Value vs Uncertainty in Real-time Preference Elicitation

Marco Faella[1] and **Alberto Finzi**[2] and **Luigi Sauro**[3]

Abstract. Learning users' preferences plays a central role in AI-supported decision making. We present a Bayesian preference elicitation framework supporting arbitrary utility functions on a multi-attribute domain of alternatives. We experimentally compare different myopic measures for optimal query selection and show that a suitable entropy-based measure can achieve the same loss reduction of a classical value-of-information measure, in significantly less time.

1 Introduction

Learning users' preferences plays a central role in AI-supported decision making. In this context, Bayesian Preference Elicitation (PE) is an active learning process that, starting from a prior belief over the user's preferences, asks a sequence of questions, while updating the current belief according to the usual Bayesian rules. As application scenario, we consider interactive recommender systems, capable of handling tens or hundreds of available alternatives. PE-based recommenders may be preferable to systems that silently track users' behaviors (e.g., songs heard or web sites visited) whenever users consult the recommender only occasionally, e.g. buying a new smartphone or booking a hotel in a large city.

To learn the users' preferences efficiently, principled elicitation strategies have been designed that, according to the answers already collected, select at each step the *most informative* query. Largely, the goodness of a query has been evaluated by the *Expected Value of Information* (EVOI). Roughly speaking, the EVOI of a query measures the improvement, in term of expected utility of the preferred alternative, that answering that query provides. Differently, in [14], the query selection aims at reducing the *uncertainty* about who the user really is, by evaluating each query in terms of the expected entropy resulting from its possible answers. It is an open question which selection methodology (EVOI vs entropy) provides better results in terms of accuracy and performance. In this paper, we tackle this issue by proposing a systematic comparison between EVOI-based and entropy-based selection strategies within a general PE-based recommender framework.

In particular, we are concerned with PE recommender systems suitable for concrete eCommerce scenarios. In these contexts, two aspects are of fundamental importance to meet the user's expectations. First, answering a query should require a low cognitive load [6, 10]. In this respect, we follow [6, 7, 18], where the query language consists in the set of pairwise comparisons of concrete alternatives. Pairwise comparisons are less ambiguous, easier to answer, and less prone to be influenced by hidden user bias with respect to queries that force the user to evaluate an alternative in absolute terms (e.g. "How much you would pay for a smartphone XYZ?"). Moreover, since pairwise comparisons are homogeneous and do not depend on any specific application domain, they allow to develop general PE systems. Secondly, PE-based recommenders should provide an on-line user experience ensuring that the lag to select the next query does not affect the fluency of the interaction. This puts serious constraints on the performances of the query-selection phase, especially in the case of pairwise comparisons, where the search space grows quadratically with respect to the number of alternatives.

As in [10, 18], we assume that alternatives are characterized by a set of attributes and, in turn, users' preferences are represented by additive value functions over attributes. In order to support arbitrary value functions and beliefs, we use the Markov Chain Monte Carlo as sampling methodology to estimate the expected values. Moreover, we employ best-arm identification techniques [1, 9, 4] to improve the efficiency of the query selection step.

Possible indecisions generate uncertainty on the answers that a user may give to a given question. In [2], each user is characterized by her utility function, and that uncertainty is modelled by a conditional distribution that, given a specific utility function and a query, returns the probability to obtain a specific answer. We follow a simpler approach. As in [10], we assume that a comparison between two items x and y may result in three possible answers ($x \prec y$, $x \succ y$, and $x \sim y$), where indifference is replied whenever the relative difference between the utility values of x and y is smaller than a prescribed threshold. In this way, modelling this component of the PE reduces from guessing an entire conditional distribution to setting a single real-valued indecision threshold.

The first contribution of this work is to compare the EVOI-based and Entropy-based query selection strategies. To this aim, we present a test suite based on two concrete eCommerce scenarios: TripAdvisor and Boston housing. In these settings, we measure both the performance and the accuracy, the latter defined as the best-item misidentification rate. The experimental assessment shows that EVOI is notably more accurate than belief entropy but, on the other hand, it is about 10x slower. This makes EVOI not suitable to on-line applications, even in case of few tens of items. The second contribution is a new entropy measure, we call *entropy of true top*, which exhibits the best qualities of both of the previous approaches: it runs approximately as quickly as the belief entropy, while providing the same accuracy as EVOI.

The paper is structured as follows. In Section 2, we describe PE-based recommender systems and the different query selection strategies in general terms. Section 3 provides a more specific model, including all the assumptions mentioned above. Moreover, we describe concrete implementations of the query selection strategies and how best-arm identification is used to optimize the sampling process. Section 4 shows the experimental results on the TripAdvisor and Boston housing test suites. Final remarks end the paper.

[1] University of Naples Federico II, Italy, email: m.faella@unina.it
[2] University of Naples Federico II, Italy, email: alberto.finzi@unina.it
[3] University of Naples Federico II, Italy, email: luigi.sauro@unina.it

2 Bayesian Preference Elicitation

In this section, we introduce our Bayesian preference elicitation framework, describing the general model, the elicitation process, and different query selection strategies proposed in the literature.

2.1 The General Model

As in [17] we consider a set $\mathcal{I}$ of possible items and we assume that users are characterized by a utility function over the items $u : \mathcal{I} \to \mathbb{R}$. Notice that the set of items $\mathcal{I}$ represents all the possible features a user takes into account when choosing among different options. However, not all elements of $\mathcal{I}$ may be actually available to the user. Some possible combinations of features may just be hypothetical and represent, for instance, options that were available in the past, or that are not on the market yet. Therefore, we distinguish a subset $\mathcal{CI} \subseteq \mathcal{I}$ of *concrete items*.

Let $\mathcal{U}$ be the set of possible utilities (a.k.a. user types), $\mathcal{P}(u)$ is a probability distribution Pr over $\mathcal{U}$ representing, at a given stage of the eliciting process, the *belief* about the type of a user. Initially, we assume a prior distribution $\hat{\mathcal{P}}(u)$ representing the background knowledge of the system before interacting with the user.

We assume that the elicitation process can be stopped at any time, providing a current belief $\mathcal{P}$ about the user. Such belief can be exploited by a recommender to provide an output $\mathsf{res}(\mathcal{P})$. Different kinds of outputs have been considered in the literature, such as an estimate of the true utility $\bar{u}$ of the user or just the ordinal preference on $\mathcal{CI}$ derived by it. In our framework, we focus on mechanisms that return the top concrete item, therefore res is a *recommendation function* from probability distributions over $\mathcal{U}$ to $\mathcal{CI}$.

In particular, in the experimental assessment we use a specific recommendation function, res_v, that selects the concrete item that maximizes the expected utility:

$$\mathsf{res}_v(\mathcal{P}) = \arg\max_{i \in \mathcal{CI}} \mathbb{E}_{\mathcal{P}}[u(i)]\,.$$

An elicitation process interacts with the user by asking *queries*. Given the sets $\mathcal{Q}$ and $\mathcal{R}$ of possible queries and corresponding answers, a *message* $q\!:\!r$ corresponds to the fact that a query $q \in \mathcal{Q}$ has been asked to a user who, in turn, replied with the answer $r \in \mathcal{R}$. Then, the update of a belief according to the message $q\!:\!r$ is defined according to Bayes' rule:

$$\Pr(u \mid q\!:\!r) = \frac{\Pr(u) \cdot \Pr(q\!:\!r \mid u)}{\Pr(q\!:\!r)}\,. \tag{1}$$

Here, $\Pr(u \mid q\!:\!r)$ is the updated belief, that will also be denoted by $\mathcal{P}_{q\,:\,r}(u)$. $\Pr(u)$ is the current belief, also denoted by $\mathcal{P}(u)$. $\Pr(q : r \mid u)$ is the probability of obtaining the answer r to the query q from the user u, and $\Pr(q : r)$ is the probability of receiving r in reply the query q, regardless of any specific user. The value of $\Pr(q : r \mid u)$ depends on how the framework models indecisions or evaluation errors of the user. For the moment, following [2], we represent it as a generic distribution $R_{q,u}(r)$ over the set of answers $\mathcal{R}$. Given this notation, $\Pr(q\!:\!r) = \mathbb{E}_{\mathcal{P}}[R_{q,u}(r)]$, and Equation (1) can be rewritten as follows:

$$\mathcal{P}_{q\,:\,r}(u) = \frac{\mathcal{P}(u) \cdot R_{q,u}(r)}{\mathbb{E}_{\mathcal{P}}[R_{q,u}(r)]}\,. \tag{2}$$

Finally, at the abstract level, the problem of myopically selecting the most appropriate query $\hat{q}$ given a belief $\mathcal{P}$ can be formalized as follows:

$$\hat{q} = \arg\max_{q \in \mathcal{Q}} EBI(\mathcal{P}, q), \tag{3}$$

where $EBI(\mathcal{P}, q)$ is the *expected belief improvement* of the query q in the context $\mathcal{P}$. In particular, assuming that $\mathcal{R}$ is a finite set, we have that

$$\begin{aligned} EBI(\mathcal{P}, q) &= \sum_{r \in \mathcal{R}} \Pr(q\!:\!r) \cdot BI(\mathcal{P}, q\!:\!r) && (4)\\ &= \sum_{r \in \mathcal{R}} \mathbb{E}_{\mathcal{P}}[R_{q,u}(r)] \cdot BI(\mathcal{P}, q\!:\!r), && (5) \end{aligned}$$

where $BI(\mathcal{P}, q\!:\!r)$ is the *belief improvement* of the message $q\!:\!r$ in the context $\mathcal{P}$. Different choices for the BI function are discussed in the next section.

2.2 Belief Improvement Functions

Most PE approaches that deal with the optimal query selection problem adopt one of two forms: value-based or uncertainty-based. The value-based improvement for a potential message $q : r$ (a.k.a. the *value of information* or *VOI*) measures how much the recommendation after $q\!:\!r$ is valued higher than the current recommendation [10]. This leads to the first realization of belief improvement, i.e., the expected value of the recommendation:

$$BI_{val}(\mathcal{P}, q\!:\!r) = \mathbb{E}_{\mathcal{P}_{q\,:\,r}}[u(\hat{\imath})],$$

where $\mathcal{P}_{q\,:\,r}$ is the belief updated with the observation $q\!:\!r$, see equation (2), and $\hat{\imath} = \mathsf{res}_v(\mathcal{P}_{q\,:\,r})$ is the recommendation based on the updated belief.

Alternatively, one may attempt to minimize the uncertainty pertaining to the user and the associated preferences, employing the standard measure of entropy. Depending on the ultimate objective, one may measure the uncertainty of a different random variable. If the objective is to fully characterize the preference of the user, one may measure the entropy of the user type. This leads to a second version of belief improvement:

$$BI_{hu}(\mathcal{P}, q\!:\!r) = -H_{\mathcal{P}_{q\,:\,r}}[u],$$

where $H[X]$ is the entropy of the random variable X.

Finally, if the objective is to identify the top item for the user, one may focus on the entropy of the top item, leading to the third and final realization of belief improvement considered in this paper:

$$BI_{hi}(\mathcal{P}, q\!:\!r) = -H_{\mathcal{P}_{q\,:\,r}}[i_u],$$

where i_u is the true top item for user u (that is, $\arg\max_i u(i)$).

The previous literature is sharply divided between the two families of improvement measures. Chajewska et al. [5] adopt the VOI approach with no mention of the alternatives. Guo and Sanner [10] compare different flavors of VOI with a couple of simpler heuristics, one of which involves identifying the item with the largest uncertainty. Jameson et al. [12] propose to minimize the uncertainty about relevant target variables (such as the importance weights), possibly using variance. They argue against the VOI approach, as in their setting (a system sustaining complex user interactions) they deem it unfeasible to univocally anticipate the ultimate consequences of eliciting a given piece of information. Lepird et al. [14] base their selection on an estimate of the entropy of the belief (similarly to our BI_{hu}). They also argue against the VOI, judging it intractable for their purposes (engineering design optimization). To the best of our knowledge, this paper offers the first systematic comparison between value-based and uncertainty-based improvement measures.

3 The Concrete Model and its Implementation

The model presented in the previous section provides an abstract framework and general definitions which are common to a large class of preference elicitation mechanisms. In this section, we introduce and discuss a concrete realization of these concepts and processes.

3.1 The Concrete Model

A concrete preference elicitation framework is characterized by several specific settings: (i) how items are represented, (ii) which class of value functions is considered, (iii) the type of queries and stopping rule,[4] (iv) which functions res and BI are used. In this section, we fix these settings and delineate the class of elicitation processes we focus on.

First, the set of items $\mathcal{I}$ consists of the product $D_1 \times \cdots \times D_n$, where each D_i represents the set of possible levels for attribute i. Therefore, an item $\mathbf{a} \in \mathcal{I}$ is denoted by a tuple of attribute levels $\mathbf{a} = (a_1, \ldots, a_n)$. Moreover, we assume that attribute values are additive-independent and normalized, that is, the utility function $u : \mathcal{I} \to \mathbb{R}$ is a convex combination of single-attribute value functions:

$$u(\mathbf{a}) = \lambda_1 v_1(a_1) + \cdots + \lambda_n v_n(a_n)$$

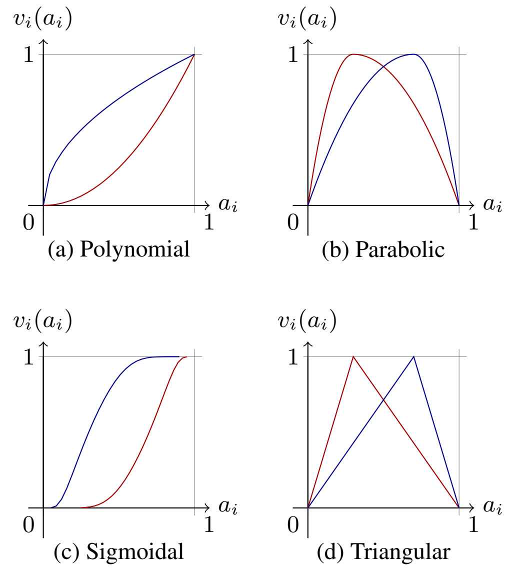
where each v_i is taken from a set $\mathcal{V}_i$ of value functions from D_i to $[0, 1]$ and the weighing constants form a standard simplex, i.e., $\lambda_i \geq 0$ and $\sum_{i=1}^{n} \lambda_i = 1$. We assume that each $\mathcal{V}_i$ is parametrized by a set of p_i real values. Consequently, the set $\mathcal{U}$ of possible user types can be represented by the product $P_{\mathcal{V}_1} \times \cdots \times P_{\mathcal{V}_n} \times [0, 1]^{n-1}$, where each vector in $P_{\mathcal{V}_i} \subseteq \mathbb{R}^{\mathsf{p}_i}$ fixes the parameters that identify a specific function in $\mathcal{V}_i$ and $[0, 1]^{n-1}$ represents the values of the weights $\lambda_1, \ldots, \lambda_{n-1}$ (λ_n is linearly dependent on the others).

We distinguish the following two types of attributes:

- *Ordered/monotonic.* The domain of such an attribute is equipped with a public linear order, and all users prefer values that are higher (or lower) on this order. However, the degree of preference varies among users. For example, other features being equal, all users prefer their smartphone to have more memory and a lower price.
- *Metric/bitonic.* The domain of such an attribute is equipped with a metric, and each user prefers values that are as close as possible to their private target value. For example, each user prefers their smartphone to have a certain screen size.

For each type, we propose a small selection of value models, whose shapes are depicted in Figure 1:

- *Ordered.* We conventionally use the domain $D_i = [0, 1]$, and we consider the *polynomial*, *piece-wise linear*, and *sigmoidal* models defined as follows:
  1. *Polynomial.* Let $\pi_i \in (0, +\infty)$ be a parameter, we set $v_i(a_i) = a_i^{\pi_i}$. The special case of linear value model is given by $\pi_i = 1$. The polynomial family contains both concave and convex monotonic functions. Concave curves (that is, $\pi_i < 1$) correspond to what economists call *diminishing returns*. For example, for a common user a camera having twice as many megapixels as another is generally less than twice as desirable. On the contrary, a professional photographer may value it more than twice as desirable, leading to a convex value shape (that is, $\pi_i > 1$).
  2. *Piece-wise linear.* We set
  $$v_i(a_i) = \begin{cases} \frac{\pi_i^y}{\pi_i^x} a_i & \text{if } a_i \leq \pi_i^x \\ \pi_i^y + \frac{1-\pi_i^y}{1-\pi_i^x}(a_i - \pi_i^x) & \text{otherwise} \end{cases}$$
  for parameters $\pi_i^x, \pi_i^y \in (0, 1)$, with linear value given by $\pi_i^x = \pi_i^y$.
  This family of value functions is a versatile way to obtain both concave and convex shapes, thanks to its two parameters. However, it is non-smooth at $a_i = \pi_i^x$.
  3. *Sigmoidal.* Let $\pi_i \in [0, 1]$ be a parameter. We set
  $$v_i(a_i) = \begin{cases} 1 & \text{if } a_i = 1 \text{ or } \pi_i = 0 \\ 0 & \text{if } \pi_i = 1 \\ \frac{1}{1+\exp^{-f(a_i)}} & \text{otherwise .} \end{cases}$$
  where
  $$f(a_i) = \frac{\pi_i \cdot a_i}{(1-a_i) \cdot (1-\pi_i)} - \frac{(1-a_i) \cdot (1-\pi_i)}{\pi_i \cdot a_i}.$$
  The sigmoidal family is the classic *smooth threshold* function, suitable for those attributes where each user aims at a (possibly different) minimum level, and is approximately indifferent between values that lie on the same side of the threshold, and far away from it.



(a) Polynomial (b) Parabolic (c) Sigmoidal (d) Triangular

Figure 1. Shapes of some 1-parameter value functions. Each function is shown twice, for two different values of its parameter.

- *Metric.* We again use the standard domain $D_i = [0, 1]$, and consider the *triangular* and *parabolic* models defined as follows:
  1. *Triangular.* We set
  $$v_i(a_i) = \begin{cases} \frac{a_i}{\pi_i} & \text{if } a_i \leq \pi_i \\ 1 - \frac{a_i - \pi_i}{1-\pi_i} & \text{otherwise .} \end{cases}$$
  Where $\pi_i \in (0, 1)$ is the target value for this user. This is the simplest bitonic shape and is non-smooth at $a_i = \pi_i$.

[4] The stopping rule may specify, for instance, a fixed number of queries or a special answer of the user which triggers the final response.

2. *Parabolic.* We set

$$v_i(a_i) = \begin{cases} -\frac{1}{\pi_i^2}a_i^2 + \frac{2}{\pi_i}a_i & \text{if } a_i \leq \pi_i \\ \frac{1}{(\pi-1)^2}\left(-a_i^2 + 2\pi_i a_i + 1 - 2\pi_i\right) & \text{otherwise}\,. \end{cases}$$

Where, as before, $\pi_i \in (0, 1)$ is the target value for this user. [5] Connecting two quadratics in this way leads to a smooth bitonic function.

3.2 Queries and Answers

The query language we consider consists of pairwise comparisons between concrete items. We also assume a fixed number of queries and no repetitions. These are the only policy rules adopted.

Pairwise or multiple comparisons have been employed in several works [6, 7, 16, 17, 18] since they involve less cognitive effort [10]. Moreover, with appropriate policy rules, more complex interactions like *order the following items according to your preferences* can be reduced to sequences of pairwise comparisons. We denote by $x \preceq y$ a pairwise comparison, which informally means *do you prefer item* y *to* x*?* We adopt a simple model with three answers: YES means that $\bar{u}(y) > \bar{u}(x) \cdot (1 + \delta)$, NO means that $\bar{u}(x) > \bar{u}(y) \cdot (1 + \delta)$, and MAYBE means that the previous inequalities do not hold, where $\bar{u}$ is the utility function of the user and $\delta \geq 0$ is a parameter called *indecision factor*. The intuition behind this answer model is that, even in the short term, utilities may oscillate due to possible lack of knowledge or other psychological aspects; when two items are too close, this sort of trembling effect generates a sense of indecision which is reflected in our model by the answer MAYBE. Then, the larger is δ the more the user's utility is trembling. For simplicity, we assume that δ is constant across all users and we fix it to 0.005 in the experiments described in Section 4.

Notice that in our model each user deterministically provides a single reply $r_{q,u}$ to a query q. Consequently, $R_{q,u}(r)$ is equal to 1 for the reply $r_{q,u}$ and zero for all the other replies. This simplifies all EBI calculations as follows:

$$EBI_{val}(\mathcal{P}, q) = \int_{u \in \mathcal{U}} u(\hat{\imath}_{r_{q,u}}) \cdot \mathcal{P}(u) du \tag{6}$$

$$EBI_{hu}(\mathcal{P}, q) = \int_{u \in \mathcal{U}} \log\left(\frac{\mathcal{P}(u)}{\mathbb{E}_{u' \sim \mathcal{P}}[R_{q,u'}(r_{q,u})]}\right) \cdot \mathcal{P}(u) du \tag{7}$$

$$EBI_{hi}(\mathcal{P}, q) = \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{CI}} Z_{q:r,j} \cdot \log\left(\frac{Z_{q:r,j}}{\mathbb{E}_{u \sim \mathcal{P}}[R_{q,u}(r)]}\right), \tag{8}$$

$$\text{with } Z_{q:r,j} = \int_{u \in \mathcal{U}} \mathbb{I}(i_u = j)\mathbb{I}(r_{q,u} = r)\mathcal{P}(u) du = \Pr(i_u = j \wedge r_{q,u} = r)\,,$$

where $\mathbb{I}(\alpha)$ is the characteristic function which is equal to 1 if the boolean condition α is true, 0 otherwise.

3.3 Estimating the EBI

Since we strive to support arbitrary value functions, we have to resort to numerical estimation to compute any of the EBIs. In our implementation, we employ the Metropolis Monte Carlo Markov chain (MCMC) method, which is known to perform well on high-dimensional problems [3].[6]

Moreover, recall that the query selection step (Equation (3)) involves estimating a finite number of expected values ($EBI(\mathcal{P}, q)$) and determining the maximum. We notice that this is akin to the best-arm identification (BAI) problem for stochastic multi-armed bandits (MAB) and we exploit this connection to improve the efficiency of the estimation.

In detail, an elementary implementation of Equation (3) consists in numerically evaluating the terms $EBI(\mathcal{P}, q)$ using the same number of samples for each query q. The literature on MABs has produced a variety of refined tecniques for this problem, with the aim of distributing more samples to the most promising arms (i.e., queries). Such techniques come in two flavors: fixed budget [1] or fixed confidence [8, 13]. The first variety takes as one of the inputs the total number of samples, to be distributed among the various arms. The second variety takes as one of the inputs the desired confidence in identifying the maximum-reward arm.[7] Being interested in real-time applications, we want to impose a hard time constraint on the estimation process. Therefore, we choose the fixed-budget approach and we adopt the recent algorithm *variance-based rejects* (VBR) [9], which was shown to perform at least as well as any competitor in a variety of scenarios.

To the best of our knowledge, the connection between optimal query selection and the best-arm identification problem has not been exploited in the literature. For example, Lepird et al. [14] perform optimal query selection using MCMC estimation, but do not mention any way to distribute samples among different queries.

The same connection applies to the step in which the system identifies the recommended item $\hat{\imath}$ as the item maximizing the expected utility. Also in this step we suggest to employ a BAI algorithm such as VBR.

More generally, we find that the three EBIs that we focus on are based on two estimation steps:

- Estimating the maximum expected value among a finite set of n random variables defined on the same sample space. We denote this step by $BAI(n)$.
- Estimating the distribution of a discrete random variable with n possible values. We denote this step by $Distr(n)$.

Table 1 summarizes the number of calls to each of the above procedures for the three versions of EBI. Given the current belief $\mathcal{P}$, all three versions involve an outmost call to $BAI(|\mathcal{Q}|)$ on the sample space $\mathcal{P}$, because they all have to determine the query with the highest expected belief improvement. Additionally, EBI_{val} invokes, for each query q and for each reply r, $BAI(|\mathcal{CI}|)$ on the sample space $\mathcal{P}_{q\,:\,r}$ in order to determine the recommendation $\hat{\imath}_{r_{q,u}}$ (see Equation 6).

EBI_{hu} and EBI_{hi}, instead, need to estimate, for a given query, the distribution of the random variable representing the reply to that query (see the denumerators in Equations 7 and 8). That justifies the terms $|\mathcal{Q}| \cdot Distr(|\mathcal{R}|)$ in Table 1. EBI_{hi} also needs to estimate for each query q the distribution of the variable $Z_{q:r,j}$ that jointly determines the reply and the top item associated to a given query. Notice that all these random variables are estimated on the current belief $\mathcal{P}$. This means that we can use a single sampling of $\mathcal{P}$ for all queries. As

[5] Note that parabolic value functions are obtained by assuming that

$$v_i(a_i) = \begin{cases} P_1(a_i) & \text{if } a_i \leq \pi_i \\ P_2(a_i) & \text{otherwise} \end{cases}$$

where $P_1(a_i)$ and $P_2(a_i)$ are quadratic functions satisfying the conditions $P_1(0) = 0$, $P_1(\pi_i) = 1$, $P_1'(\pi_i) = 0$, $P_2(\pi_i) = 1$, $P_2'(\pi_i) = 0$, and $P_2(1) = 0$.

[6] More advanced MCMC methods may be used. For example, knowledge of the gradient of the density would allow us to use Hamiltonian Monte Carlo techniques.

[7] Of course, the algorithm can only satisfy this requirement probabilistically.

shown in the next section, this notably improves performances. Conversely, sampling reuse cannot be adopted in EBI_{val}, because computing the recommendation $\hat{\imath}_{r_{q,u}}$ for each potential message $q : r$ requires a BAI on a different distribution $\mathcal{P}_{q:r}$.

Version	Main estimation steps
EBI_{val}	$BAI(\|\mathcal{Q}\|, \mathcal{P}) + \|\mathcal{Q}\| \cdot \|\mathcal{R}\| \cdot BAI(\|\mathcal{CI}\|)$
EBI_{hu}	$BAI(\|\mathcal{Q}\|) + \|\mathcal{Q}\| \cdot Distr(\|\mathcal{R}\|)$
EBI_{hi}	$BAI(\|\mathcal{Q}\|) + \|\mathcal{Q}\| \cdot Distr(\|\mathcal{R}\|) + \|\mathcal{Q}\| \cdot Distr(\|\mathcal{R}\| \cdot \|\mathcal{CI}\|)$

Table 1. Number and type of estimation steps for each version of EBI.

4 Experiments

In the following experiments, we had to identify a way to put on equal footing three very different belief improvement notions. As explained earlier, the three realizations of the belief improvement function are all based on two numerical estimation procedures, denoted by BAI and $Distr$. To compare the performance of the three approaches, in each experiment we fixed an integral parameter s, representing the basic number of samples to be used for estimation. Then, each call to $BAI(n)$ (resp., $Distr(n)$) is assigned a total budget of $s \cdot n$ samples.

Our prototype implementation, nicknamed Quest, is written in Java and its source code is publicly available.[8] Basic probability densities are provided by the Apache Commons Math library. All experiments were performed on an 8-core 16-thread AMD Ryzen 2700X, clocked at 3.7 Ghz and equipped with 16 GB of RAM. Currently, our tool is entirely sequential, so its running time is unlikely to be affected by the system load.

4.1 Boston Housing

The first experimental setting simulates the problem of choosing a neighborhood when buying a house. The Boston housing dataset comprises information about 506 areas of the city of Boston [11]. Of the 14 attributes in the original dataset, we used the following four:

1. CRIM: per capita crime rate
2. NOX: nitric oxides concentration
3. PTRATIO: pupil-teacher ratio
4. MEDV: median value of owner-occupied homes

The reason for restricting the set of attributes is twofold: first, for some attributes[9] we found it impractical to pick a reasonable value model without further information; second, we needed to bound the execution time both to match our real-time motivation and to perform the large variety of simulations required to compare different parameter combinations.

For the first three attributes in the above list, one can safely assume that any user would prefer lower levels. We attach the simplest monotonically decreasing value model to the CRIM and PTRATIO attributes: the inverse linear model. For the pollution indicator NOX, we assume an inverse polynomial value model (Figure 1(a)). Finally, for the last attribute MEDV, we assume that each user has a target level[10] and therefore we adopt the triangular value model for it (Figure 1(d)). This choice of value model leads to a six-dimensional problem (four weights for the attributes and two shape parameters for the attributes NOX and MEDV).

Set-up. We run a simulated elicitation process on 1000 users, where each user is asked 5 queries before being given a recommendation. Two different settings have been considered for the prior belief and the sampling distribution for simulated users. In the first setting, called for brevity S1, they are both uniform. In the second setting, called S2, the prior belief is uniform, whereas users are generated according to a cylindrical Gaussian distribution with variance 0.3 and mean value 0.5 in all six dimensions. The intent behind S2 is to experiment with a sample distribution that is simple but different from the prior assumed by the system.

The basic number of samples is fixed to $s = 5$. Such number was chosen heuristically, as a compromise between execution time and accuracy of the estimates.

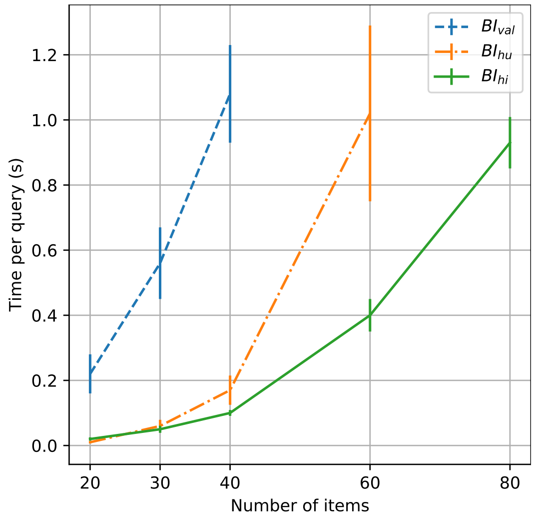


Figure 2. Time to compute the next optimal query. Selected rows from the Boston housing dataset. Mean and standard deviation over 1000 simulated users. Each BI function is included only when it can compute the next optimal query within approximately 1 second.

Execution time. In terms of execution time, settings S1 and S2 behave very similarly. Figure 2 plots the average time needed to identify the next best query for each realization of BI, when the number of items varies between 20 and 80, in setting S1. The performance of each BI function is plotted until it significantly exceeds the threshold of 1 second. The experiment confirms the intrinsic quadratic complexity, while revealing a significant difference between the time needed by the three versions: BI_{val} is the most expensive, followed by BI_{hu} and then BI_{hi}. This implies that an elicitation based on BI_{val} could handle about 40 items within the conventional limit of 1 second per query. On the other hand, in our setup BI_{hi} can handle up to 80 items within in the same time, while achieving a quality of recommendation that is similar to the one of BI_{val}, as shown in the following section.

[8] See https://bitbucket.org/mfaella/quest.

[9] Like attribute "B", defined as "$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town".

[10] Most likely the highest median home value that they can afford.

The performance difference is easily explained: BI_{val} requires the computation of the recommendation $\hat{\imath}$ resulting from the updated belief $\mathcal{P}'$, given a candidate query and a potential reply. In practice, this involves running a best-arm identification algorithm for all pairs of queries and replies. The other improvement measures avoid these calculations.

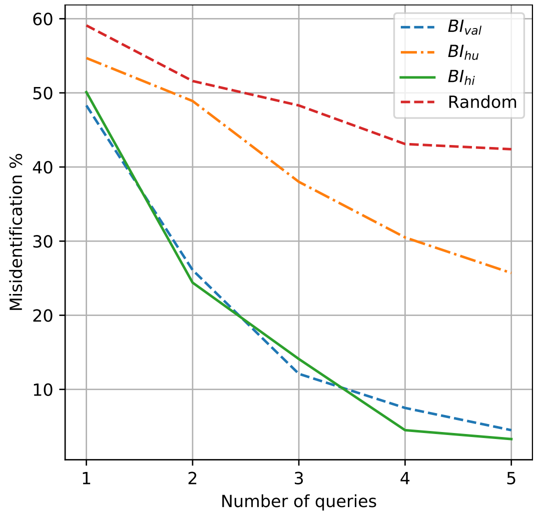


Figure 3. Quality of the recommendation for different improvement measures. Selected rows from the Boston housing dataset (30 items), presented to 1000 simulated users, with uniform prior and generator (setting S1).

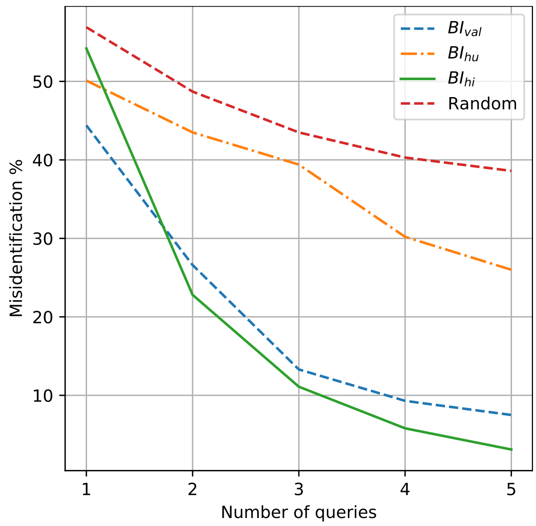


Figure 4. Quality of the recommendation for different improvement measures. Selected rows from the Boston housing dataset (30 items), presented to 1000 simulated users, with uniform prior and Gaussian generator (setting S2).

Quality of the recommendation. Figures 3 and 4 report the percentage of misidentification[11] achieved in the settings S1 and S2, respectively, by the elicitation process on 30 items after 1 to 5 queries, when the latter are chosen according to the three belief improvement measures. As a baseline, we also report the behavior obtained by choosing the next query uniformly at random.

The data shows that BI_{val} and BI_{hi} perform significantly better than BI_{hu}, which in turn performs somewhat better than the random strategy. The poor performance of BI_{hu} is explained by the fact that, contrary to the other two measures, it focuses on identifying the user type, rather than identifying the user's top item. As a consequence, it may favor queries that discriminate between different classes of users, even if those classes exhibit the same top item.

Focusing on BI_{val} and BI_{hi}, in the S1 regime, there appears to be a striking overlap in the performance, whereas BI_{hi} clearly outperforms BI_{val} in S2 (after 5 queries the misidentification rate is less than half).

4.2 TripAdvisor

A second experimental setting relies on the TripAdvisor dataset [15], which contains 235 793 hotel reviews, each featuring the following fields: *Hotel ID*, *User ID*, *Price*, *Location*, *Overall Rating*, *Value Rating*, *Rooms Rating*, *Location Rating*, *Cleanliness Rating*, *Front Desk Rating*, *Service Rating*, *Business Service Rating*. All the review-level ratings are on a discrete ordinal scale from 1 to 5 and represent the assessment provided by the user with *User ID* for the hotel with *Hotel ID*. To cast the data into our model, we start by assuming that each hotel is characterized by the following intrinsic attributes, corresponding to five aspect-specific fields in the dataset:

1. VALUE: value for price;
2. ROOMS: quality of the rooms;
3. LOCATION: quality of the hotel location;
4. CLEAN: cleanliness of the hotel;
5. FRONT: quality of the front desk.

For each of these attributes, we assume the linear value model, leading to a five-dimensional problem. Then, we interpret the *Overall Rating* field as a utility value. At this point, we can use the data to estimate a model of the prior belief, as illustrated below.

Items and Prior model. Given the dataset, we extract the set of concrete items $\mathcal{CI}$ as the list of hotels associated with their mean evaluations (one for each attribute). Specifically, each hotel gives rise to a concrete item with attribute levels $(r_1^m, r_2^m, r_3^m, r_4^m, r_5^m)$, where r_j^m is the mean value of the j-th aspect-specific rating for that hotel.

The prior belief on user types is computed in two phases. First, we extract a set of user types from the dataset by applying multiple linear regression. Specifically, for each *User ID* with more than four reviews, we estimate the 5-tuple of coefficients $(\lambda_1, \ldots, \lambda_5)$ as the best fit of $\sum_{j=1}^{5} \lambda_j \cdot r_j = r_{\text{all}}$, where r_j are the selected ratings (as enumerated above) and r_{all} is the associated *Overall Rating* provided by the user. In this setting, coefficient estimation can than be obtained by deploying a multilinear regression with positive coefficients and no intercept (we employed lasso with $\alpha = 0.0001$). Secondly, given the set of extracted user types $\{(\lambda_{i,1}, \ldots, \lambda_{i,5})\}$, one for each user i, the prior belief is obtained as an estimation of the multivariate Gaussian mixture that best fits these data. In particular, the best fit was provided by a single five-dimensional Gaussian distribution.

[11] Here, by misidentification we mean that the recommended item has a value for the user that is more than 1% lower than the value of the top item for that user.

Set-up. We run a simulated elicitation process on 1000 users, where each user is asked 4 queries before being given a recommendation. The prior belief is the estimated multivariate introduced above. The same multivariate is here exploited as the sampling distribution for the simulated users. The basic number of samples is fixed to $s = 10$.

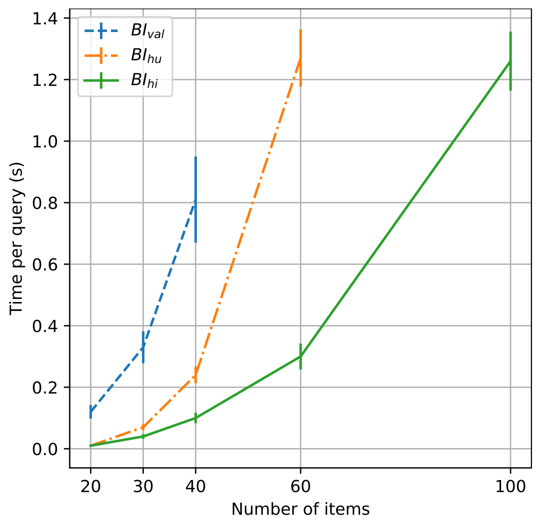


Figure 5. Time to compute the next optimal query. Selected rows from the TripAdvisor dataset. Mean and standard deviation over 1000 simulated users.

Execution time. Figure 5 illustrates the average time needed to identify the next best query for each realization of BI, when the number of items varies between 20 and 100. The experimental results confirm the computation effectiveness of BI_{hi} with respect to the other strategies in a realistic case study. Moreover, similarly to the Boston housing setting, this significant reduction in time performance seems to preserve the quality of the recommendation, as discussed below.

Quality of the recommendation. Figure 6 reports the percentage of misidentification achieved by the elicitation process on 40 items after 1 to 4 queries. Also in this case, we compare the results with respect to a random choice baseline. The collected results seem to confirm the performance observed in the previous case study: BI_{val} and BI_{hi} show a similar improvement of the quality of recommendation (with respect to the number of queries) and dominate the other two strategies.

5 Conclusions

We presented a general Bayesian preference elicitation framework supporting arbitrary value functions on a multi-attribute domain of alternatives. We focused on PE methods aiming at designing effective recommender systems suitable for on-line interactive applications.

We illustrated the overall framework, describing the general model along with the associated elicitation process. In this setting, we introduced and discussed different myopic measures for optimal query selection, proposing a systematic comparison between value-based and uncertainty-based measures. In order to provide an empirical

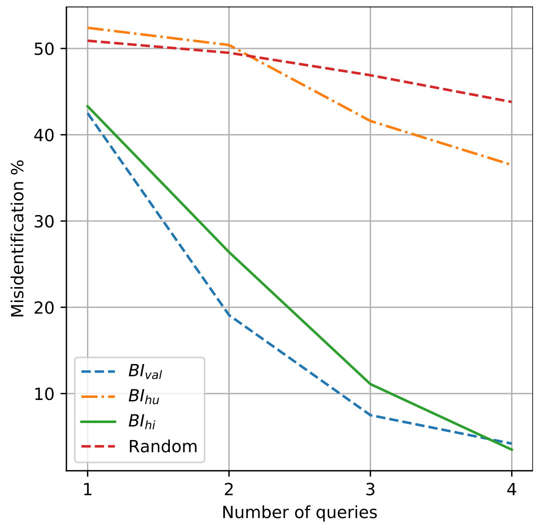


Figure 6. Quality of the recommendation for different improvement measures. Selected rows from the TripAdvisor dataset (40 items), presented to 1000 simulated users.

evaluation of these methods, we introduced a concrete model and concrete implementations of the query selection strategies. In this setting, query selection with arbitrary utility functions and beliefs is enabled by MCMC sampling combined with best-arm identification methods. MCMC sampling has already been used in [14] whereas, to the best of our knowledge, this is the first work in Bayesian PE exploiting best-arm identification to improve the efficiency of the query selection.

In order to compare the proposed measures for query selection, we introduced a test suite based on two concrete eCommerce scenarios: TripAdvisor and Boston housing. In these settings, we compared EVOI-based and entropy-based methods, assessing their performance and accuracy. The collected results suggest that the EVOI-based approaches are more accurate than the ones based on belief entropy, but computationally more expensive. On the other hand, a more focused entropy measure, *entropy of the true top*, seems to combine the advantages of the previous approaches by performing approximatively as fast as belief entropy while providing the same accuracy of EVOI.

Future works encompass several possible directions. First, in concrete scenarios recommender systems generally do not provide just a single option, but rather a manageable shortlist of the most promising items. Consequently, we will generalize the recommendation function res so that it returns the top-k items. Clearly, also the belief improvement functions have to be extended accordingly. Secondly, we will include in our framework recommendation functions that try to minimize the expected regret. Regret-based recommendations can be more accurate that res_v in case users show different levels of satisfaction w.r.t. the whole collection of concrete items. However, measuring the regret in absolute terms as in [2] tends to overweight enthusiastic users. For this reason, we intend to use the relative regret in our recommendation functions. Finally, performances are heavily affected by the fact that pairwise comparisons grow quadratically in the number of concrete items. Thus, we will investigate the costs and benefits of restricting, as in [10, 16], the query selection process to a number of queries that is linear in the number of items.

REFERENCES

[1] Jean-Yves Audibert and Sébastien Bubeck, 'Best arm identification in multi-armed bandits', in *COLT 2010 - 23th Conference on Learning Theory*, pp. 41–53, (2010).

[2] Craig Boutilier, 'A pomdp formulation of preference elicitation problems', in *AAAI*, (2002).

[3] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[4] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz, 'Pure exploration in multi-armed bandits problems', in *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, ALT'09, pp. 23–37, Berlin, Heidelberg, (2009). Springer-Verlag.

[5] Urszula Chajewska, Daphne Koller, and Ronald Parr, 'Making rational decisions using adaptive utility elicitation', in *Proceedings of 17th National Conference on Artificial Intelligence (AAAI)*, pp. 363–369. AAAI Press, (2000).

[6] Vincent Conitzer, 'Eliciting single-peaked preferences using comparison queries', *J. Artif. Intell. Res.*, **35**, 161–191, (2009).

[7] Palash Dey and Neeldhara Misra, 'Elicitation for preferences single peaked on trees', in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9-15 July*, pp. 215–221, (2016).

[8] Eyal Even-Dar, Shie Mannor, and Yishay Mansour, 'Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems', *J. Mach. Learn. Res.*, **7**, 1079–1105, (2006).

[9] Marco Faella, Alberto Finzi, and Luigi Sauro, 'Rapidly finding the best arm using variance', in *Proc. of ECAI, 24th European Conference of Artificial Intelligence*, (2020).

[10] Shengbo Guo and Scott Sanner, 'Real-time multiattribute bayesian preference elicitation with pairwise comparison queries', *Journal of Machine Learning Research - Proceedings Track*, **9**, 289–296, (12 2010).

[11] D. Harrison and D.L. Rubinfeld, 'Hedonic prices and the demand for clean air', *J. Environ. Economics & Management*, **5**, 81–102, (1978).

[12] Anthony Jameson, Ralph Schäfer, Joep Simons, and Thomas Weis, 'Adaptive provision of evaluation-oriented information: Tasks and techniques', in *IJCAI*, (1995).

[13] Kevin G. Jamieson, Matthew Malloy, Robert D. Nowak, and Sbastien Bubeck, 'lil' ucb : An optimal exploration algorithm for multi-armed bandits.', in *COLT*, eds., Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvri, volume 35 of *JMLR Workshop and Conference Proceedings*, pp. 423–439. JMLR.org, (2014).

[14] John Lepird, Michael Owen, and Mykel Kochenderfer, 'Bayesian preference elicitation for multiobjective engineering design optimization', *Journal of Aerospace Information Systems*, **12**, 1–12, (10 2015).

[15] D. Marcheggiani, O. Täckström, A. Esuli, and Sebastiani F., 'Hierarchical multi-label conditional random fields for aspect-oriented opinion mining', in *Proceedings ECIR 2014, Lecture Notes in Computer Science*, volume 8416 LNCS, pp. 273–285. Springer Verlag, (2014).

[16] Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier, 'Gradient-based optimization for bayesian preference elicitation', *CoRR*, **abs/1911.09153**, (2019).

[17] Paolo Viappiani and Craig Boutilier, 'Optimal bayesian recommendation sets and myopically optimal choice query sets', in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, eds., John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, pp. 2352–2360. Curran Associates, Inc., (2010).

[18] Luisa M. Zintgraf, Diederik M. Roijers, Sjoerd Linders, Catholijn M. Jonker, and Ann Nowé, 'Ordered preference elicitation strategies for supporting multi-objective decision making', in *Proc. of the 17th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS), Stockholm, Sweden, July 10-15*, (2018).