Preference Elicitation for Risky Prospects

Greg Hines
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
ggdhines@cs.uwaterloo.ca

Kate Larson
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
klarson@cs.uwaterloo.ca

ABSTRACT

Minimax-regret preference elicitation allows intelligent decisions to be made on behalf of people facing risky choices. Standard gamble queries, a vital tool in this type of preference elicitation, assume that people, from whom preference information is being elicited, can be modeled using expected utility theory. However, there is strong evidence from psychology that people may systematically deviate from expected utility theory. Cumulative prospect theory is an alternative model to expected utility theory which has been shown empirically, to better explain humans' decision making in risky settings. We show that the current minimax-regret preference elicitation techniques can fail to properly elicit appropriate information if the preferences of the user follow cumulative prospect theory. As a result, we develop a new querying method for preference elicitation that is applicable to cumulative prospect theory models. Simulations show that our method can effectively elicit information for decision making in both cumulative prospect theory and expected utility theory settings, resulting in a flexible and effective preference elicitation method.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Economics

Keywords

Preference Elicitation, Prospect Theory

1. INTRODUCTION

In many areas of artificial intelligence, we are interested in making decisions on behalf of users [3]. We are often specifically interested in cases where these decisions involve a degree of risk. For example, we may want to create an optimal policy for a Markov Decision Process [6], help people make tough medical choices [2] or help people plan trips, taking into account the probabilities of delays [10]. Computers can help people make better decisions while freeing them to do other things. However, this help requires getting information about the user's utility values. Preference elicitation considers questions such as how best to query the user about their utility values and how much information is needed.

A common method of querying a user is a *standard gamble query* (SGQ) which asks the user to decide between two outcomes. From the user's response, we are able to infer a constraint about their utility function. Such constraints give the set of all possible utility functions. If we have access to a probability distribution of utility functions for a population, we should choose a decision that maximizes the *expected expected utility*; *i.e.* the expected utility of the decision according to each possible utility function multiplied by the probability of the user having that utility function [1, 2]. Without such a probability distribution, it is reasonable to pick a decision which guarantees the best worst-case utility for the user; *i.e.*, minimize the minimax regret [10]. Due to the lower information requirements, we favour the second approach.

SGQs assume that users follow the predictions of *expected utility theory* (EUT). There is strong empirical evidence, however, that people systematically break such predictions [7]. *Cumulative prospect theory* (CPT) is a predominant theory that is better able to explain preferences between risky choices [8]. In this paper, we show that SGQs and CPT are not always compatible.

While there has been work done on preference elicitation with CPT, we discuss why these approaches are not compatible with minimax regret. Then we develop a new querying method which is able to combine CPT with minimax regret. Since choosing an optimal query is difficult we develop heuristics that help us in measuring the value of each possible query.

The paper is organized as follows. Section 2 reviews work on preference elicitation and introduces cumulative prospect theory. We introduce our model for preference elicitation in Section 3. In Section 4 we develop our querying method and in Section 5 we present our heuristics for choosing optimal queries. Experimental results are discussed in Section 6 before concluding with a discussion of future work.

2. BACKGROUND

Our goals for this section are twofold. We first describe a standard setup for preference elicitation for domains where agents must make a decision when the outcome is uncer-

Cite as: Preference Elicitation for Risky Prospects , Greg Hines and Kate Larson, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

tain. In particular, we describe a *regret-minimization* technique that has proven to be effective in settings where users' preferences follow the axioms of expected utility theory. We then introduce cumulative prospect theory, an alternative model of decision making when there are risky outcomes. We conclude the section with a short overview of current preference elicitation approaches which are applicable to a cumulative prospect setting.

2.1 Preference Elicitation for Risky Decisions

Traditional preference elicitation work for risky choices is based on a *decision scenario* with a set of possible outcomes $X = (x_0, \ldots, x_n)$ and a user with a private utility function $u : X \rightarrow [0, 1]$ such that $u(x_0) = 0$ and $u(x_n) = 1$ [10].[1] There exists a finite set of decisions D which we may make on behalf of the user. Each decision is a prospect over X.[2]

DEFINITION 1. *The prospect* $[p_0, x_0; \ldots; p_n, x_n]$ *is the probability distribution* $(p_0, \ldots, p_n)$ *over the set of possible outcomes* $\{x_0, \ldots, x_n\}$.

Expected utility theory (EUT) states that the overall expected utility of a decision d is

$$EU(d, u) = \sum_{x \in X} \Pr_d(x) u(x), \quad (1)$$

and that the best decision, d^*, is the one that maximises the expected utility of the user.

The goal in preference elicitation (PE) is to be able to make a decision for the user that maximizes their expected utility. This requires having adequate knowledge of the user's utility function. Information about the user's utility function is stored as a set of constraints $\{[u_{\min}(x_i), u_{\max(x_i)}]\}$ on the value of $u(x_i)$ for each outcome x_i, *i.e.*, $u_{\min}(x_i) \leq u(x_i) \leq u_{\max}(x_i)$. If nothing is known about a utility function, then the set of constraints is $C_{u(x_0)} = [0, 0]$, $C_{u(x_n)} = [1, 1]$ and $C_{u(x_i)} = [0, 1]$ for all i, $0 < i < n$. The set of constraints for all possible outcomes is C and the set of all utility functions satisfying C is $\mathbb{U}|_C$.

To update constraints, a *standard gamble query* is typically used.

DEFINITION 2. *The standard gamble query (SGQ),* $q_i(p)$*, is a query which asks the user whether they prefer the prospect* $[1 - p, x_0; p, x_1]$ *over the certain outcome* x_i *for some probability* p*. If the user says yes, then we can use algebraic derivations to infer that* $u(x_i) < p$*. Otherwise, we infer that* $u(x_i) > p$*.*

If we knew the user's specific utility function u, then the optimal decision to make would be

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

The *regret* of using decision d_i instead of d^*_u is

$$R(d_i, u) = EU(d^*_u, u) - EU(d_i, u).$$

Given a set of utility constraints C, the maximum possible regret for using decision d_i is

$$MR(d_i, C) = \max_{u \in \mathbb{U}|_C} R(d_i, u).$$

[1]Utility functions are unique up to positive affine transformations, and thus it is always possible to scale them so that their range is the interval [0,1] [5].

[2]We assume there is some maximum outcome that can be achieved by each decision.

The decision which minimizes the maximum possible regret is

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

i.e., d^*_C guarantees the best worst-case regret, also known as the *minimax regret*. The minimax regret with respect to C is

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

The minimax regret can also be found by calculating the *pairwise maximum regret* (PMR) between every pair of decisions,

$$PMR(d_i, d_j, C) = \max_{u \in \mathbb{U}|_C} [EU(d_j, u) - EU(d_i, u)]$$

In the absence of any other information about a user's utility function (such as the probability distribution over the set of all possible utility functions for a given population), Wang and Boutilier argue that choosing a decision that achieves minimax regret is a reasonable approach [10]. Wang and Boutilier's approach is to continue asking SGQs until a desired minimax regret is achieved. Since choosing an optimal SGQ is a "hard" problem, in part because a sequence of SGQs may be more useful than each SGQ in isolation, Wang and Boutilier propose heuristics, known as *myopic elicitation strategies*, for picking which SGQ to ask next. Their most successful strategy was the *maximum expected improvement* (MEI) strategy. MEI estimates the expected improvement of a query as

$$\begin{aligned} EI(q_i(p), C) = MMR(C) - \\ &[\Pr(yes|q_i(p), C) \cdot MMR_{yes}(C, i, p) \\ &+ \Pr(no|q_i(p), C) \cdot MMR_{no}(C, i, p)], \end{aligned}$$

where $MMR_{yes}(C, i, p)$ is the resulting minimax regret if the user responds *yes* to $q_i(p)$, and $MMR_{no}(C, i, p)$ is similarly defined.

2.2 Cumulative Prospect Theory

There has been considerable empirical evidence found that people may systematically violate the axioms of EUT [7]. One of the axioms of EUT is the *axiom of independence* which states that if a user prefers the outcome x_i over x_j, then that preference will hold regardless of alternative choices or scaling of probabilities [5]. However, while most people prefer the guaranteed outcome of \$3,000 over the prospect $[0.2, \$0; 0.8, \$4,000]$, people tend to prefer the prospect $[0.8, \$0, 0.2, \$4,000]$ over the prospect $[0.75, \$0; 0.25, \$3,000]$ [4]. Since the ratio of the probabilities for the non-zero outcomes are the same in both prospects, this violates the axiom of independence. It can be shown that there exists no possible utility function for Equation 1 that explains such behaviour. The premise of this example has been repeated in numerous experiments, including those with nonmonetary outcomes [4].

Numerous alternative models to EUT have been proposed, including those which focus on relaxing the axioms behind EUT. While no model so far has managed to explain all experimental evidence, several of these models have proven to be better at explaining and predicting human decisions than EUT. Perhaps the most successful and most famous of these alternative models is *cumulative prospect theory* (CPT) proposed by Kahneman and Tversky [8].

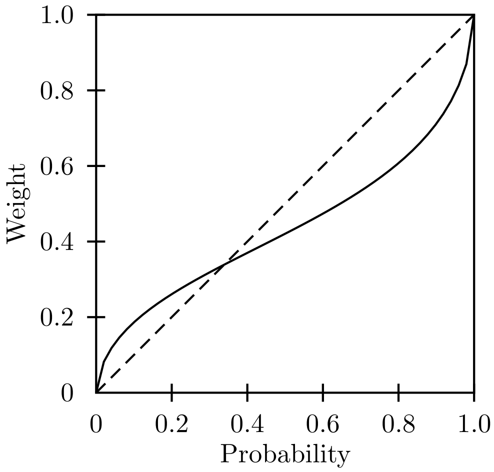

Figure 1: A probability weighting function $w(p)$.

Two key features of human behaviour that CPT captures are *loss aversion* and *probability weighting.* People are loss averse when they are more sensitive to losses than to gains. Probability weighting distorts the probabilities people consider when judging the utility of a risky prospect. It is this second behaviour that we focus on in this paper; our approach, however, can be easily extended to deal with losses as well as gains.

CPT models probability weighting using a *weighting function*, $w(p)$, an example of which is shown in Figure 1. While different functions have been proposed for $w(p)$, they all capture specific human behaviours including overestimating the likelihood of a low probability outcome while underestimating the likelihood of a high probability outcome [8, 11].

As is typical, we set $w(0) = 0$ and $w(1) = 1$. In CPT, the weight of a probability is also dependent on the rank of the respective outcome. For the prospect $X = [p_0, x_0; \ldots; p_n, x_n]$ where $u(x_i) < u(x_{i+1})$, the overall weight, π, for probability $p(x_i)$ is

$$\pi(x_n) = w(p_n)$$
$$\pi(x_i) = w(p_i + \ldots + p_n) - w(p_{i+1} + \ldots + p_n),$$

and the overall utility for the prospect is,

$$U([p_0, x_0; \ldots; p_n, x_n]) = \sum_{x_i} \pi(x_i) u(x_i). \quad (2)$$

Therefore, the utility of a prospect $[1-p, x_0; p, x_n]$ is

$$U([1-p, x_0; p, x_n]) = (1 - w(p))u(x_0) + w(p)u(x_n) = w(p).$$

This immediately shows a challenge of eliciting CPT utilities using SGQ. In particular, a SGQ can only compare $u(x_i)$ against $w(p)$ instead of p. If w is private, we may not be able to elicit preferences using SGQs.

Alternative preference elicitation approaches to SGQs have been used with CPT. One approach is parametric in nature, where specific forms of the utility and probability weighting functions are assumed. Using methods like least-squares fitting, it is possible to approximate the actual functions of the queried user. However, to date, this approach has had more success explaining aggregate results for a group than individual results [11]. Furthermore, to date, no proposed weighting function has been able to correctly model all experimental evidence, so choosing a specific weighting function creates error in the approach [11].

A newer, non-parametric approach called the *two-step method*, by Wakker and Deneffe, provides queries that are able to detach probability weighting from utility queries [9]. Given some initial outcome x_0, two reference outcomes r and R such that $r < R$, and some probability p, the user is asked to find an outcome x_1 such that they are indifferent between the prospects $[1-p, x_0; p, R]$ and $[1-p, x_1; p, r]$. The user is then asked to find an outcome x_2 such that they are indifferent between the prospects $[1-p, x_1; p, R]$ and $[1-p, x_2; p, r]$. As long as $r > x_1$, these two indifferences imply that

$$u(x_2) - u(x_1) = u(x_1) - u(x_0).$$

This process can be repeated to create a *standard sequence of outcomes*, $\{x_0, x_1, \ldots, x_j\}$ such that $u(x_{i+1}) - u(x_i) = u(x_i) - u(x_{i-1})$. By the uniqueness of u, we can then let $u(x_0) = 0$, $u(x_j) = 1$ and $u(x_i) = i/j$. The two-step method has been used successfully in a number of human trials [9].

The disadvantage of the two-step method is that $u(x_i) - u(x_{i+1})$ is constant for a given standard sequence. This means that there is no way to ask queries about outcomes between those in a standard sequence, which limits our ability to ask queries that are best able to reduce the minimax regret. Our goal is to create a querying technique, inspired by the two-step method, that can be efficiently used in a CPT setting to minimize the minimax regret.

3. PREFERENCE ELICITATION MODEL

We begin by defining our model: our assumptions about the user's utility and probability weighting functions, as well as what data we have about the user's utility function at any given time.

Let the set of possible outcomes, Y, be isomorphic to $\mathbb{R}^n_{\geq 0}$ and the user's utility function be $u : Y \to \mathbb{R}_{\geq 0}$. We assume that u is both continuous and monotonically increasing. The user has a probability weighting function w, which we also assume is continuous and monotonically increasing; this characterization of w is supported by experimental evidence [11]. The user's overall utility of a prospect is given by Equation 2.

We have a set of decisions D which can be viewed as probability distributions over $X = [x_0, \ldots, x_n]$, a finite subset of outcomes of Y.

Example: A roll of a dice determines how much money a user wins. In this case, $X = [\$1, \ldots, \$6]$. Then $Y = \{\$y | y \in \mathbb{R}_{\geq 0}\}$.

We scale u such that $u(x_0) = 0$ and $u(x_n) = 1$. For $x_i \in X$, we have a set of constraints for $u(x_i)$,

$$C_{u(x_i)} = [u_{\min}(x_i), u_{\max}(x_i)]$$

where $u_{\min}(x_i)$ is the minimum possible value for $u(x_i)$ and $u_{\max}(x_i)$ is similarly defined. Initially $C_{u(x_0)} = (0, 0)$, $C_{u(x_n)} = (1, 1)$ and $C_{u(x_i)} = (0, 1)$ for all $0 < i < n$. The set of all constraints is C which includes all constraints necessary to ensure the monotonicity of u. The set u_{known} is the set of outcomes in Y for which we know the exact utilities. Initially, $u_{\text{known}} = \{x_0, x_n\}$. For $u_{\min}(x_i)$, it is convenient for us to define $u^{-1}_{\min}(x_i)$ as the outcome with the utility $u_{\min}(x_i)$ and $u^{-1}_{\max}(x_i)$ in an analogous manner.

Our goal is to select a decision which guarantees a minimax regret below some desired threshold [10].

4. QUERIES

In this section we describe the queries we ask the user in order to determine their preferences. We use two types of queries: *configuration queries* and *outcome queries*. Configuration queries provide information about the user's probability weighting function. Outcome queries obtain information about the user's utility function. The preference elicitation process works by initially asking only configuration queries. After enough information has been gathered about the user's probability weighting function, we proceed to ask only outcome queries. Only outcome queries can reduce the regret.

4.1 Configuration Queries

Configuration queries are used to solve

$$\frac{w(p)}{w(1-p)} = \frac{1}{2}, \tag{3}$$

which, as shown in Equations 8 and 9 of Section 4.2, is all we need to know about w to be able to remove the effects of probability weighting from the user's response to an outcome query.

We start by picking two outcomes r and R in Y and asking the user to pick a probability p such that

$$[1-p, x_0; p, R] \sim [1-p, x_n; p, r], \tag{4}$$

i.e., the user is indifferent between the two prospects. Since w is dependent on the ordering of the outcomes, we fix $R \succ r \succ x_n$. Since X is a finite subset of the continuous set Y, we can always find values for r and R satisfying this constraint. Any such r and R will work. We then ask the user to pick some outcome $z \in Y$ such that

$$[1-p, x_n; p, R] \sim [1-p, z; p, r]. \tag{5}$$

If $z \succ r$, we need to increase r and R and repeat these two queries.

According to CPT, Equation 4 implies

$$w(p)[u(R) - u(r)] = (1 - w(p))[u(x_n) - u(x_0)], \tag{6}$$

and Equation 5 implies

$$w(p)[u(R) - u(r)] = (1 - w(p))[u(z) - u(x_n)]. \tag{7}$$

Together, Equations 6 and 7 imply that

$$u(z) - u(x_n) = u(x_n) - u(x_0),$$

which means that $u(z) = 2$.

Since

$$\frac{w(p)}{w(1-p)}$$

equals 0 when p equals 0 and approaches infinity as p approaches 1, by the Intermediate Value Theorem, there exists some probability p^* which satisfies Equation 3. Since $w(p)$ is monotonically increasing, the LHS of Equation 3 is also monotonically increasing with respect to p. As a result, we can do a binary search for p^*. The range of possible values for p^* is $[p^*_{\min}, p^*_{\max}]$, where initially $[p^*_{\min}, p^*_{\max}] = [0, 1]$. Let

$$p^*_{BS} = \frac{p^*_{\min} + p^*_{\max}}{2}.$$

We now ask the user to compare the prospects

$$f_1 = [1 - p^*_{BS}, x_0; p^*_{BS}, z]$$

and

$$f_2 = [p^*_{BS}, x_0; 1 - p^*_{BS}, x_n].$$

If the user prefers f_1 over f_2, then

$$\begin{aligned} &w(p^*_{BS})u(z) + [1 - w(p^*_{BS})]u(x_0) \\ &\qquad > w(1 - p^*_{BS})u(x_n) + [1 - w(1 - p^*_{BS})]u(x_0) \end{aligned}$$

$$\begin{aligned} \Rightarrow w(p^*_{BS})u(z) &> w(1 - p^*_{BS})u(x_n) \\ \frac{w(p^*_{BS})}{w(1 - p^*_{BS})} &> \frac{u(x_n)}{u(z)}. \\ &= \frac{1}{2} \end{aligned}$$

Therefore, our estimate of p^* is too high and we update $[p^*_{\min}, p^*_{\max}]$ to be $[p^*_{\min}, p^*_{BS}]$. By analogous reasoning, if the user prefers f_2 over f_1, our estimate of p^* is too low and we update $[p^*_{\min}, p^*_{\max}]$ to be $[p^*_{BS}, p^*_{\max}]$. By repeating this binary search, we can eventually find the value of p^*. In our experiments, we found repeating the search for 10 iterations gave an accurate enough value for p^* for our approach to always work.

4.2 Outcome Queries

To update the utility constraints in C, we need to know more about the user's utility function. Outcome queries indirectly pick a utility value and ask the user what outcome has that utility value. The queries are designed so that the user's probability weighting can be factored out of the user's response.

For any two outcomes s and t in $\mathbb{R}^n_{\geq 0}$ with known utilities, *i.e.*, $\{s, t\} \subseteq u_{\text{known}}$, such that $u(s) < u(t)$, the outcome query (s, t) asks the user to pick an outcome v in $\mathbb{R}^n_{\geq 0}$ such that

$$[1 - p^*, s; p^*, t] \sim [p^*, s; 1 - p^*; v].$$

This indifference implies,[3]

$$\begin{aligned} &w(1 - p^*)u(v) + [w(1) - w(1 - p^*)]u(s) \\ &\qquad = w(p^*)u(t) + [w(1) - w(p^*)]u(s) \\ &\Rightarrow w(1 - p^*)u(v) - w(1 - p^*)u(s) = w(p^*)u(t) - w(p^*)u(s) \end{aligned}$$

$$\Rightarrow u(v) = u(s) + \frac{w(p^*)}{w(1 - p^*)}(u(t) - u(s)) \tag{8}$$

$$= u(s) + \frac{1}{2}(u(t) - u(s)) \tag{9}$$

$$= \frac{u(s) + u(t)}{2}.$$

Since we know $u(s)$ and $u(t)$, we can add v into u_{known} and update the constraints in C as applicable, *e.g.*, for the outcome $x_i \in X$, if $u^{-1}_{\min}(x_i) \prec v \preceq x_i$, then we update $u_{\min}(x_i)$ to be $u(v)$. Finally, we calculate the new minimax regret. If the regret is below some desired threshold, we terminate the process. Otherwise, we continue asking outcome queries.

In summary, our approach begins by asking configuration queries to learn about the user's probability weighting

[3]Due to the rank-dependent nature of w, the derivation assumes $v \succ s$. To prove $v \succ s$, we use proof by contradiction which follows from algebraic derivation.

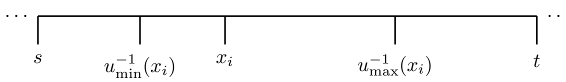

Figure 2: A possible outcome query (s, t) that may be able to update either $u_{\min}(x_i)$ or $u_{\max}(x_i)$.

function. Once we have enough information about w to solve Equation 3, we begin asking outcome queries. Outcome queries provide information about the user's utility values. These queries are designed so that a user's probability weighting may be factored out of their responses. After each outcome query, we update any possible constraints which then reduces the minimax regret. If the minimax regret is below a desired threshold, we terminate the process. Otherwise, we continue asking outcome queries.

5. HEURISTIC ELICITATION STRATEGIES

In this section, we consider the problem of how to choose the next outcome query to ask. The binary search done by configuration queries is completely deterministic, so the next configuration query is always chosen for us. However, we can ask an outcome query for any pair of outcomes in u_{known}. A sequence of outcomes queries may also be more useful than each individual outcome query [10]. Therefore, we consider two heuristics for estimating the best query for reducing the minimax regret.

The first, *most likely minimax regret* (MLMR), uses a parametric approach where we choose some utility function to approximate the user's. By using a method such as least squares fitting we are able to estimate the most likely response to an outcome query, which then allows us to estimate the most likely resulting minimax regret. The process is repeated for every pair of values in u_{known}. We then choose the query with the lowest MLMR value.

Unfortunately, as we mention in Section 6, the MLMR heuristic sometimes fails. This happens when the MLMR value for all queries is equal to the current minimax regret. That is, the most-likely outcome for all queries gives no improvement. In this case, we rely on a backup heuristic, the *expected minimax regret* (EMR).

EMR approximates the PMR between decisions d_i and d_j as

$$= \sum_{x \in X} (p_{d_j}(x) - p_{d_i}(x)) \cdot \begin{cases} u_{\max}(x) & \text{if } p_{d_j}(x) \geq p_{d_i}(x) \\ u_{\min}(x) & \text{otherwise.} \end{cases}$$

Therefore, to estimate the minimax regret after a query, we need to estimate the changes to $u_{\min}(x_i)$ and $u_{\max}(x_i)$ for all x_i. The outcome query (s, t), shown in Figure 2, will be able to update $u_{\min}(x_i)$ if and only if $u^{-1}_{\min}(x_i) \prec v \preceq x_i$, where v is the user's response. To estimate the probability of this occurring, we assume that the probability density of v is uniform between s and t. As a result, the probability of $u_{\min}(x_i)$ being updated is

$$\frac{|x_i - u^{-1}_{\min}(x_i)|}{|t - s|}. \tag{10}$$

If $u_{\min}(x_i)$ is updated, we need to estimate by how much. We assume that the user's utility function is linear between $u^{-1}_{\min}(x_i)$ and $u^{-1}_{\max}(x_i)$. Under this assumption, between $u^{-1}_{\min}(x_i)$ and $u^{-1}_{\max}(x_i)$, the slope of u is

$$\frac{u_{\max}(x_i) - u_{\min}(x_i)}{|u^{-1}_{\max}(x_i) - u^{-1}_{\min}(x_i)|}.$$

With the assumption of an uniform distribution for v, the expected value of v is

$$\left(\frac{|x_i - u^{-1}_{\min}(x_i)|}{2}\right).$$

Therefore, if $u_{\min}(x_i)$ is updated, the expected change in $u_{\min}(x_i)$ is

$$\left(\frac{|x_i - u^{-1}_{\min}(x_i)|}{2}\right)\left(\frac{u_{\max}(x_i) - u_{\min}(x_i)}{|u^{-1}_{\max}(x_i) - u^{-1}_{\min}(x_i)|}\right). \tag{11}$$

The overall expected change to $u_{\min}(x_i)$, $\Delta(u_{\min}(x_i))$, is given by Equation 10 multiplied by Equation 11. We can calculate $\Delta(u_{\max}(x_i))$, the expected change to $u_{\max}(x_i)$, in an analogous manner. For any two decisions d_i and d_j, the expected change to the PMR between those two decisions is

$$\Delta PMR(d_i, d_j) = \sum_{x \in X} (p_{d_j}(x) - p_{d_i}(x)) \cdot \begin{cases} \Delta(u_{\max}(x)) & \text{if } p_{d_j}(x) \geq p_{d_i}(x) \\ \Delta(u_{\min}(x)) & \text{otherwise.} \end{cases}$$

For each possible query, we calculate the overall ΔPMR. This allows us to estimate the expected PMR resulting from any query. We then choose the query which gives the lowest expected PMR.

In summary, we have two heuristics for choosing the optimal query. MLMR estimates the most-likely minimax regret resulting from a query. EMR estimates the expected minimax regret from a query. We choose whichever query minimizes the metric we decide to use.

Error	0	0.01-0.1	0.11 - 0.15	> 0.15
Percentage	68.5	19	8.5	4

Table 1: Error rates for preference elicitation using SGQs on users with CPT-modelled preferences.

6. EXPERIMENTAL RESULTS

To verify the effectiveness of our preference elicitation approach we conducted a series of experiments. Our goals were as follows. First, to understand if previous preference elicitation models could elicit appropriate preference information from users whose preferences were described by CPT. Second, to understand if our proposed approach could effectively elicit preference information from users with CPT preferences. Finally, to determine if our approach was also an effective model for EUT situations.

We studied the performance of previous preference elicitation models in a CPT setting by implementing Wang and Boutilier's minimax regret model. The implementation included their most successful elicitation strategy, MEI. The experiment included 4 possible outcomes, $\{0, o_2, o_3, 3500\}$, where o_2 and o_3 varied between 0 and 3500 in increments of 50. Since CPT and EUT are most different when probabilities are close to either 0 or 1, we created a simple decision choice between two decisions $[0.05, 0, 0.95, 0]$ and $[0.15, 0, 0, 0.85]$.

The user was simulated using the utility function

$$u(x) = x^{0.88} \tag{12}$$

and weighting function

$$w(p) = \frac{p^{0.61}}{(p^{0.61} + (1-p)^{0.61})^{1/0.61}}. \tag{13}$$

Both of these functions and their parameters are from the literature [8].

The elicitation process was run until the minimax regret was at most 0.01. At this point, the minimax regret decision was selected, and then evaluated using the user's actual utility and weighting functions. We then computed the optimal decision, according to the user's utility function, and thus, determined the *actual regret* of the decision. Since minimax regret is supposed to be a guarantee of the worst case regret, error was measured as

$$\max\{0, \text{actual regret} - \text{minimax regret}\}.$$

An error value greater than zero indicates that the actual regret was higher than the "guaranteed" minimax regret, indicating that the utility function of the user was not being properly modeled. The results, shown in Table 1 show that while 70% of the time there was no error, 19% of the runs resulted in an error between 0 and 0.1, in 8.5% the error was between 0.1 and 0.15 and in 4% the error was greater than 0.15. This represents a potentially significant loss of utility for the user.

The decision scenario used for the rest of our experiments used 8 outcomes, chosen uniformly at random between 0 and 500, and 27 decisions. To create more difficult elicitation problems, decisions were chosen that helped to maximize the minimax regret. Decisions were added in an iterative fashion. For each new decision, 50 candidate decisions were chosen uniformly at random. The decision that maximized the minimax regret with respect to all the decisions already picked was chosen as the next decision. While this helped to create more difficult problems, the monotonicity constraint was a limitation. Without the monotonicity constraint, we would have been able to create significantly more difficult problems. Unless noted otherwise, Equations 12 and 13 were used for the user's utility and weighting functions, respectively throughout the experiments. The parameters for Equations 12 and 13 were fixed throughout the experiments. All experiments were repeated 60 times and run for at most 20 queries. All of our results do not include configuration queries, which averaged an additional 14 queries.

For clarity, our graphs show only the minimax regret and not the actual regret. The actual regret was typically very low; starting at between 0.05 and 0.1 for all of our experiments. By the 5th query, the actual regret was typically around 0.01. However, we believe that a low actual regret is not a good measure of the difficulty of the elicitation problem. To investigate this possibility, we conducted some preliminarily experiments using adversarial users and our querying method from this paper. These users were not required to choose utility values in advance. Instead, the users chose responses to queries that attempted to keep the minimax regret as high as possible. At the end of experiment, the adversarial users had to choose utility values that were consistent with all the query responses they had given while also maximizing the actual regret. The results, which are not presented here, show while the adversarial users were

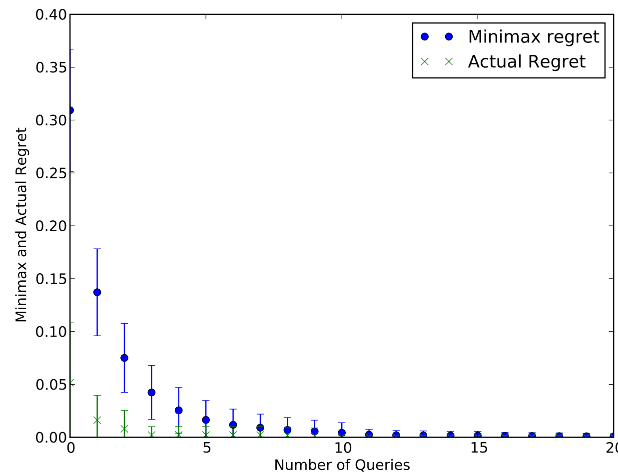

Figure 3: Performance of CPT-based PE with MLMR and EMR on an CPT agent

able to keep the minimax regret relatively high, the actual regret wound up being relatively low. This suggests that a high actual regret is not needed to create situations were reducing the minimax regret is difficult.

We next examined how efficient our approach was with a user with CPT preferences. To establish an baseline for evaluating our approach, we first implemented a random-elicitation strategy. This strategy picks an outcome query by randomly choosing $\{s,t\} \subseteq u_{\text{known}}$ where we have not already queired about the utility $(u(s)+u(t))/2$. The results are shown in Figure blah. The random approach preformed reasonably well since the monotonicity constraints meant that the results from a random query could still be used to update many utility constraints.

For the first test of our approach, we relied on both eliciation heuristics. For the MLMR heuristic, we used the function

$$u(x) = x^{\alpha}, \tag{14}$$

to approximate the user's utility function, where $0 < \alpha \leq 1$. The parameter α was estimated using a non-linear least squares method done on the points in u_{known}. Figure blah also shows the results from this experiment.

shows the minimax regret and actual regret over the course of the elicitation process. After 7 queries, the mean minimax regret was under 0.01. Of the 351 queries, MLMR failed only 4 times.

We next tested our approach with an EUT-based user using the utility function in Equation 12. The results, which are not shown, are very close to those in Figure 3. After 7 queries, the mean minimax regret was 0.011. This suggests that our approach is equally suited to dealing with CPT and EUT users. As a comparison for EUT-based users, we ran Wang and Boutilier's elicitation method using MEI, their most successful elicitation strategy. The minimax and actual regret are shown in Figure 4. At the end of 20 queries, the mean minimax regret was still well above 0.05.

To compare our two heuristics for choosing queries, MLMR and EMR, we ran our approach using only EMR. The results are shown in Figure 5. While EMR was always able

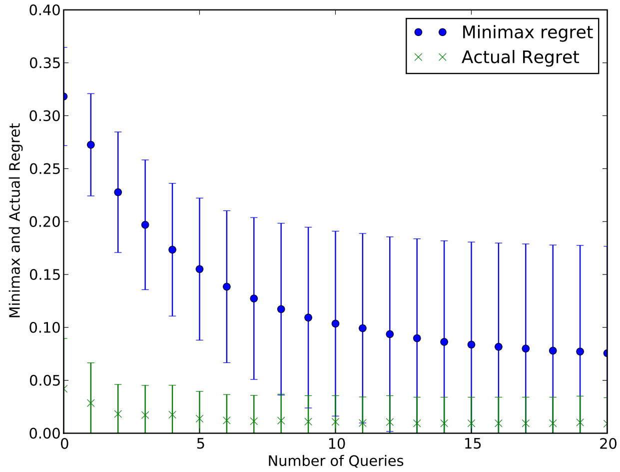

Figure 4: Performance of SGQ-based PE using MEI on an EUT agent.

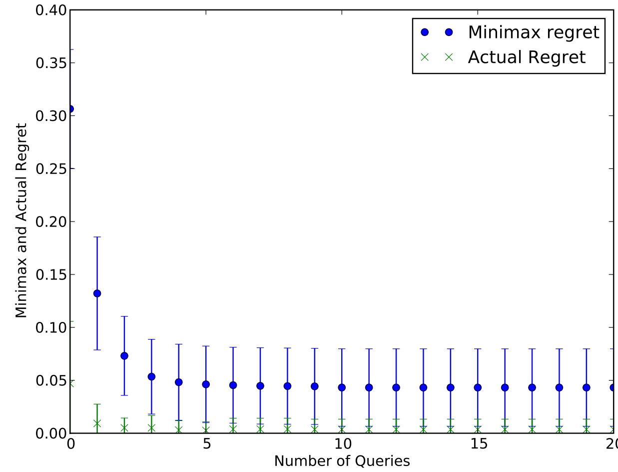

Figure 5: Performance of CPT-based PE using only EMR on a CPT agent.

to suggest a query, the heuristic was unable to get the minimax regret below 0.05. This suggests that MLMR, though sometimes unable to suggest a query, is the more powerful of the two heuristics.

Finally, we were interested in how well the MLMR metric worked when the user's utility function was not of the form in Equation 14. Figure 6 shows the minimax and actual regret for our approach with a user with the logarithmic utility function,

$$u(x) = \log(x + 1).$$

After 7 queries, the minimax regret was below 0.01. We also examined the alternative utility function

$$u(x) = 0.5x^{0.88}.$$

With this utility function the minimax regret was still below 0.01 after 7 queries. This implies that MLMR is flexible enough to deal with different types of utility functions.

7. CONCLUSION AND FUTURE WORK

In this paper, we introduced a querying method that allows the combination of minimax regret preference elicitation and cumulative prospect theory, a descriptive model of human reasoning for risky choices. The main challenge was to design queries that could remove the effects of probability weighting from users' answers. Since choosing optimal queries is a challenging problem, we proposed two heuristics for measuring the value of a query. Our first heuristic, MLMR, was on average efficient but sometimes failed to suggest any query. The EMR heuristic was less efficient but empirically did not fail. The combination of these two metrics provided strong results. Even with preferences following expected utility theory, our approach was more efficient than previously proposed preference elicitation approaches.

Our goal is to be able to apply this approach to real-world preference elicitation situations. It may not be possible for people to answer queries comparing prospects such as f_1 and f_2 with the desired level of accuracy. In this case we may not be able to know the exact value for p^* and we may have to query using both our lower and upper bound constraints on p^*. Additionally, people's utility functions may not always be strictly increasing, and this would make choosing queries more complicated.

8. REFERENCES

[1] C. Boutilier. On the foundations of expected expected utility. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, 2003.

[2] U. Chajewwska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 363–369, Austin, TX, 2000.

[3] L. Chen and P. Pu. Survey of preference elicitation methods. Technical report, Ecole Politechnique Federale de Lausanne, 2004.

[4] D. Kahnemann and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47:263–291, 1979.

[5] A. Mas-Colell, M. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[6] K. Regan and C. Boutilier. Regret-based reward elicitation for Markov decision processes. In *Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence (UAI-09)*, Montreal, Canada, 2009.

[7] C. Starmer. Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk. *Journal of Economic Literature*, 38:332–382, 2000.

[8] A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, October 1992.

[9] P. Wakker and D. Deneffe. Eliciting von Neumann-morgenstern utilities when probabilities are distorted or unknown. *Management Science*, 42:1131–1150, 1996.

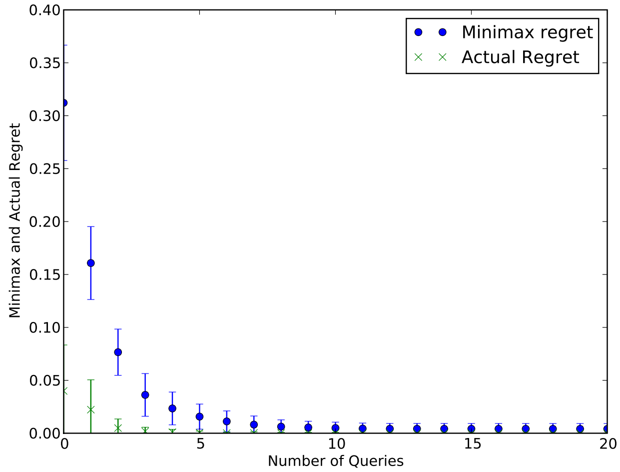

Figure 6: Performance of CPT-based PE with MLMR and EMR on a CPT agent using a log utility function.

[10] T. Wang and C. Boutilier. Incremental utility elicitation with the minimax regret decision criterion. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 309–318, Acapulco, Mexico, 2003.

[11] G. Wu and R. Gonzalez. Curvature of the probability weighting function. *Management Science*, 42:1676–1690, 1996.