Real-time Multiattribute Bayesian Preference Elicitation with Pairwise Comparison Queries

Shengbo Guo
Australian National University - NICTA

Scott Sanner
NICTA - Australian National University

Abstract

Preference elicitation (PE) is an important component of interactive decision support systems that aim to make optimal recommendations to users by actively querying their preferences. In this paper, we outline five principles important for PE in real-world problems: (1) real-time, (2) multiattribute, (3) low cognitive load, (4) robust to noise, and (5) scalable. In light of these requirements, we introduce an approximate PE framework based on TrueSkill for performing efficient closed-form Bayesian updates and query selection for a multiattribute utility belief state — a novel PE approach that naturally facilitates the efficient evaluation of value of information (VOI) heuristics for use in query selection strategies. Our best VOI query strategy satisfies all five principles (in contrast to related work) and performs on par with the most accurate (and often computationally intensive) algorithms on experiments with synthetic and real-world datasets.

1 Introduction

Preference elicitation (PE) is an important component of eCommerce and recommender systems that propose items or services from a potentially large set of available choices but due to practical constraints may only query a limited number of preferences. The PE task consists of (a) querying the user about their preferences and (b) recommending an item that maximizes the user's latent utility. Of course, a PE system is limited by real-world performance constraints that require phase (a) to be efficient while ensuring phase (b) can make an optimal recommendation with high certainty. To this end, we outline five principles important for the practical application of PE in real-world settings used to guide our research in this work:

1. *Real-time*: A PE system that takes more than a few seconds to propose a query or that asks a large number of uninformative queries will likely not be viewed as useful by a user.
2. *Multiattribute*: Exploiting the natural attribute structure of services or items in the form of multiattribute utility functions (Keeney and Raiffa, 1976) is crucial when the number of recommendable items exceeds the number of queries a PE system can reasonably ask. In this case, learning preferences over attribute dimensions can simultaneously inform preferences over many items.
3. *Low cognitive load*: Since the task of utility elicitation is cognitively difficult and error prone (Chajewska et al., 2000), queries that are more difficult for users lead to higher noise and less certainty in the utility elicited. Thus, we focus on pairwise comparison queries known to require low cognitive load for users (Conitzer, 2009).
4. *Robust to noise*: A real-world PE system has to make robust utility predictions in the presence of noisy query responses. Bayesian PE approaches that maintain a belief distribution over utility functions and update beliefs using a realistic query confusion model are one natural way to handle noise, although exact inference in these Bayesian models may often be intractable.
5. *Scalable*: Since many real-world decision problems (real estate, consumer electronics) involve large numbers of items (Chajewska et al., 2000), a scalable PE system should not evaluate more than $O(m)$ queries per PE stage. Furthermore, for scalability, it is crucial for a PE system not only to choose informative queries, but also to choose queries that help it discriminate among the highest utility items actually available in the item set.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

In the following sections, we develop an approximate Bayesian PE framework to satisfy all five of these principles (in contrast to related work) and demonstrate this empirically on synthetic and real-world datasets.

2 Bayesian Preference Elicitation

2.1 User Utility Model

In multiattribute utility theory (MAUT) (Keeney and Raiffa, 1976), utilities are modeled over a D-dimensional *attribute set* $\mathcal{X} = \{X_1, \ldots, X_D\}$ with *attribute choices* $X_d = \{x_{d1}, \ldots, x_{d|X_d|}\}$ (where $|X_d|$ denotes the cardinality of X_d). An *item* is described by its attribute choice assignments $\mathbf{x} = (x_1, \ldots, x_D)$ where $x_d \in X_d$. In our model, an *attribute weight vector* $\mathbf{w} = (w_{11}, \ldots, w_{1|X_1|}, \ldots, w_{D1}, \ldots, w_{D|X_D|})$ describes the utility of *each* attribute choice in *each* attribute dimension.

We assume that the utility $u(\mathbf{x}|\mathbf{w})$ of item $\mathbf{x}$ w.r.t. attribute weight vector $\mathbf{w}$ decomposes *additively* over the attribute choices of $\mathbf{x}$, i.e.,

$$u(\mathbf{x}|\mathbf{w}) = \sum_{d=1}^{D} \mathbf{w}_{d,\#(\mathbf{x},d)}, \quad u^*(\mathbf{x}) = \sum_{d=1}^{D} \mathbf{w}^*_{d,\#(\mathbf{x},d)} \quad (1)$$

where $\#(\mathbf{x}, d)$ returns index in $\{1, \ldots, |X_d|\}$ for attribute choice x_d of $\mathbf{x}$ and u^* represents the user's true utility w.r.t. their true (but hidden) $\mathbf{w}^*$.

Since $\mathbf{w}^*$ is unknown to the decision support system, it is the goal of preference elicitation to learn an estimate $\mathbf{w}$ of $\mathbf{w}^*$ with enough certainty to yield a low expected loss on the item recommended. We take a Bayesian perspective on learning $\mathbf{w}$ (Chajewska and Koller, 2000) and thus maintain a probability distribution $P(\mathbf{w})$ representing our beliefs over $\mathbf{w}^*$.

Because $P(\mathbf{w})$ is a distribution over a multidimensional continuous random variable $\mathbf{w}$, we represent this distribution as a Gaussian with diagonal covariance, represented compactly in a factorized format as follows:

$$P(\mathbf{w}) = \prod_{d=1}^{D} \prod_{i=1}^{|X_d|} p(w_{di}) = \prod_{d=1}^{D} \prod_{i=1}^{|X_d|} \mathcal{N}(w_{di}; \mu_{di}, \sigma^2_{di}). \quad (2)$$

We assume the vectors μ and σ represent the respective mean and standard deviation for the normal distribution over each corresponding attribute choice in $\mathbf{w}$. While the use of a diagonal covariance is a strong modeling assumption, we can exploit its properties for efficient computation; furthermore, the number of parameters to learn (i.e, μ and σ) scales linearly with the size of $\mathbf{w}$ rather than quadratically as would be the case with a full covariance assumption.

2.2 Query & User Response Model

In this paper, we focus on pairwise comparison queries known to require low cognitive load for users (Conitzer, 2009), hence reducing noise in the elicitation process. We use $Q_{ij} = \{i \succ j, i \prec j, i \sim j\}$ to represent a pairwise comparison query indicating the user's preferences of item $\mathbf{x}_i$ vs. item $\mathbf{x}_j$ (henceforth just i and j). Depending on the user's attribute weight vector $\mathbf{w}$ and the corresponding item utilities, $u(i|\mathbf{w})$ and $u(j|\mathbf{w})$, the user's response $q_{ij} \in Q_{ij}$ indicates the following:

- $i \succ j$: the user prefers i to j,
- $i \prec j$: the user prefers j to i,
- $i \sim j$: the user is indifferent between i and j.

If the difference between two item utilities is large, it is easy for the user to answer the query; otherwise, confusion plays a role in deciding the preference. The Bradley-Terry model of confusion (Bradley and Terry, 1952) provides one way to model such noise in the case of *strict* pairwise preference $i \succ j$ or $i \prec j$; while numerous extensions attempt to model the additional indifference choice $i \sim j$ we require, none of these extensions directly lend themselves to closed-form Bayesian updates with a tractable family of belief distributions.

To facilitate efficient approximate Bayesian inference (as shown in the next section), we represent the user query model in (4) with an indicator function[1] over the pairwise utility difference

$$\begin{aligned} P(Q_{ij} = i \succ j|\mathbf{w}) &= \mathrm{I}[u(i|\mathbf{w}) - u(j|\mathbf{w}) > \epsilon] \\ P(Q_{ij} = i \prec j|\mathbf{w}) &= \mathrm{I}[u(j|\mathbf{w}) - u(i|\mathbf{w}) > \epsilon] \\ P(Q_{ij} = i \sim j|\mathbf{w}) &= \mathrm{I}[|u(i|\mathbf{w}) - u(j|\mathbf{w})| \leq \epsilon], \end{aligned} \quad (3)$$

where we can modulate the range of utility differences for which the user is indifferent by adjusting ϵ. Note that by definition, $\sum_{q_{ij}} P(q_{ij}|\mathbf{w}) = 1$.

2.3 PE Graphical Model and Inference

In this paper, we take a Bayesian approach to PE. Thus, given a prior utility belief $P(\mathbf{w}|R^n)$ w.r.t. a (possibly empty) set of $n \geq 0$ query responses $R^n = \{q_{kl}\}$ and a new query response q_{ij}, we perform the following Bayesian update to obtain a posterior belief $P(\mathbf{w}|R^{n+1})$ where $R^{n+1} = R^n \cup \{q_{ij}\}$:

$$\begin{aligned} P(\mathbf{w}|R^{n+1}) &\propto P(q_{ij}|\mathbf{w}, R^n)P(\mathbf{w}|R^n) \\ &\propto P(q_{ij}|\mathbf{w})P(\mathbf{w}|R^n). \end{aligned} \quad (4)$$

[1] We use $\mathrm{I}[\cdot]$ as an indicator function taking the value 1 when its argument is true and 0 otherwise.

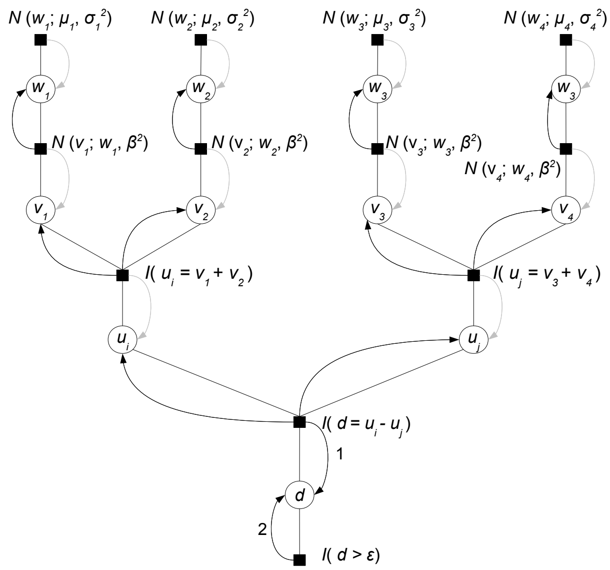

Figure 1: PE factor graph from TrueSkill (Herbrich et al., 2006) for $q_{ij} = i \succ j$. Items i and j have two attribute choices each with respective weights (w_1, w_2) and (w_3, w_4). The posterior over (w_1, w_2, w_3, w_4) can be inferred with the following message passing schedule: (1) messages pass along *gray* arrows from top to bottom, (2) the marginal over d is updated via message 1 followed by message 2 (which required moment matching), (3) messages pass from bottom to top along *black* arrows.

Assuming that our query likelihood $P(q_{ij}|\mathbf{w})$ is modeled as described in (3), we note that the form of the exact posterior is not a diagonal Gaussian as is the initial prior $P(\mathbf{w}|R^0) = P(\mathbf{w}|\emptyset) = P(\mathbf{w})$ defined in (2), rather, it is a mixture of truncated Gaussians where the number of mixture components grows exponentially with the number of queries.

To avoid this exponential exact inference, we must turn to approximate Bayesian inference techniques. First we note that the use of (4) leads to a slight variation on the *TrueSkill*™ (Herbrich et al., 2006) graphical model for multiattribute PE shown in Figure 1. Consequently, our approximate Bayesian inference method for (4) will be to adopt the TrueSkill approach of approximating the posterior in the family of diagonal Gaussians using the technique of moment matching, known to minimize the Kullback-Leibler divergence for Gaussian distributions. By combining moment matching and the sum-product algorithm for factor graphs (Kschischang et al., 2001), we can then obtain the approximate posterior marginal over $\mathbf{w}$ in the approximate Bayesian updating scheme known as *assumed density filtering.* We set $\beta = 1$ and use the TrueSkill model as shown for Bayesian updating, however we note that the variables $\mathbf{v}$ do not play a role in our value of information analysis.

Of key importance in this approximate Bayesian updating scheme is to note that from prior sufficient statistics μ^n and σ^n for $P(\mathbf{w})$ in the form of (2), the update with the $n+1$st query response q_{ij} results in posterior sufficient statistics μ^{n+1} and σ^{n+1}. While not guaranteed in practice due to approximation, ideally we would expect in the limit of queries as $n \to \infty$, our belief distribution will approach full certainty in the user's hidden utility, i.e., $\mu^n \to \mathbf{w}^*$ and $\sigma^n \to \mathbf{0}$.

Update equations for the (cached) marginals and messages for different factor types in Figure 1 have been presented for the "team model" in (Herbrich et al., 2006). One exception is the update equations for factors and variables involving shared attribute choices. Unlike TrueSkill that assumes "team players" are not shared, our preference elicitation queries permit two multiattribute items to share up to $D-1$ common attribute choices, which leads to shared variables and factors in the PE factor graph.

Shared variables and factors induce loops in the TrueSkill factor graph, though we note that the introduction of shared attributes does not make any difference for the variable d as follows. Imagine that item i and j have a shared attribute choice indicated by w_2 and w_3, we thus have that v_2 and v_3 correspond to the same variable. It is easy to see that the deterministic factors encode the following equations (Figure 1):

$$\begin{aligned} u_i &= v_1 + v_2, \\ u_j &= v_3 + v_4, \\ d &= u_i - u_j. \end{aligned}$$

Hence we obtain

$$d = u_i - u_j = (v_1 + v_2) - (v_3 + v_4) = v_1 - v_4, \quad (5)$$

where we note that d does not depend on v_2 or v_3. Clearly, in exact Bayesian inference, this algebraic transformation would not make any difference. Therefore, we can simply omit shared attributes when performing Bayesian updating.

3 Value of Information

Now that we know how to efficiently update our multiattribute utility distribution based on a user's query responses, we are left with the question of how to formulate a query strategy. While all queries should improve the certainty of our utility estimate w.r.t. some items, we are most concerned with finding the optimal item with high certainty.

One way to evaluate different queries is to measure the extent to which they help the PE system reach this optimal decision, which can be formalized using *value* *of information* (VOI) (Howard, 1966). VOI plays an important role in many Bayesian PE strategies, as first proposed in (Chajewska et al., 2000) and our Bayesian PE framework naturally facilitates an approximation of VOI as we show next.

One way to formalize the VOI of a query in our PE framework is to note that the query which maximizes our VOI is the one that most reduces our loss. Unfortunately, we can never know our *true loss* for recommending an item, we can only calculate our *expected loss* — the query leading to the maximum reduction in expected loss will then maximize our expected VOI.

But how do we define the expected loss at any stage of PE? First we note that if we stop PE after eliciting query response set R, then we have posterior utility beliefs $P(\mathbf{w}|R)$ summarized by sufficient statistics (μ^R, σ^R). From this, we can efficiently compute the highest expected utility item i^*_R:[2]

$$\begin{aligned}
i^*_R &= \arg\max_i \; \mathrm{E}_{P(\mathbf{w}|R)}[u(i|\mathbf{w})] \\
&= \arg\max_i \int \cdots \int_{\mathbb{R}^D} \prod_{e=1}^{D} \mathcal{N}(\mathbf{w}; \mu, \mathrm{diag}(\sigma^2)) \sum_{d=1}^{D} w_{d,\#(i,d)} d\mathbf{w} \\
&= \arg\max_i \; \sum_{d=1}^{D} \int_{\mathbb{R}} \mathcal{N}(w_{d,\#(i,d)}; \mu_{d,\#(i,d)}, \sigma^2_{d,\#(i,d)}) \\
&\qquad\qquad \cdot w_{d,\#(i,d)} dw_{d,\#(i,d)} \\
&= \arg\max_i \; u(i|\mu^R). \qquad (6)
\end{aligned}$$

This straightforward result exploits the fact that $P(\mathbf{w}|R)$ is diagonal Gaussian and thus the expectation factorizes along each attribute dimension.

Now let us assume that we have access to the true utilities of items i and k, respectively $u^*(i)$ and $u^*(k)$ recalling (1). If we recommend item i in place of item k, then our loss for doing so is $\max(0, u^*(k) - u^*(i))$, i.e., if $u^*(k) > u^*(i)$ then we lose $u^*(k) - u^*(i)$ by recommending i, otherwise we incur no loss.

Of course, we do not have the true item utilities to compute the actual loss. However, in the Bayesian setting, we do have a belief distribution over the item utilities, which we can use to compute the expected loss. Thus, to compute the expected loss (EL) of recommending the best item i^*_R instead of recommending item k, we would evaluate the following expectation:

$$\mathrm{EL}(k, R) = \mathrm{E}_{P(\mathbf{w}|R)} \left[\max\left(0, u(k|\mathbf{w}) - u(i^*_R|\mathbf{w})\right)\right]. \qquad (7)$$

Unfortunately, the computation of EL is difficult because the expectation integral over the max prevents the calculation from factorizing along attribute dimensions of the Gaussian utility beliefs. For this reason, we opt for a computationally simpler approximation of the expected loss ($\widehat{\mathrm{EL}}$) where we use the expected utility $u(i^*_R|\mu^R)$ of i^*_R from (6) as a surrogate for its true utility, leading to the *closed-form* calculation:

$$\begin{aligned}
\widehat{\mathrm{EL}}(k, R) &= \mathrm{E}_{P(\mathbf{w}|R)} \left[\max(0, u(k|\mathbf{w}) - u(i^*_R|\mu^R))\right] \\
&= \int_{\mathbb{R}} \left(\max(0, u(k|\mathbf{w}) - u(i^*_R|\mu^R))\right) P(\mathbf{w}|R) d\mathbf{w} \\
&= (\mu_{i^*_R} - \mu_k)(1 - \Phi_{\mu_k, \sigma^2_k}(\mu_{i^*_R})) \\
&\quad - \frac{\sigma_k}{\sqrt{2\pi}} \exp\left(-\frac{(\mu_{i^*_R} - \mu_k)^2}{2\sigma^2_k}\right). \qquad (8)
\end{aligned}$$

Here, Φ_{μ_k,σ^2_k} is the normal CDF, $\mu_k = \sum_d \mu_{d,\#(k,d)}$, $\sigma^2_k = \sum_d \sigma^2_{d,\#(k,d)}$, and $\mu_{i^*_R} = \sum_d \mu_{d,\#(i^*_R,d)}$. (Space limitations require omission of the derivation.)

From this single item expected loss, we can then determine the maximum expected loss (MEL) we might incur by recommending i^*_R instead of *some* other k:

$$\mathrm{MEL}(R) = \max_k \widehat{\mathrm{EL}}(k, R). \qquad (9)$$

From MEL, we can finally approximate the expected reduction in loss — the expected VOI (EVOI) — of obtaining query response q_{ij} for items i and j:

$$\begin{aligned}
&\mathrm{EVOI}(R, i, j) \qquad (10) \\
&= -\mathrm{MEL}(R) + \mathrm{E}_{P(\mathbf{w}|R)} \sum_{q_{ij}} \left[P(q_{ij}|\mathbf{w}) \mathrm{MEL}(R \cup \{q_{ij}\})\right] \\
&= -\mathrm{MEL}(R) + \sum_{q_{ij}} \left[\mathrm{E}_{P(\mathbf{w}|R)} P(q_{ij}|\mathbf{w})\right] \mathrm{MEL}(R \cup \{q_{ij}\}).
\end{aligned}$$

The only part of the last expression that we have not covered yet is the computation of $\mathrm{E}_{P(\mathbf{w}|R)} P(q_{ij}|\mathbf{w})$. Recalling the definition of $P(q_{ij}|\mathbf{w})$ from (3) based on the difference of $u_i = u(i|\mathbf{w})$ and $u_j = u(j|\mathbf{w})$, we note this can be computed easily in closed-form as follows. First, we define the difference random variable $d = u_i - u_j$ and note that it has univariate distribution $\mathcal{N}(d; \mu_d, \sigma^2_d)$ where $\mu_d = \sum_d (\mu_{d,\#(i,d)} - \mu_{d,\#(j,d)})$ and $\sigma^2_d = \sum_d (\sigma^2_{d,\#(i,d)} - \sigma^2_{d,\#(j,d)})$ (which follow from well-known sums and differences of normally distributed random variables). Then using the same draw margin $\epsilon \geq 0$ as (3), we use the normal CDF function Φ_{μ_d,σ^2_d} to compute the probability that d exceeds ϵ:

$$\mathrm{E}_{P(\mathbf{w}|R)}[P(Q_{ij} = i \succ j|w)] = 1 - \Phi_{\mu_d,\sigma^2_d}(\epsilon). \qquad (11)$$

Likewise, we compute:

$$\begin{aligned}
&\mathrm{E}_{P(\mathbf{w}|R)}[P(Q_{ij} = i \prec j|w)] = \Phi_{\mu_d,\sigma^2_d}(-\epsilon), \qquad (12) \\
&\mathrm{E}_{P(\mathbf{w}|R)}[P(Q_{ij} = i \sim j|w)] \\
&\qquad = 1 - \mathrm{E}_{P(\mathbf{w}|R)}[P(Q_{ij} = i \succ j|w)] \\
&\qquad\quad - \mathrm{E}_{P(\mathbf{w}|R)}[P(Q_{ij} = i \prec j|w)]. \qquad (13)
\end{aligned}$$

[2] We assume any item is synonomous with its feature vector (e.g., i^*_R and $\mathbf{x}_{i^*_R}$ are used interchangeably).

With this, we now have all of ingredients required to efficiently compute an efficient approximation of the EVOI for use in PE.

4 PE Query Selection Strategies

A *query strategy* simply specifies what comparison query between item i and item j should be asked when given the current query response set $R^n = \{q_{kl}\}$ after n queries have been asked.

The primary aim of any query strategy should be to choose a query so that the updated response set $R^{n+1} = R^n \cup \{q_{ij}\}$ optimally reduces the true loss $(\max_j u^*(j) - i^*_R)$ of stopping after the query response and recommending the optimal item $i^*_{R^{n+1}}$.[3] Of course, the true loss is not actually known to the PE system, so it must heuristically choose queries in an attempt to reduce this loss. In this section, we describe heuristic query strategies for doing so.

We begin first with value of information based heuristics for Bayesian PE based on the derivation in the previous section. If given response set R^n, we simply choose a comparison query between items i and j based on $\arg\max_{i,j} = \mathrm{EVOI}(R^n, i, j)$ defined in (10) using approximations of the Q_{ij} outcomes provided by (11,12,13), we refer to this as *Informed VOI*. Alternately, one might suggest that the approximations may be inaccurate and hence a simple fixed weighting such as (p_1, p_2, p_3) (s.t. $\sum_{i=1}^3 p_i = 1$) respectively for query responses $\{i \succ j; i \prec j; i \sim j\}$ might be better. We call this alternate approximate weighting the *Uninformed VOI* scheme and note that $(0.45, 0.45, 0.1)$ yielded best results and is used in the experiments.

Unfortunately, if there are m items then both Informed VOI and Uninformed VOI must evaluate $O(m^2)$ pairwise queries per preference elicitation stage.[4] Clearly, this will become intractable as m grows very large. Consequently, we provide an additional *Restricted* variant of the above algorithms that at any stage restricts the pairwise query between item i and item j given query response set R^n to include $i^*_{R^n}$. We note the complexity of the *Restricted* variant is $O(m)$ query evaluations per PE stage and as we will see, yields very little loss compared to unrestricted $O(m^2)$ strategies.

[3] This optimal item can be computed from (6) for Bayesian PE systems; for non-Bayesian systems, we explicitly note which item is recommended.

[4] In general, we note that even though the EVOI calculation for a single query q_{ij} requires examining all m items, this computation time is dominated by the more expensive approximate Bayesian update required in $\mathrm{MEL}(R \cup \{q_{ij}\})$, even for large item sets up to $m = 10^4$ items. Hence the number of queries for which EVOI is evaluated effectively determines the time complexity of a PE query selection.

We also define strategy *Simple VOI*: given response set R^n, we simply choose a comparison query between items $i^*_{R^n}$ and j using $\arg\max_j = \widehat{\mathrm{EL}}(R^n, i^*_{R^n}, j)$ defined in (8). This query leads to either j becoming the optimal item or the certainty in j's utility increasing — both reducing expected loss w.r.t. $i^*_{R^n}$.

Aside from the above strategies, we also experimented with the PE query strategies below that use the Bayesian update defined in Section 2.3 and recommend the best expected item, but do not use VOI heuristics:

- *Random Two*: a baseline strategy that randomly picks two items for a query and serves as an upper bound for worst-case performance.
- *Best Two*: picks the current highest and second highest items in expectation. This algorithm works best when repeated queries are explicitly prohibited and is the version reported here.
- *Best & Largest Uncertainty*: selects the current best item and the one with largest uncertainty.

5 Experimental Results

5.1 Datasets

In this section, we present experimental results on three datasets, a synthetic dataset and two real datasets. For synthetic data, we generate items with all combinations of three item attributes of interest, with 2, 2, and 5 choices, respectively, making 20 items total. In this dataset, we assume all attribute combinations are feasible. Two real datasets we used are the *PC* dataset (McGinty and Smyth, 2003) and the *Boston Housing* dataset (Asuncion and Newman, 2007). The *PC* dataset consists of actual 120 PC items, each described in terms of 8 attributes including manufacturer, processor, memory, etc. The Boston housing data has 506 items, each annotated with 13 continuous attributes and 1 binary valued attribute where the continuous attributes have been discretized. We note that neither of the item sets for Boston Housing or PC were fully exhaustive of all attribute combinations, reflecting implicit real-world constraints.

5.2 User Simulation

To simulate the user response process, we drew random utilities for the attribute choice vector $\mathbf{w}$ according to two models: (a) a uniform distribution over $[1, 100]$ for each attribute choice, and (b) a normal distribution with mean μ drawn uniformly from $[1, 100]$ for each attribute choice and covariance $\Sigma = \mathrm{diag}(\frac{\mu}{\mathbf{3}})$. We also experimented with sampling random positive semidef-

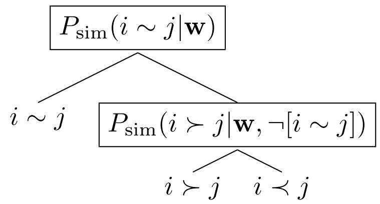

Figure 2: A generative query response model $P_{\text{sim}}(q_{ij}|\mathbf{w})$: left branches indicate true results for the sampled argument of $P(\cdot)$, right branches indicate false. Sampling proceeds from the root at each $\Box$ until a leaf is encountered, the probability of a leaf being the probability of parent decisions $\Box$. Definitions of each $\Box$ are given in (14) and (15).

inite matrices for use as full covariance matrices but noted little deviation in results from (b).

We simulate the user's query response given the following Bradley-Terry confusion model (Bradley and Terry, 1952) extended with indifference. We note that this model *intentionally* does *not* match the query portion of the Bayesian graphical model used for updating in Section 2.3; this will provide evidence of whether inference in our Bayesian PE approach is robust to a different and more complex user response model.

For q_{ij}, the Bradley-Terry model provides the probability of a user responding $i \succ j$ given their attribute weight vector $\mathbf{w}$ and assuming the user is not indifferent between i and j:

$$P_{\text{sim}}(i \succ j|\mathbf{w}, \neg[i \sim j]) = \frac{\exp(\alpha[u(i|\mathbf{w}) - u(j|\mathbf{w})])}{1 + \exp(\alpha[u(i|\mathbf{w}) - u(j|\mathbf{w})])}. \tag{14}$$

Here, $\alpha > 0$ is a user-specific model parameter that must be fit for a specific user or user population. For the simulations, we simply assume $\alpha = 1$.

We model the case of indifference as an exponential distribution

$$P_{\text{sim}}(i \sim j|\mathbf{w}) = \exp(-\beta|u(i|\mathbf{w}) - u(j|\mathbf{w})|), \tag{15}$$

where the probability of indifference peaks at 1 when the true utilities are equal and trails off to 0 as the absolute utility difference increases. As for the Bradley-Terry model, the parameter $\beta > 0$ must be fit for a specific user or user population; for the simulations, we use $\beta = 1$. While many alternate models may yield similar qualitative properties for modeling indifference, we use (15) since it is log-linear in the units of utility as for the Bradley-Terry model.

With both the indifference and preference model pieces in place, we now provide a generative model for $P_{\text{sim}}(q_{ij}|\mathbf{w})$ in Figure 2, where we exploit context-specific independence (CSI) (Boutilier et al., 1996) inherent in the definition of (14) w.r.t. (15) to represent the different cases as paths in a tree. To simulate the user's query response, we first draw a sample from $P_{\text{sim}}(i \sim j|\mathbf{w})$ returning indifference ($i \sim j$) if true, otherwise we sample from $P_{\text{sim}}(i \succ j|\mathbf{w}, \neg[i \sim j])$ returning $i \succ j$ if true, otherwise $i \prec j$ if false.

5.3 Results

All of the following experiments were implemented in Matlab (code available on request), under Windows, using an Intel(R) Core™2 Quad CPU Q9550, 2.83GHz, 3Gb RAM PC. $\epsilon = 10$ for Bayesian updates.

5.3.1 Time per Query

In Table 1, we show the time it took for each of the main PE algorithms to propose a query on the synthetic dataset. With only 20 items, we already see that the full Informed and Uninformed VOI algorithms require unreasonable times for query selection and thus are too slow to evaluate on the larger datasets.

Table 1: Time (ms) required to propose a query on the synthetic dataset.

Query Strategy	Mean	Std
Random Two	0.1	0.0057
Best & Larg. Uncer.	0.7	0.0150
Best Two	1	0.1521
Simplified VOI	2	0.0347
Restr. Inform VOI	1045	2.1168
Informed VOI	10427	9.9465
Uninformed VOI	10365	8.2497

Consequently, we note that it is absolutely crucial for the time efficiency of PE on large item sets to restrict query evaluation to always include the best item; this reduces the query evaluation complexity for m items from $O(m^2)$ to $O(m)$. Of course this naturally raises the question as to how much Restricted Informed VOI loses by limiting VOI query strategies in this way? To answer, very little — after the 2nd query, 100% of all Informed and Uninformed VOI queries were selected from the Restricted set for all of our experiments!

5.3.2 Expected Loss

We show a plot of the normalized average loss ($\max_j u^*(j) - i^*_R$) of all algorithms vs. the number of query responses elicited in Figure 3. That is, on the y-axis, we show for 20 averaged trials what fraction of the total loss was incurred by each algorithm after the x-axis specified number of queries. A result of 0 indicates no loss and is optimal.

The key observations here are that the VOI heuristics always perform the best, with the Restricted Informed VOI heuristic in particular among the top performing query strategies over all domains while being fast enough to run efficiently on the PC and Housing datasets (unlike the unrestricted versions). We also note that Restricted Informed VOI has excellent anytime performance — it always reduces the average loss on each additional query and over all numbers of queries, it is among the best algorithms in terms of quickly reducing average loss.

6 Related Work

Space limitations prevent a thorough literature review; we briefly discuss how related work addresses the five principles from Section 1 as summarized in Table 2.

While a variety of early PE research influenced many of the design decisions in this work (Chajewska et al., 2000, 2001; Boutilier, 2002) such as the Bayesian modeling approach, factorized belief representation, and VOI, these papers typically relied on either *standard gamble* queries requiring users to state their preference over a probability distribution of outcomes or they directly elicit utility values. While theoretically sound, these methods may require high cognitive load for elicitation, and thus are prone to error (Chajewska et al., 2000); we rely on pairwise comparison queries known to require low cognitive load (Conitzer, 2009).

Noise-free methods optimize queries to minimize target regret functions, but assume that no confusion takes place in the user's query response (Conitzer, 2009; Viappiani and Boutilier, 2009). Because these systems cannot always recover from the inevitable confused user response (thus potentially ruling out the true utility), we refer to these methods as non-robust.

While recent work (Doshi and Roy, 2008) has pushed on scalability by extending Boutilier (2002) to exploit symmetries in sequential query optimization (ideas that could be incorporated in future work with our approach), such work has not explicitly addressed a factorized belief representation for scaling to *large asymmetric multiattribute* problems with hundreds of items like the PC and Boston Housing datasets we used.

7 Conclusion

In light of the PE requirements in Section 1, we developed a highly efficient Bayesian PE framework based on TrueSkill for performing efficient closed-form multiattribute utility belief updates — a novel PE approach that facilitated efficient closed-form VOI approximations for PE query selection. This contrasted with related work that failed to satisfy all requirements. As demonstrated on synthetic and real-world data, the Restricted Informed VOI query strategy is real-time, multiattribute, low cognitive load via pairwise queries, robust to noise and mismatched user response simulation models, and scalable — each PE stage evaluates $O(m)$ queries per m items; this is much more efficient than the full $O(m^2)$ VOI query strategies while achieving close to the same accuracy and comparable to the best among the remaining algorithms.

Acknowledgements

We thank Thore Graepel, Thomas Minka, and Ralf Herbrich for interesting discussions about approximate inference and query strategies. We would also like to thank anonymous reviewers for insightful suggestions, especially for providing the justification for omitting shared variables from the Bayesian update. NICTA is funded by the Australian Government's Backing Australia's Ability initiative, and the Australian Research Council's ICT Centre of Excellence program.

References

Asuncion, A. and Newman, D. J. (2007). UCI machine learning repository.

Boutilier, C. (2002). A POMDP formulation of preference elicitation problems. In *AAAI*.

Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in bayesian networks. In *UAI*.

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: The method of paired comparison. *Biometrika*, 39:324–345.

Chajewska, U. and Koller, D. (2000). Utilities as random variables: Density estimation and structure discovery. In *UAI*.

Chajewska, U., Koller, D., and Ormoneit, D. (2001). Learning an agent's utility function by observing behavior. In *UAI*.

Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *AAAI*.

Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191.

Doshi, F. and Roy, N. (2008). The permutable POMDP: fast solutions to POMDPs for preference elicitation. In *AAMAS*.

Herbrich, R., Minka, T., and Graepel, T. (2006). Trueskill^{TM}: A Bayesian skill rating system. In *NIPS*.

Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26.

Keeney, R. L. and Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

McGinty, L. and Smyth, B. (2003). Tweaking critiquing. In *IJCAI Workshop on Intel. Tech. for Personalization*.

Viappiani, P. and Boutilier, C. (2009). Regret-based optimal recommendation sets in conversational recommender systems. In *ACM Conf. on Recomm. Systems*.

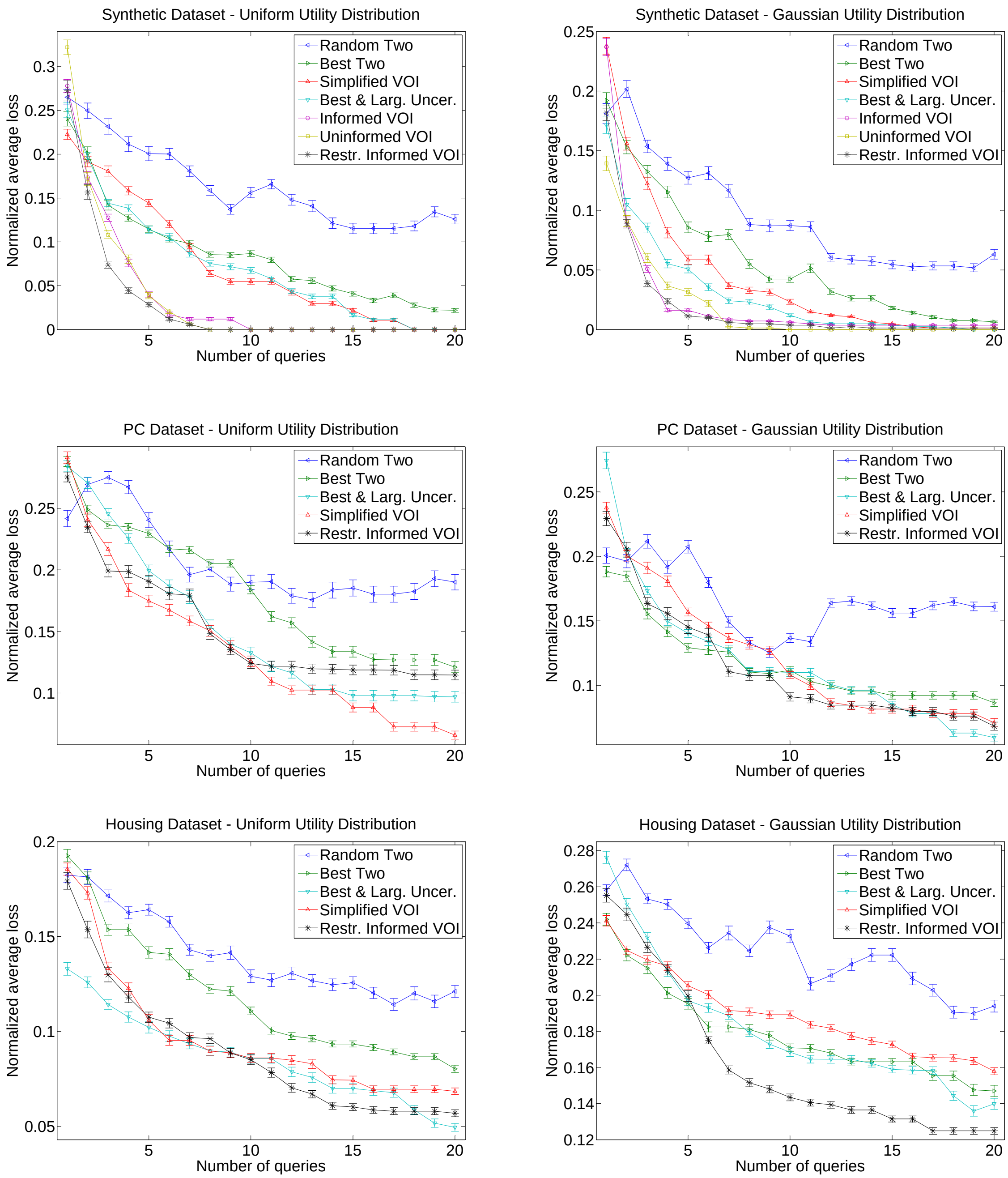

Figure 3: Expected loss vs. number of queries for various PE strategies on two datasets. Error bars indicate standard error. Upper panel: results on the synthetic dataset with uniform utility (left) and diagonal Gaussian utility (right). Middle panel: results on the PC dataset with uniform utility (left) and diagonal Gaussian utility (right). Bottom panel: results on the Housing data set with uniform utility (left) and diagonal Gaussian utility (right).

Table 2: Comparison among PE algorithms in terms of five requirements.

Literature	Real-time	Multiattribute	Low cognitive load	Robustness	Scalable
(Chajewska et al., 2000)		√		√	
(Chajewska et al., 2001)		√		√	
(Boutilier, 2002)				√	
(Doshi and Roy, 2008)	√		√	√	
(Conitzer, 2009)	√		√		√
(Viappiani and Boutilier, 2009)	√	√			
Our approach	√	√	√	√	√