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man and Cvbemetics,* 14(2):223-229,1984.