Multiattribute Bayesian Preference Elicitation with Pairwise Comparison Queries

Shengbo Guo[1] and Scott Sanner[2]

[1] ANU/NICTA, Locked Bag 8001, Canberra ACT 2601, Australia
[2] NICTA/ANU, Locked Bag 8001, Canberra ACT 2601, Australia
{Shengbo.Guo,Scott.Sanner}@nicta.com.au
http://users.rsise.anu.edu.au/~sguo/

Abstract. Preference elicitation (PE) is an very important component of interactive decision support systems that aim to make optimal recommendations to users by actively querying their preferences. In this paper, we present three principles important for PE in real-world problems: (1) multiattribute, (2) low cognitive load, and (3) robust to noise. In light of three requirements, we introduce an approximate PE framework based on a variant of 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) for use in query selection strategies. Our VOI query strategy satisfies all three principles and performs on par with the most accurate algorithms on experiments with a synthetic data set.

Keywords: preference elicitation, decision-making under uncertainty.

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. *Multiattribute*: Exploiting the natural attribute structure of services or items in the form of multiattribute utility functions [10] 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.

L. Zhang, J. Kwok, and B.-L. Lu (Eds.): ISNN 2010, Part I, LNCS 6063, pp. 396–403, 2010.

2. *Low cognitive load*: Since the task of utility elicitation is cognitively difficult and error prone [5], 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 [6].
3. *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.

In the following sections, we develop an approximate Bayesian PE framework to satisfy all three of these principles and demonstrate this empirically on a synthetic dataset.

2 Bayesian Preference Elicitation

2.1 User Utility Model

In multiattribute utility theory (MAUT) [10], 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.,

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

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 $\mathbf{w}$ [3] 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:

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

We assume the vectors μ and σ represent the respective mean and standard deviation for the normal distribution over each corresponding attribute choice in $\mathbf{w}$.

2.2 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 $R^{n+1} = R^n \cup \{q_{ij}\}$:

$$\begin{aligned} P(\mathbf{w}|R^{n+1}) &\propto P(q_{ij}|\mathbf{w}, R^n)P(\mathbf{w}|R^n) \\ &\propto P(q_{ij}|\mathbf{w})P(\mathbf{w}|R^n) \end{aligned} \tag{3}$$

Assuming that our query likelihood $P(q_{ij}|\mathbf{w})$ is modeled as an indicator function, 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 (3) leads to a slight variation on the *TrueSkill*™ [8] graphical model for multiattribute PE shown in Figure 1.

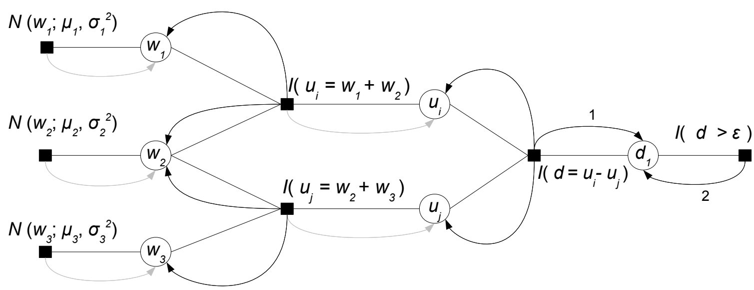

Fig. 1. PE factor graph variant of TrueSkill for $q_{ij} = i \succ j$. Items i and j have two attribute choices each with respective weights (w_1, w_2) and (w_2, w_3) (note that i and j share the common attribute choice with weight w_2). The posterior over (w_1, w_2, w_3) can be inferred with the following message passing schedule: (1) messages pass along *gray* arrows from left to right, (2) the marginal over d is updated via message 1 followed by message 2 (which required moment matching), (3) messages pass from right to left along *black* arrows.

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 that 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 $\sigma^n \to \mathbf{0}$.

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) [9]. VOI plays an important role in many Bayesian PE strategies, as first proposed in [5] 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[1]:

$$i^*_R = \arg\max_i \; \mathrm{E}_{P(\mathbf{w}|R)}[u(i|\mathbf{w})] = \arg\max_i \; u(i|\mu^R). \tag{4}$$

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 item the best item i^*_R instead of recommending item k, we evaluate the following expectation:

$$\mathrm{EL}(k, R) = \mathrm{E}_{P(\mathbf{w}|R)} \left[\max\left(0, u(k|\mathbf{w}) - u(i^*_R|\mathbf{w})\right)\right] \tag{5}$$

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 (4) as a surrogate for its true utility, leading to the *closed-form* calculation:

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

$$\widehat{\mathrm{EL}}(k,R) = (\mu_{i_R^*} - \mu_k)(1 - \Phi_{\mu_k,\sigma_k^2}(\mu_{i_R^*})) - \frac{\sigma_k}{\sqrt{2\pi}} \exp\left(-\frac{(\mu_{i_R^*} - \mu_k)^2}{2\sigma_k^2}\right) \quad (6)$$

Here, Φ_{μ_k,σ_k^2} is the normal CDF, $\mu_k = \sum_d \mu_{d,\#(k,d)}$, $\sigma_k^2 = \sum_d \sigma^2_{d,\#(k,d)}$, 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 determine the maximum expected loss (MEL) we might incur by recommending i_R^* instead of *some* other k:

$$\mathrm{MEL}(R) = \max_k \widehat{\mathrm{EL}}(k,R). \quad (7)$$

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:

$$\mathrm{EVOI}(R,i,j) = -\mathrm{MEL}(R) + \sum_{q_{ij}} \left[\mathrm{E}_{P(\mathbf{w}|R)} P(q_{ij}|\mathbf{w})\right] \mathrm{MEL}(R \cup \{q_{ij}\}) \quad (8)$$

4 PE Query Selection Strategies

A query strategy 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.

Ideally, we are supposed to propose the query q_{ij} such that the user's response can maximally reduce the expected loss with regard to the belief on $\mathbf{w}$, i.e.,

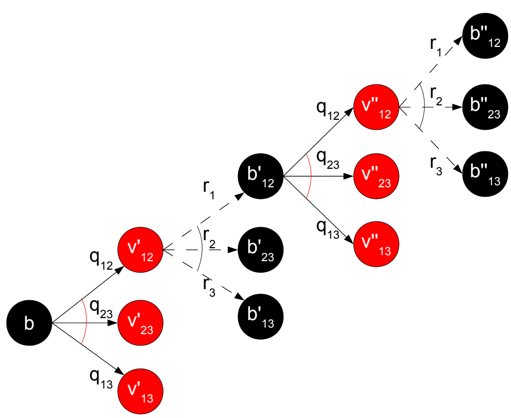

Fig. 2. Illustration of two-step look ahead with maximization and expectation query strategy (incomplete for illustration purpose). For a given belief state b, suppose there are three possible queries, a_1, a_2, a_3, each with three possible responses, we represent the intermediate belief state of asking a_1 at b by v_1'. For given v_1', we use b_{11}' to represent the posterior belief when observing response r_1. Same operation applies to b_{11}' and yields the two-step look ahead search tree. The operators associates with red circle and black circle represent *expectation* and *maximization*, respectively.

the expected value of information. We adopt a two-step look ahead strategy involving maximization and expectation to find the best query (Figure 2), and we call it PE-ME. Given the current belief state and each possible query, we simulate the user's response and evaluate the posterior belief state. Then, we apply the same operation to each posterior belief state. Once we compute the EVOI for each query and possible user's response, we compute the expectation of EVOI of asking a query with different user responses, and select the query that maximizes the EVOI for each query. Then, we repeat the expectation and maximization operations, and select the query with maximal EVOI.

Aside from the above strategies, we also experimented with the PE query strategies that do not use VOI heuristics. The first query strategy is *Random Two*, a baseline strategy that randomly picks two items for a query and serves as an upper bound for worst-case performance. The second is *Preference elicitation upper confidence bounds (PE-UCB)*, which queries the items with the largest and second largest upper confidence bounds. Given μ (mean) and σ (standard deviation) of the item belief, the upper confidence bounds are $\mu + c\sigma$, where $c > 0$ is a constant. We use $c = 1$ (PE-UCB(1)), $c = 2$ (PE-UCB(2)), and $c = 3$ (PE-UCB(3)), respectively.

5 Experimental Results

5.1 Data Set and User Simulation

We evaluate our approach using a synthetic data set. For this data set, 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.

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 sampling random positive semidefinite matrices for use as full covariance matrices.

5.2 Results

All of the following experiments were implemented in Matlab (code available on request), under Windows, using an Intel(R) Core$^{\text{TM}}$2 Quad CPU Q9550, 2.83GHz, 3Gb RAM PC. $\epsilon = 5$ for Bayesian updates.

We show a plot of the normalized average loss ($\max_j u^*(j) - u^*(i^*_R)$) of all algorithms vs. the number of query responses elicited in Figure 3. That is, on the y-axis, we show for 50 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.

For uniformly distributed utility distributions, 'PE-ME' query strategy outperforms 'Random Two' and 'PE-UCB(c)' when less than 4 queries were asked.

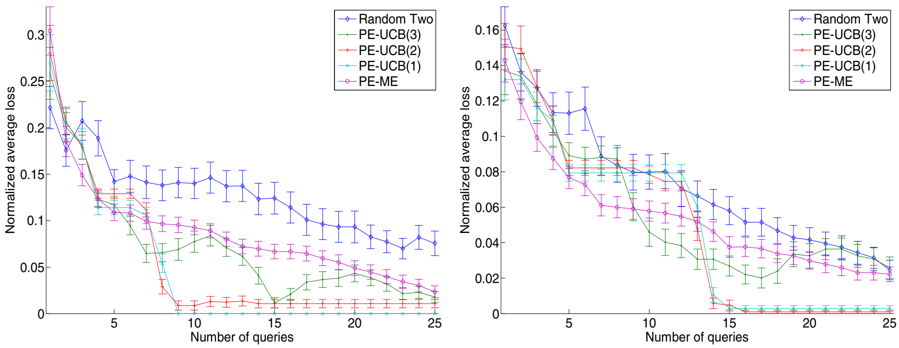

Fig. 3. Normalized average loss vs. number of queries for various PE strategies. Error bars indicate standard error.

After that, all the 'PE-UCB(c)' strategies start to get better performance. Amongst those strategies, the performance of 'PE-UCB(3)' is the best when less than 8 queries are asked, however, it is not stable. Since the ninth queries, 'PE-UCB(2)' and 'PE-UCB(1)' become the best two. For Gaussian utility functions, 'PE-ME' achieves the best performance till ninth query is asked. It turns out that 'PE-ME' is very effective in reducing normalized average when a few queries are asked.

6 Related Work

Space limitations prevent a thorough literature review; we briefly discuss how related work addresses the three PE principles, which is summarized Section in 1. While various early PE research influenced many of the design decisions in this work [5, 4, 1] 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

Table 1. Comparison among PE algorithms in terms of three requirements

Literature	Multiattribute	Low cognitive load	Robustness
[5]	√		√
[4]	√		√
[1]			√
[7]		√	√
[6]		√	
[12]	√		
Our approach	√	√	√

methods may require high cognitive load for elicitation, and thus are prone to error [5]; we rely on pairwise comparison queries known to require low cognitive load [6].

7 Conclusion

In light of the PE requirements, we developed an effective Bayesian PE framework based on a variant of 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 the synthetic data set, the 'PE-ME' query strategy is multiattribute, low cognitive load via pairwise queries, robust to noise.

References

[1] Boutilier, C.: A POMDP formulation of preference elicitation problems. In: AAAI, pp. 239–246 (2002)

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

[3] Chajewska, U., Koller, D.: Utilities as random variables: Density estimation and structure discovery. In: UAI, pp. 63–71 (2000)

[4] Chajewska, U., Koller, D., Ormoneit, D.: Learning an agent's utility function by observing behavior. In: ICML, pp. 35–42 (2001)

[5] Chajewska, U., Koller, D., Parr, R.: Making rational decisions using adaptive utility elicitation. In: AAAI, pp. 363–369 (2000)

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

[7] Doshi, F., Roy, N.: The permutable POMDP: fast solutions to POMDPs for preference elicitation. In: AAMAS, vol. 1, pp. 493–500 (2008)

[8] Herbrich, R., Minka, T., Graepel, T.: Trueskill^{TM}: A Bayesian skill rating system. In: NIPS, pp. 569–576 (2006)

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

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

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

[12] Viappiani, P., Boutilier, C.: Regret-based optimal recommendation sets in conversational recommender systems. In: RecSys (2009)