



HAL
open science

An Efficient Non-Bayesian Approach for Interactive Preference Elicitation Under Noisy Preference Models

Samira Pourkhajouei, Federico Toffano, Paolo Viappiani, Nic Wilson

► **To cite this version:**

Samira Pourkhajouei, Federico Toffano, Paolo Viappiani, Nic Wilson. An Efficient Non-Bayesian Approach for Interactive Preference Elicitation Under Noisy Preference Models. The 17th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Sep 2023, Arras, France. hal-04292429

HAL Id: hal-04292429

<https://hal.science/hal-04292429>

Submitted on 17 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Efficient Non-Bayesian Approach for Interactive Preference Elicitation Under Noisy Preference Models

Samira Pourkhajouei¹, Federico Toffano¹, Paolo Viappiani², and Nic Wilson¹

¹ Insight Centre for Data Analytics, School of Computer Science and IT,
University College Cork, Cork, Ireland

{samira.pourkhajouei,federico.toffano,nic.wilson}@insight-centre.org

² LAMSADE, CNRS,

Université Paris-Dauphine, Université PSL, 75016, Paris, France

paolo.viappiani@lamsade.dauphine.fr

Abstract. The development of models that can cope with noisy input preferences is a critical topic in artificial intelligence methods for interactive preference elicitation. A Bayesian representation of the uncertainty in the user preference model can be used to successfully handle this, but there are large costs in terms of the processing time required to update the probabilistic model upon receiving the user’s answers, to compute the optimal recommendation and to select the next queries to ask; these costs limit the adoption of these techniques in real-time contexts. A Bayesian approach also requires one to assume a prior distribution over the set of user preference models. In this work, dealing with multi-criteria decision problems, we consider instead a more qualitative approach to preference uncertainty, focusing on the most plausible user preference models, and aim to generate a query strategy that enables us to find an alternative that is optimal in all of the most plausible preference models. We develop a non-Bayesian algorithmic method for recommendation and interactive elicitation that considers a large number of possible user models that are evaluated with respect to their degree of consistency of the input preferences. This suggests methods for generating queries that are reasonably fast to compute. Our test results demonstrate the viability of our approach, including in real-time contexts, with high accuracy in recommending the most preferred alternative for the user.

Keywords: Preference Elicitation · Preference Learning · Decision making · User preference models

1 Incremental elicitation

In the last decade, there has been a huge growth in the use of artificial intelligence technologies in recommending products, entertainment content and services. As a consequence, AI-based recommenders, which act on behalf of users, need adequate mechanisms to assess users’ preferences. Preference elicitation

naturally emerges as having an important role, and approaches that elicit preferences incrementally are particularly suited to AI applications (in contrast with standardised protocols from classic decision theory).

Incremental elicitation methods ask queries to the user in order to acquire new preference information. The uncertainty over the user’s preference model (often represented by a parameterised utility function) is gradually reduced as the user answers more queries. Queries are generated adaptively, i.e., they do not follow a fixed protocol, but, at each stage of the interaction the system may select the “best” query given what it already knows about the user (the nature of the best query depends on the specific approach). The interaction ends either when an optimal solution is found (the information provided by the user allows the system to infer optimality), or until a termination condition is met, for instance when some notion of loss is lower than a threshold, or when exceeding some notion of cognitive or time cost, or because of the user’s fatigue. In the case of early termination, the system should be able to provide a recommendation based on the preference information that has been elicited.

Methods for interactive elicitation typically represent the uncertainty about the user’s preference model in some principled way. In several works, the parameter space is reduced at each step by converting the new information into a constraint on some utility parameters. This is the approach taken by methods based on *minimax regret* [8]. These methods are efficient since updating the model is quick: whenever a query is answered, the space of feasible parameters is reduced. But in the case of an erroneous answer, strict constraints on the preference state space may exclude the true user preference model; thus, the quality of the resulting recommendation may be abysmal.

A way to overcome this issue is to use probabilistic approaches that allow one to deal with the uncertainty about the decision-maker’s answers. In such Bayesian approaches, the uncertainty about the real parameter value is represented by a probability distribution that is updated when new preference statements are collected and a noisy *response model* accounts for the possibility that a decision-maker may make a choice that does not maximise their utility. In [22] and [5] the authors introduce an incremental preference elicitation procedure able to deal with noisy responses of a user. They propose a Bayesian approach for choosing a preferred solution among a set of alternatives in a multi-criteria decision problem. However, the Bayesian approach can be computationally expensive, making it difficult to use in real-time contexts, and it also makes assumptions about a prior probability distribution over preference models.

This paper presents a new, non-Bayesian, incremental preference elicitation technique that can handle noisy responses. We want to deal with a situation in which occasionally the user responses are inaccurate. We use a more qualitative representation of uncertainty, focusing on the set of the most plausible user models, which are those that are the most consistent with the answers to the queries. Our purpose is to develop a method that is robust to incorrect input preferences from the decision-maker, but still relatively efficient in terms of number of queries required, and computational time to generate the queries.

The rest of the paper is organised as follows. We next, in Section 2, discuss the related work; then, in Section 3 we define the formal settings including the terminology. Section 4 describes our approach in detail, including the stopping criterion for the algorithm. The query generation approach is described in Section 5, and Section 6 presents computational results showing how the methods perform, and include also a comparison with a Bayesian approach. Section 7 concludes.

A longer version of the paper is available online [13]; this includes more discussion and experimental results.

2 Related Work

It is becoming important to develop methods that identify and assess user preferences. Research on preference elicitation has been widely conducted in decision analysis and artificial intelligence. As part of decision analysis [11,15,23] and artificial intelligence [2,7,9,10], automated decision support software is being developed. In this context, an active elicitation of a decision-maker's preferences can be crucial for user satisfaction. An automated agent can actively elicit decision-maker's preferences by asking queries about their preferences [12,15,7].

Many methods for active preference elicitation have been developed, where the decision support system explicitly queries the decision-maker about her preferences. Previous works can be classified according to two main classes of models of preference uncertainty and optimisation.

In the *robust approach*, recommendations are generated according to the minimax-regret criterion [8,3]; the system ask queries that are likely to decrease regret.

In the *Bayesian approach* to elicitation, the system maintains a distribution over the utility function's parameters and that is updated using the Bayes rule whenever new information is received from the decision-maker (as answers to queries). Choosing queries is primarily based on expected value of information and the alternative with the highest expected utility is recommended [4,21,22].

The Minimax regret method is applied both as a recommendation criterion as well as a technique for driving elicitation in a variety of settings, but it fails to tolerate user inconsistency. In an ideal setting, a decision-maker would always select the item with the highest utility with respect to her true utility function and never commits mistakes [14,3]. However, this is not realistic in general, and in learning a user utility function one needs to deal with uncertain and possibly inconsistent feedback from the decision-maker. The Bayesian framework can include user noise in the elicitation process. This is done by building a probability distribution, exploiting prior information, reasoning about the likelihood of user responses, and recommending options that are optimal in expectation [6,22,16,21]. The problem with this approach is that it is computationally expensive, does not scale and may even not be feasible in many scenarios. Our goal is to use a non-Bayesian approach to handle uncertainty in the elicitation and to be able to deal with noisy responses of a decision-maker.

Another non-Bayesian approach is based on a possibilistic extension of regret [1], though it is also computationally expensive. Finally we mention an interactive elicitation method based on maximum margin [18]; this method is interesting as it is resistant to noise, but it is, however, focused on configuration domains, while we focus on settings where alternatives are explicitly given in a dataset, as in multiple-criteria decision-making.

3 Problem setting

Let us suppose that a system is assigned the task of recommending an option to a user among a finite set A of alternatives. Alternative $\alpha \in A$ is characterized by a vector $(\alpha(1), \dots, \alpha(p))$ where each $\alpha(i)$ represents the value of the alternative α with respect to criterion (or objective) i . For convenience (and without loss of generality) we assume that the scales are arranged so that higher values of a criterion are better.

Utility function. We assume the decision-maker has a utility function $u : (\mathcal{W}, A) \rightarrow \mathbb{R}$ which is parameterised by some parameter vector $w \in \mathcal{W}$, where \mathcal{W} is the space of user preferences defined as the set of all the normalised non-negative weights vectors w , $\{w \in \mathbb{R}^p : \sum_{i=1}^p w(i) = 1; w(i) \geq 0; \forall i = 1, \dots, p\}$. In particular, we assume that the decision-maker's utility function evaluating alternatives is the weighted sum of the vector of criteria, with the weights vectors $w \in \mathcal{W}$ representing the possible decision-maker preferences. Given a vector of weights $w \in \mathcal{W}$, an alternative α has then a utility value $u(\alpha, w) = \sum_{i=1}^p w(i)\alpha(i)$. Let w^* be the true preferences of a decision-maker; this is unknown to the decision support system (and also typically unknown to the decision maker). The preference statement $\alpha \succ \beta$ represents a decision-maker preference of alternative α over alternative β . Thus, for a particular decision-maker with preferences w^* , $\alpha \succ \beta \iff u(\alpha, w^*) \geq u(\beta, w^*)$.

Our goal is to find the most preferred alternative of a decision-maker, i.e., $\arg \max_{\alpha \in A} u(\alpha, w^*)$, without showing all the possible alternatives, and without knowing the true user preference w^* .

Example: Consider a scenario in which a decision-maker wants to select a house from a list of houses that are available to rent as follows:

$$A = \{\alpha = (12.7, 5, 3), \beta = (13, 3, 2), \gamma = (10.5, 3, 2)\}.$$

The utility of each house is represented with a vector $(\alpha(1), \alpha(2), \alpha(3))$ representing monthly rent, distance from the city centre and the number of bedrooms, where the values of each criterion have been scaled so that the higher the value of each criterion, the better. Assume that the decision-maker uses a weighted sum model with vector of weights $w^* = (0.7, 0.1, 0.2)$. The utility function $u(\alpha_i, w^*) \in \mathbb{R}$ returns a real number representing the decision-maker score for the corresponding house. The most preferred house will then be the one with the highest score. In this example, we know that the most preferred alternative of the decision maker is α since $u(\alpha, w^*) = 12.7 \times 0.7 + 5 \times 0.1 + 3 \times 0.2 = 9.99$, $u(\beta, w^*) =$

$13 \times 0.7 + 3 \times 0.1 + 2 \times 0.2 = 9.8$ and $u(\gamma, w^*) = 10.5 \times 0.7 + 3 \times 0.1 + 2 \times 0.2 = 8.05$. In this example we are assuming that we know the decision-maker preference model. However, in the real world we don't know the weights vector representing the decision-maker preferences, but our interactive preference elicitation method can be used to estimate it, and to discover their optimal alternative.

Possibly Optimal alternatives. We say that an alternative α is optimal in a set of alternatives A with respect to weights vector w if and only if $u(\alpha, w) \geq u(\beta, w)$ for any $\beta \in A$. An alternative $\alpha \in A$ is *possibly optimal* in A , with respect to a set \mathcal{W} of weights vectors, if and only if there exists $w \in \mathcal{W}$ such that $u(\alpha, w) \geq u(\beta, w)$ for all $\beta \in A$, i.e., such that α is optimal in A with respect to w . We define $\text{PO}(A, \mathcal{W})$ as the set of all possibly optimal alternatives in A with respect to \mathcal{W} .

We can compute $\text{PO}(A, \mathcal{W})$ with a linear programming solver (see, e.g., [20]). Briefly, we can test if $\alpha \in \text{PO}(A, \mathcal{W})$ evaluating the feasibility of the set of linear constraints $u(\alpha, w) \geq u(\beta, w)$ for all $\beta \in A \setminus \{\alpha\}$ with $w \in \mathcal{W}$. We focus only on the alternatives in $\text{PO}(A, \mathcal{W})$ because the decision-maker's most preferred alternative must be optimal for the true preference w^* . Thus, we do not need to consider alternatives $\beta \notin \text{PO}(A, \mathcal{W})$ and these alternatives can be filtered out as pre-processing.

Queries. Our approach focuses on binary queries, i.e., on asking the decision-maker to express their preferences with respect to pairwise comparisons of alternatives. We define a query as a pair (α, β) with $\alpha, \beta \in A$ (with $\alpha \neq \beta$), and the corresponding question for the decision-maker is ‘Do you prefer α or β ?’. With a query (α, β) the decision-maker prefers α if and only if $w^* \cdot (\alpha - \beta) \geq 0$, and so otherwise, if $w^* \cdot (\alpha - \beta) < 0$ then the decision-maker prefers β . Many incremental preference elicitation procedures (see, e.g., [24,17,22,19]) iteratively ask this type of query with the purpose of reducing the space of feasible weights vectors by adding such hard constraints. However, a drawback of adding hard constraints to the set of feasible weights vectors is that we may exclude the optimal preference vector w^* if we receive an incorrect decision-maker answer.

User model. We assume a simple form of user model with two parameters: the preference vector $w^* \in \mathcal{W}$ and the noise parameter ρ with $0 \leq \rho < 1$, e.g., $\rho = 0.1$. Given a query (α, β) , with probability $1 - \rho$ the user will answer correctly, i.e., answer α if and only if $w^* \cdot (\alpha - \beta) \geq 0$, and answer β otherwise; with probability ρ , the user will answer incorrectly, answering β iff $w^* \cdot (\alpha - \beta) \geq 0$. Note that neither w^* nor ρ are known by the learning system (only the answers to the queries).

Simplifying assumption. We assume that, for each preference vector w , there is a unique element α_w in A that maximises $u(\alpha, w)$. With A consisting of random real-valued vectors this will almost certainly hold, and the assumption considerably simplifies the notation and the description of the algorithms. All the methods can be easily extended for situations in which this does not hold.

4 The Idea Behind Our Approach

If it were the case that there exists a unique possibly optimal alternative α in A , i.e., if $\text{PO}(A, \mathcal{W}) = \{\alpha\}$, then clearly we should recommend α , since it is optimal in A with respect to every preference vector w in \mathcal{W} (and thus, in particular, with respect to the unknown user preference vector w^*). Similarly, if we were certain that the user was always answering our queries correctly, and \mathcal{U} is the set of preference vectors compatible with the user’s answers, and $\text{PO}(A, \mathcal{U}) = \{\alpha\}$, then we should recommend α . In our context, where we are never certain about the correctness of the user’s answers, we can adapt this idea by considering $\text{PO}(A, \mathcal{U}')$, where \mathcal{U}' is the set of most plausible preference vectors. We are making no assumptions about a prior distribution over \mathcal{W} , so the plausibility of a preference vector w relates to how closely the user’s answers are to those that would have been given if w were the true user preference vector w^* (and the user gave accurate answers). Thus, for a given query (α, β) with answer α , we test if $w \cdot (\alpha - \beta) \geq 0$ to check if α is preferred to β according to the weights vector w . We suppose that the more inequalities are satisfied for a given $w \in \mathcal{W}$, the more plausible it is that w has a similar preference order to that of the true decision-maker preference. This is formalised with the function $\text{mistakes}(\cdot)$.

The function $\text{mistakes}(w)$. To find the preference vectors in \mathcal{W} that corresponds most closely to the decision-maker input preferences, we count the number $\text{mistakes}(w)$ of mistakes that the decision-maker would have made if $w \in \mathcal{W}$ were the true user preference vector w^* , i.e., the number of times the inequality $w \cdot (\alpha - \beta) \geq 0$ is not satisfied, for each query (α, β) (or (β, α)) with answer α . For example, with a query (α, β) , if the decision-maker answers α , and $w \cdot (\alpha - \beta) < 0$, we increment $\text{mistakes}(w)$ by one unit.

Because the user’s answers can be incorrect, $\text{mistakes}(w^*)$ will often be greater than zero. In particular, because we are considering a simple noisy user model, with a chance ρ of giving an incorrect answer independently for each query, the random variable $\text{mistakes}(w^*)$ is binomially distributed with expected value ρK , where K is the number of queries asked.

Of course, we do not know $\text{mistakes}(w^*)$ since the true user preference w^* is unknown; however, we can consider the set \mathcal{W}_{min}^k of the most plausible preference vectors, including w such that $\text{mistakes}(w)$ is within the threshold k of the minimal number of mistakes (defined below).

Finite approximation \mathcal{W}' of \mathcal{W} . It is computationally convenient to approximate \mathcal{W} by a finite set of points \mathcal{W}' . There are various ways this can be done; in our experiments we randomly sample elements of \mathcal{W} using a uniform distribution over the probability simplex \mathcal{W} ; currently we use the same set \mathcal{W}' throughout the whole iterative interaction process.

The set \mathcal{W}_{min}^k of k -plausible preference vectors. Let μ be the minimum number of $\text{mistakes}(w)$ over all $w \in \mathcal{W}'$. For parameter $k \geq 0$ define \mathcal{W}_{min}^k to be all the preference points $w \in \mathcal{W}'$ such that $\text{mistakes}(w) \leq \mu + k$. We use

\mathcal{W}_{min}^k to generate queries for the decision-maker. (Note that \mathcal{W}_{min}^k depends on the randomly chosen set \mathcal{W}' , although our notation does not make this explicit.)

We say that w and w^* differ on a query (α, β) if either (a) $w \cdot \alpha \geq w \cdot \beta$ and $w^* \cdot \alpha < w^* \cdot \beta$; or (b) $w^* \cdot \alpha \geq w^* \cdot \beta$ and $w \cdot \alpha < w \cdot \beta$. The result below throws some light on how the set \mathcal{W}_{min}^k will look after a sequence of queries.

Proposition 1. *Let $w \in \mathcal{W}$. Consider a sequence of queries, and let K_w be the number of queries in the sequence in which w and w^* differ. Let the random variable $\mathbf{X}_w^{w^*}$ be equal to $mistakes(w) - mistakes(w^*)$. Then $\mathbf{X}_w^{w^*} \sim K_w - 2B(K_w, \rho)$, where $B(K_w, \rho)$ is a binomial distribution with K_w experiments and probability of success ρ . The expected value $E[\mathbf{X}_w^{w^*}]$ of $\mathbf{X}_w^{w^*}$ is equal to $K_w(1 - 2\rho)$, and the standard deviation of $\mathbf{X}_w^{w^*}$ equals $2\sqrt{K_w\rho(1 - \rho)}$.*

Proof: Let us label the queries in the sequence on which w and w^* differ as (α_i, β_i) for $i = 1, \dots, K_w$, and let the Boolean random variable \mathbf{Z}_i be such that $\mathbf{Z}_i = 1$ if and only if the user answers query (α_i, β_i) incorrectly. The variables \mathbf{Z}_i for $i = 1, \dots, K_w$ are independent with $\Pr(\mathbf{Z}_i = 1) = \rho$. If $\mathbf{Z}_i = 1$ then $mistakes(w^*)$ is incremented and $mistakes(w)$ is unchanged, so $mistakes(w) - mistakes(w^*)$ is decremented; and if $\mathbf{Z}_i = 0$ then $mistakes(w)$ is incremented and so $mistakes(w) - mistakes(w^*)$ is incremented. So, in both cases, $\mathbf{X}_w^{w^*}$ changes by $1 - 2\mathbf{Z}_i$. (For the other queries, on which w and w^* do not differ, $mistakes(w) - mistakes(w^*)$ is unchanged.) Thus, $\mathbf{X}_w^{w^*} = mistakes(w) - mistakes(w^*)$ is equal to $\sum_{i=1}^{K_w} (1 - 2\mathbf{Z}_i)$, i.e., $K_w - 2\sum_{i=1}^{K_w} \mathbf{Z}_i$. Therefore, $\mathbf{X}_w^{w^*} \sim K_w - 2B(K_w, \rho)$, because $\sum_{i=1}^{K_w} \mathbf{Z}_i$ has the binomial distribution $B(K_w, \rho)$.

The expected value of $\sum_{i=1}^{K_w} \mathbf{Z}_i$ is $K_w\rho$, and so $E[\mathbf{X}_w^{w^*}] = K_w(1 - 2\rho)$. The variance of \mathbf{Z}_i is equal to $E[(\mathbf{Z}_i)^2] - (E[\mathbf{Z}_i])^2 = \rho - \rho^2$, and so the variance of $\mathbf{X}_w^{w^*}$, which equals the variance of $2\sum_{i=1}^{K_w} \mathbf{Z}_i$, is $4K_w\rho(1 - \rho)$; hence the standard deviation of $\mathbf{X}_w^{w^*}$ is equal to $2\sqrt{K_w\rho(1 - \rho)}$. \square

Proposition 1 implies that after a number of queries, the w (in \mathcal{W}') with minimal K_w will tend to be in \mathcal{W}_{min}^k . In particular, if w^* were in \mathcal{W}' and K_w is reasonably large, then it is very unlikely that w will be in \mathcal{W}_{min}^k for small k such as $k \in \{0, 1, 2\}$, especially so for larger K_w . This is because if w were in \mathcal{W}_{min}^k then we would have $\mathbf{X}_w^{w^*} \leq k$, which would make the approximately normally distributed random variable $\mathbf{X}_w^{w^*}$ at least $\frac{K_w(1-2\rho)-k}{2\sqrt{K_w\rho(1-\rho)}}$ standard deviations from its mean. For instance, with $k = 2$ and $\rho = 0.1$ and $K_w = 10$ (respectively, $K_w = 15$), $\mathbf{X}_w^{w^*}$ will be more than 3 (respectively, 4.3) standard deviations from its mean.

More generally, the weights vectors w that order the alternatives in A most similarly to how w^* orders them, will tend to be in \mathcal{W}_{min}^k , increasingly so as we ask more queries. Therefore, the most plausible user models w , i.e., those most likely to be the true user preference model w^* (or close to it), are those with smaller values of $mistakes(w)$, which is why \mathcal{W}_{min}^k may be considered as consisting of the most plausible user preference models.

Stopping criterion. Our algorithm makes use of \mathcal{W}_{min}^k , for $k = 0, \dots, \kappa$ where we focus on $\kappa = 2$ in our experimental testing. The stopping criterion for our algorithm is that all the most plausible user preference models agree on which alternative α is best, i.e., $\text{PO}(A, \mathcal{W}_{min}^\kappa) = \{\alpha\}$. With sufficient appropriately chosen queries this can be made to hold eventually (with probability tending to one). Generally our query strategies aim to ensure that the stopping criterion is satisfied as soon as possible. In particular, we can limit ourselves to queries of the form (α_v, α_w) , where $v, w \in \mathcal{W}_{min}^\kappa$, since there always exists such a query if the stopping criterion is not yet satisfied, and such a query will with probability $1 - \rho$ increment the mistakes function for w , if w differs with w^* on this query (and thus, v agrees with w^*). If we were to keep repeating this query then, with high probability, w will be eliminated from \mathcal{W}_{min}^κ .

Our active learning method is summarized by Algorithm 1.

We select the query using the method described in Section 5. The decision-maker response will be used to update $mistakes(w)$ for each $w \in \mathcal{W}$ and to recompute \mathcal{W}_{min}^k for $k \in \{0, 1, 2\}$. We iterate this procedure until $\text{PO}(A, \mathcal{W}_{min}^2)$ becomes a singleton set, say $\{\alpha\}$ (so also, $\text{PO}(A, \mathcal{W}_{min}^1) = \text{PO}(A, \mathcal{W}_{min}^0) = \{\alpha\}$); then α is our estimation of the most preferred alternative of the decision-maker, and we recommend it. Intuitively, this loop will tend to exclude weights vectors from \mathcal{W}_{min}^k for $k \in \{0, 1, 2\}$ whose preference orders are very different from the preferences of the decision-maker. Thus, weights vector in \mathcal{W}_{min}^k for $k \in \{0, 1, 2\}$ are more likely to have similar preference orders of that of w^* .

Algorithm 1

```

1: procedure RECOMMEND_ALTERNATIVE( $A, \mathcal{W}'$ )
2:   repeat
3:      $q \leftarrow \text{Select\_query}(A, \mathcal{W}_{min}^k \text{ for } k \in \{0, 1, 2\})$ 
4:     Ask query  $q$  to the decision-maker
5:     Update  $\mathcal{W}_{min}^k$  for  $k \in \{0, 1, 2\}$ 
6:   until  $|\text{PO}(A, \mathcal{W}_{min}^2)| = 1$ 
7:   return The unique alternative in  $\text{PO}(A, \mathcal{W}_{min}^2)$ 

```

5 Query selection

We select as a query a pair of alternatives (α, β) that are optimal in A with respect to a maximum number of $w \in \mathcal{W}_{min}^k$, focusing first on lower values of k . More precisely, let $Opt^k(\alpha)$ be the number of preference points $w \in \mathcal{W}_{min}^k$ with α as the most preferred alternative α_w with respect to w , i.e., maximising $u(\alpha, w)$. We select the query (α, β) as follows:

- If $|\text{PO}(A, \mathcal{W}_{min}^0)| > 1$, select a query (α, β) with $\alpha, \beta \in \text{PO}(A, \mathcal{W}_{min}^0)$, $Opt^0(\alpha) \geq Opt^0(\gamma)$ and $Opt^0(\beta) \geq Opt^0(\gamma)$ for each $\gamma \in \text{PO}(A, \mathcal{W}_{min}^0) \setminus \{\alpha, \beta\}$.

- If $\text{PO}(A, \mathcal{W}_{min}^0) = \{\alpha_0\}$ and $|\text{PO}(A, \mathcal{W}_{min}^1)| > 1$, select a query (α_0, β) with $\beta \in \text{PO}(A, \mathcal{W}_{min}^1)$, $\beta \neq \alpha_0$ and $Opt^1(\beta) \geq Opt^1(\gamma)$ for each $\gamma \in \text{PO}(A, \mathcal{W}_{min}^1) \setminus \{\alpha_0, \beta\}$.
- If $\text{PO}(A, \mathcal{W}_{min}^0) = \text{PO}(A, \mathcal{W}_{min}^1) = \{\alpha_0\}$ and $|\text{PO}(A, \mathcal{W}_{min}^2)| > 1$, select a query (α_0, β) with $\beta \in \text{PO}(A, \mathcal{W}_{min}^2)$, $\beta \neq \alpha_0$ and $Opt^2(\beta) \geq Opt^2(\gamma)$ for each $\gamma \in \text{PO}(A, \mathcal{W}_{min}^2) \setminus \{\alpha_0, \beta\}$.

6 Experimental Results

In this section we discuss the results of the experimental testing of our approach applied to randomly generated decision problems.

A random problem is represented by a random set A of possibly optimal utility vectors³ and a simulated decision-maker with utility function $u(\alpha, w^*)$. The goal is to find the most preferred alternative of the decision-maker, i.e., the alternative $\alpha_{w^*} \in A$ maximising $u(\alpha, w^*)$ for any $\alpha \in A$. We simulate a decision-maker for each experiment generating a random weights vector w^* . With a noise-free user model, the simulated decision-maker response to a comparison query of two alternatives (α, β) will be α if $w^* \cdot \alpha \geq w^* \cdot \beta$, and β otherwise. However, we want to simulate noisy user responses, therefore we take into account a fixed probability ρ (e.g., $\rho = 0.1$) of receiving the incorrect answer.

In Table 1 we show the average number of queries, the average iteration time and the accuracy. The accuracy is the fraction of experiments in which the correct alternative was recommended, i.e., the optimal alternative α_{w^*} in A according to the unknown true user model w^* . The results are an average of 100 experiments with random sets \mathcal{W}' of 4000 weights vectors, and input sets A of 1000 random possibly optimal alternatives. We always used the same set A of alternatives, and 100 random user models w^* . The accuracy was high and we always had at least one $w \in \mathcal{W}'$ with $\alpha_w = \alpha_{w^*}$.

Table 1: Experimental results w.r.t. the number of criteria p , with $|A| = 1000$, $\rho = 0.1$ and $|\mathcal{W}'| = 4000$.

| p | Queries | Time[s] | Accuracy |
|-----|---------|---------|----------|
| 3 | 10.66 | 0.038 | 1.00 |
| 4 | 22.16 | 0.038 | 1.00 |
| 5 | 30.44 | 0.038 | 0.97 |
| 6 | 35.88 | 0.038 | 1.00 |

In general, as the number of criteria increases, so does the number of queries.

³ Only the possibly optimal alternatives in a set A of alternatives are relevant, so if we didn't enforce that all alternatives are possibly optimal, then we would effectively be dealing with a (perhaps very much) smaller problem.

Regarding the iteration time, the method seems to be roughly independent of the number of criteria. This is because the most time-consuming operation in this case is the update of $mistakes(w)$ for each $w \in \mathcal{W}$, which is not affected much by the number of criteria since the most expensive operation is the computation of $|\mathcal{W}'|$ dot products. For example, with $p = 5$, we required on average 0.34ms to compute the query, 36.2ms to update $mistakes(w)$ for each $w \in \mathcal{W}$ and 1.05ms to compute the sets \mathcal{W}_{min}^k .

Table 2: Experimental results for regret-based elicitation with the current solution query strategy; $|A| = 1000$ and $\rho = 0.1$.

| p | Queries | Time[s] | Accuracy |
|-----|---------|---------|----------|
| 3 | 4.58 | 4.12 | 0.60 |
| 4 | 7.82 | 3.41 | 0.34 |
| 5 | 11.93 | 3.66 | 0.34 |
| 6 | 16.69 | 3.69 | 0.28 |

We provide, for comparison, the simulation results obtained by using state-of-the-art elicitation methods on the same datasets. In Table 2 we show the performance of interactive elicitation based on minimax regret with queries generated using the current solution strategies. Unsurprisingly, regret-based elicitation provides low accuracy as it cannot appropriately deal with user noise.

Table 3: Experimental results for Bayesian elicitation with the queries generated with greedy maximisation of value of information; $|A| = 1000$ and $\rho = 0.1$.

| p | Queries | Time[s] | Accuracy |
|-----|---------|---------|----------|
| 3 | 12.05 | 2.62 | 0.97 |
| 4 | 17.93 | 3.68 | 0.99 |
| 5 | 26.46 | 5.07 | 0.97 |
| 6 | 35.86 | 5.50 | 1.00 |

In Table 3 we consider the Bayesian elicitation method from [22]; queries are chosen to maximise Expected Value of Selection (EUS), a proxy of myopic value of information, using greedy maximisation; Bayesian updates are performed using Monte Carlo methods with 50000 particles. The elicitation stops when the expected loss is less than 0.001; when this happens the alternative with highest expected utility is recommended. As the table shows, the Bayesian method achieves high accuracy but at the cost of large computation times (higher accuracy may be obtained using more particles, but compaction time will increase even further).

In Table 4 we show the performances of our method with respect to the size of \mathcal{W}' . The accuracy increases with increasing $|\mathcal{W}'|$, as does the computation time, and, to a lesser extent, the average number of queries.

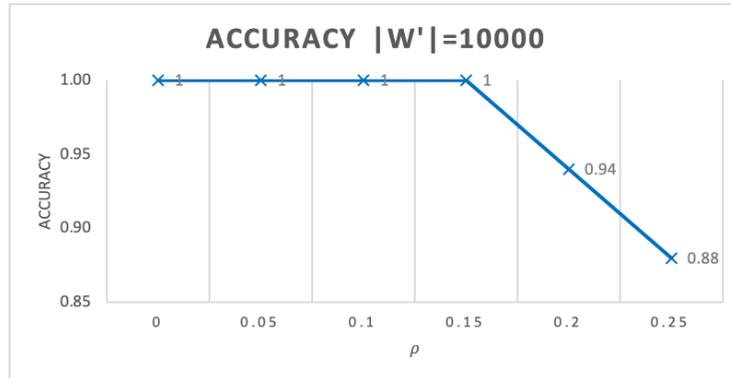
Table 4: Experimental results w.r.t. the number $|\mathcal{W}'|$ of user preference models, with $|A| = 1000$, $\rho = 0.1$ and $p = 5$.

| $ \mathcal{W}' $ | Queries | Time[s] | Accuracy |
|------------------|---------|---------|----------|
| 2000 | 27.28 | 0.019 | 0.97 |
| 4000 | 30.44 | 0.038 | 0.97 |
| 6000 | 32.71 | 0.057 | 1.00 |
| 8000 | 33.12 | 0.075 | 0.99 |
| 10000 | 34.50 | 0.096 | 1.00 |

We also tested the performances with respect to the number of alternatives $|A| \in \{200, 400, 600, 800, 1000\}$ with $|\mathcal{W}'| = 4000$, $p = 4$ and $\rho = 0.1$. However, we didn't notice any significant difference in terms of accuracy and execution time. The average number of queries was affected slightly more, i.e., between 20.67 and 22.16, with lower values for lower $|A|$.

In Figure 1 we show the accuracy varying with respect to the user noise ρ , with $|A| = 1000$, $|\mathcal{W}'| = 10000$ and $p = 4$. Unsurprisingly, the accuracy decreases with increasing user noise ρ . However, this picture shows that our model can achieve good performance also with more noisy responses. In this case, the average query time was 0.093s. This dropping off of the accuracy for larger ρ tallies with our analysis around Proposition 1, and, to maintain very high accuracy, we will need to increase the parameter κ in our algorithm (from its current value of 2), in order to make the stopping condition harder to satisfy.

Fig. 1: Accuracy varying with user noise ρ over 100 experiments with $|A| = 1000$, $|\mathcal{W}'| = 10000$ and $p = 4$.



7 Conclusions and Discussion

We have described a novel, non-Bayesian, approach for interactive elicitation for a setting in which the user's answers are not completely reliable. Our approach is

based on maintaining a set of plausible preference models and to reason about the alternatives that are optimal according to these preference models. We provide fast and effective methods for generating comparison queries. In our model the stopping criterion is defined so that the system finishes the interaction, and recommends an alternative α , when α is the optimal alternative in all the most plausible preference models. The notion of plausibility of a preference model is based on how close the answers from that model would be to the received answers. For computational reasons we focus attention on a finite approximation \mathcal{W}' of the set of all preference models.

Our results show that the approach is very fast and suitable for real-time applications, and maintains good accuracy with reasonable lengths of interactions. We have also compared our approach with the Bayesian approach from [22], and our approach is very much faster, and with similar or perhaps slightly higher accuracy, and with similar number of queries required.

A variation of our approach that may further increase the accuracy, but at some computational cost, would be to update the finite set \mathcal{W}' of preference models as the interaction progresses, so that the models become more densely populated in areas where they are most plausible.

Our stopping criterion, that a single alternative α in the set of alternatives A is optimal in all the most plausible user models \mathcal{W}_{min}^κ , is equivalent to the max regret of α (over $w \in \mathcal{W}_{min}^\kappa$) being zero. We can weaken this condition by instead enforcing that this minimum max regret is less than a small threshold ϵ , which will allow our method to be applied to some more complex situations and with different user models, and perhaps reducing the number of interactions with the decision-maker.

We have shown that our approach can deal with significant numbers of alternatives, and the number of alternatives does not appear to very strongly affect the computation time or performance of the algorithm. A natural further step would be to develop the approach for combinatorial problems, where there are an exponential number of alternatives. In this case, for each preference model w in the finite set \mathcal{W}' , we determine an optimal alternative α_w of the combinatorial problem with respect to the linear objective function given by w , and again the queries can be based on the possibly optimal elements with respect to the most plausible user models \mathcal{W}_{min}^κ .

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund; and by the EU *Network of Excellence* project, TAILOR.

References

1. Adam, L., Destercke, S.: Possibilistic preference elicitation by minimax regret. In: de Campos, C.P., Maathuis, M.H., Quaeghebeur, E. (eds.) Proceedings of the

- Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021. Proceedings of Machine Learning Research, vol. 161, pp. 718–727. AUAI Press (2021)
2. Blythe, J.: Visual exploration and incremental utility elicitation. In: Eighteenth National Conference on Artificial Intelligence. p. 526–532. American Association for Artificial Intelligence, USA (2002)
 3. Bourdache, N., Perny, P.: Anytime algorithms for adaptive robust optimization with OWA and WOWA. In: Rothe, J. (ed.) Algorithmic Decision Theory - 5th International Conference, ADT 2017. Lecture Notes in Computer Science, vol. 10576, pp. 93–107. Springer (2017)
 4. Bourdache, N., Perny, P., Spanjaard, O.: Incremental elicitation of rank-dependent aggregation functions based on bayesian linear regression. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 2023–2029 (2019). <https://doi.org/10.24963/ijcai.2019/280>
 5. Bourdache, N., Perny, P., Spanjaard, O.: Bayesian preference elicitation for multi-objective combinatorial optimization. In: DA2PL 2020 - From Multiple Criteria Decision Aid to Preference Learning. Trento, Italy (Nov 2020), <https://hal.archives-ouvertes.fr/hal-02979845>
 6. Bourdache, N., Perny, P., Spanjaard, O.: Bayesian preference elicitation for multiobjective combinatorial optimization. CoRR **abs/2007.14778** (2020), <https://arxiv.org/abs/2007.14778>
 7. Boutilier, C.: A POMDP formulation of preference elicitation problems. In: Proceedings of AAAI02. pp. 239–246 (2002)
 8. Boutilier, C.: Computational decision support: Regret-based models for optimization and preference elicitation (2012). <https://doi.org/10.1093/acprof:oso/9780199856800.003.0041>
 9. Chajewska, U., Getoor, L., Norman, J., Shahar, Y.: Utility elicitation as a classification problem. In: UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. pp. 79–88. Morgan Kaufmann (1998)
 10. Chajewska, U., Koller, D., Parr, R.: Making rational decisions using adaptive utility elicitation. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. pp. 363–369 (2000)
 11. Dyer, J.S.: Interactive goal programming. *Manage. Sci.* **19**(1), 62–70 (sep 1972). <https://doi.org/10.1287/mnsc.19.1.62>, <https://doi.org/10.1287/mnsc.19.1.62>
 12. Keeney, R., Raiffa, H., L, K., Meyer, R.: Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Wiley series in probability and mathematical statistics. Applied probability and statistics, Cambridge University Press (1993), <https://books.google.ie/books?id=GPE6ZAqGrnC>
 13. Pourkhajouei, S., Toffano, F., Viappiani, P., Wilson, N.: An efficient non-Bayesian approach for interactive preference elicitation under noisy preference models (longer version) (2023), <http://ucc.insight-centre.org/nwilson/ECSQARU23longer.pdf>
 14. Price, R., Messinger, P.R.: Optimal recommendation sets: Covering uncertainty over user preferences. In: AAAI (2005)
 15. Salo, A., Hämäläinen, R.P.: Preference ratios in multiattribute evaluation (prime)-elicitation and decision procedures under incomplete information. *IEEE Trans. Syst. Man Cybern. Part A* **31**, 533–545 (2001)
 16. Sauré, D., Vielma, J.P.: Ellipsoidal methods for adaptive choice-based conjoint analysis. *Oper. Res.* **67**(2), 315–338 (mar 2019). <https://doi.org/10.1287/opre.2018.1790>

17. Steuer, R.E., Choo, E.U.: An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming* **26**(3), 326–344 (1983)
18. Teso, S., Passerini, A., Viappiani, P.: Constructive preference elicitation by set-wise max-margin learning. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*. pp. 2067–2073 (2016)
19. Toffano, F., Viappiani, P., Wilson, N.: Efficient exact computation of setwise min-max regret for interactive preference elicitation. In: *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems*. pp. 1326–1334 (2021). <https://doi.org/10.5555/3463952.3464105>
20. Toffano, F., Wilson, N.: Minimality and comparison of sets of multi-attribute vectors. *Autonomous Agents and Multi-Agent Systems* **36**(2), 1–66 (2022)
21. Vendrov, I., Lu, T., Huang, Q., Boutilier, C.: Gradient-based optimization for bayesian preference elicitation. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(06), 10292–10301 (2020). <https://doi.org/10.1609/aaai.v34i06.6592>
22. Viappiani, P., Boutilier, C.: On the equivalence of optimal recommendation sets and myopically optimal query sets. *Artificial Intelligence* **286**, 103328 (2020). <https://doi.org/https://doi.org/10.1016/j.artint.2020.103328>, <https://www.sciencedirect.com/science/article/pii/S0004370220300849>
23. White III, C.C., Sage, A.P., Dozono, S.: A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Trans. Syst. Man Cybern.* **14**(2), 223–229 (1984). <https://doi.org/10.1109/TSMC.1984.6313205>
24. Zionts, S., Wallenius, J.: An interactive programming method for solving the multiple criteria problem. *Management science* **22**(6), 652–663 (1976)