

# Preference Elicitation with Soft Attributes in Interactive Recommendation

ERDEM BIYIK, Thomas Lord Department of Computer Science, University of Southern California, USA

FAN YAO, University of Virginia, USA

YINLAM CHOW\*, Google Research, USA

ALEX HAIG, Google Research, USA

CHIH-WEI HSU, Google Research, USA

MOHAMMAD GHAVAMZADEH, Google Research, USA

CRAIG BOUTILIER, Google Research, USA

*Preference elicitation* plays a central role in interactive *recommender systems*. Most preference elicitation approaches use either item queries that ask users to select preferred items from a slate, or attribute queries that ask them to express their preferences for item characteristics. Unfortunately, users often wish to describe their preferences using *soft* attributes [4] for which no ground-truth semantics is given. Leveraging *concept activation vectors* [21] for soft attribute semantics, we develop novel preference elicitation methods that can accommodate soft attributes and bring together both item and attribute-based preference elicitation. Our techniques query users using both items and soft attributes to update the recommender system’s belief about their preferences to improve recommendation quality. We demonstrate the effectiveness of our methods *vis-à-vis* competing approaches on both synthetic and real-world datasets.

Additional Key Words and Phrases: Preference elicitation, Interactive recommender systems, Personalized semantics, Concept activation vectors (CAVs)

## 1 INTRODUCTION

Recommender systems (RSs) play a central role in connecting users to products, content and services by predicting user *preferences* for candidate items. While practical RSs are often trained using implicit user feedback on recommended items (e.g., clicks, consumption, ratings), increasingly interactive and conversational systems explicitly elicit user preferences to more effectively guide recommendations. Approaches to *preference elicitation* (PE) include both item-based [3, 41] and attribute-based methods [13, 46], but rarely are these considered in conjunction. Moreover, attribute-based methods generally rely on *hard attributes*, those for which a definitive source of ground truth associates attribute values (e.g., actors in movies, color of a product) with items. Unfortunately, attributes with which users often wish to describe their preferences are *soft* [4]—there is no source of ground truth associating such attribute values to items; instead this association, or *semantics*, must be learned. For instance, information about movie attributes like ‘funny,’ ‘thought-provoking,’ or ‘inspiring’ may only be learnable via sparse, noisy user comments, reviews, or tags. Because of this, practical attribute-based PE must account for the inherent noise and uncertainty in the semantics of soft attributes.<sup>1</sup>

In this work, we develop PE methods that can accommodate such soft attributes in item-based and attribute-based PE. To do so, we leverage the recent method by Göpfert et al. [21], who use *concept activation vectors* (CAVs) [28] to discover the semantics of soft attributes w.r.t. a RS’s item representation. We show how to incorporate this semantics into interactive RSs with *active critiquing*: given a slate of items, our techniques determine the soft attributes about which to elicit user feedback (e.g., “do you prefer a funnier or a less funny movie than this slate of movies?”), interpret the user’s response relative to this semantics, and update RS beliefs about the user’s preferences. Because the semantics are

\*Contact author.

<sup>1</sup>Soft attribute usage may also be *subjective* [21], an issue we set aside in this work.

learned from sparse and noisy data, the RS may have varying degrees of certainty or confidence in the CAV semantics of different soft attributes. This should, in turn, impact how the RS interprets user responses to attribute-based PE-queries and updates its beliefs. We extend the CAV semantics to handle such uncertainty and incorporate this into PE-driven belief updates.

Our key contributions are as follows: (i) we leverage CAV semantics to extend attribute-based PE for interactive RSs to handle soft attributes (in contrast to Göpfert et al. [21], who model user-initiated critiques); (ii) we propose two novel query types: queries over attributes, and their combination with well-studied item queries, along with the human response models to these new query types, which enable us to efficiently update RS beliefs over user preferences in our PE framework; (iii) we propose several PE (query) selection schemes and optimization methods that balance information gathering and recommendation quality; and (iv) we extend our techniques to incorporate uncertainty in an attribute’s CAV semantics. We illustrate the efficacy of our methods on both synthetic data and MovieLens data [22].

## 2 PROBLEM FORMULATION

In this section, we outline our problem formulation and key assumptions, then briefly discuss related work.

**2.1 Preference Prediction.** We assume a standard *collaborative filtering (CF)* setting [25], where users  $u \in \mathcal{U}$  rate items  $i \in \mathcal{I}$  with  $r_{u,i} \in \mathcal{R} \cup \{0\}$ . Here  $\mathcal{R}$  is the set of possible ratings (e.g., 1–5 stars) and  $r_{u,i} = 0$  indicates that user  $u$  has not rated item  $i$ . Let the *ratings dataset* be  $\mathbf{R} = \{(u, i, r_{u,i}) : r_{u,i} \neq 0\}$ .

The RS learns user and item representations from the ratings dataset  $\mathbf{R}$  using some form of CF. Let  $\mathcal{X} \subseteq \mathbb{R}^d$  be a user-item embedding space. An *item embedding*  $\phi_I : \mathcal{I} \rightarrow \mathcal{X}$  maps each item  $i \in \mathcal{I}$  into a vector representation  $\phi_I(i)$  of its (latent) attributes in  $\mathcal{X}$ ; note that this embedding is typically not interpretable. We assume that user preferences are captured by a similar mapping  $\phi_U : \mathcal{U} \rightarrow \mathcal{X}$ . However, to capture the preference variations of users due to their latent state (e.g., mood, activity, location), the RS learns an *embedding distribution* for each user, i.e.,  $P_U : \mathcal{U} \rightarrow \mathbb{P}(\mathcal{X})$ . As such, the embedding  $\phi_U(u) = \phi_u \sim P_U(u)$  in  $u$ ’s session is a sample from her embedding distribution. Since  $\phi_u$  is unknown to the RS, we use PE to uncover it.<sup>2</sup>

To learn user and item embeddings, we use a *two-tower model* (or *dual encoder*) in which users and items are passed through separate but co-trained deep neural networks (DNNs) [49, 50]. The item encoder outputs  $\phi_I(i)$  for any item  $i \in \mathcal{I}$ , while the user encoder outputs a  $d$ -dimensional multivariate Gaussian distribution  $P_U(u) = \mathcal{N}(\phi_{\mu,U}(u), \phi_{\sigma,U}(u))$  for any user  $u \in \mathcal{U}$ , where  $\phi_{\mu,U}(u)$  is the mean user embedding vector and  $\phi_{\sigma,U}(u)$  is a (lower-triangular) scale matrix. Suitable methods include probabilistic matrix factorization [40] and certain forms of neural CF [6, 49]. These methods often assume linear user utility, i.e., (predicted) user-item affinity is  $\hat{r}_{u,i} = \phi_u^\top \phi_I(i)$ . Hence, we treat  $\phi_u$  as the parameters of  $u$ ’s utility function (w.r.t. the learned item representation). Our Gaussian user embedding gives the following *belief state* over the user’s utility function:  $P(r_{u,i} | P_U, \phi_I) = \mathcal{N}(\phi_{\mu,U}(u)^\top \phi_I(i), \phi_I(i)^\top \phi_{\sigma,U}(u)^\top \phi_{\sigma,U}(u) \phi_I(i))$ . We train these encoders by minimizing negative log-likelihood:  $\mathcal{L} = -\frac{1}{|\mathbf{R}|} \sum_{(u,i,r_{u,i}) \in \mathbf{R}} \log P(r_{u,i} | P_U, \phi_I)$ .

**2.2 Soft Attributes & Tags.** In addition to *hard (known or objective) attributes*, users often describe items using *soft attributes* which have no “ground truth” semantics or association with items. Unlike hard attributes, such as ‘genre,’ ‘artist,’ or ‘director,’ these terms are not part of an agreed-upon formal specification of an item. They are neither applied universally to all items, nor by all users.

Some RSs support user-supplied *tags* (e.g., MovieLens), which may be specified directly in the RS, or extracted from user descriptions, reviews, or other sources. A tag can denote a hard or soft attribute. We let  $\mathcal{T}$  be the set of

<sup>2</sup>Our techniques can be applied, *mutatis mutandis*, to the RS’s uncertainty about a user’s *stationary preferences* and its gradual refinement of beliefs over time, rather than assuming that preferences actually change (are resampled) with each session.

tags that users may adopt to describe items. We assume that tags are used in a binary way—users simply choose to apply a tag or not. However, the actual attributes corresponding to tags can be ordinal or cardinal. Indeed, the user study in Balog et al. [4] shows that soft attributes often exhibit “relative” degrees. For example, applying the tag ‘violent’ may reflect that a user believes a movie exhibits a *degree of violence* exceeding her tolerance. Let  $t_{u,i,g} = 1$  if user  $u$  applies tag  $g \in \mathcal{T}$  to item  $i$ , and  $t_{u,i,g} = 0$  otherwise. For each  $g \in \mathcal{T}$ , let  $\mathbf{T}_g = \{(u, i) : t_{u,i,g} = 1\}$  and  $\mathbf{T}_{\bar{g}} = \{(u, i) : t_{u,i,g} = 0, \exists g' \in \mathcal{T} : t_{u,i,g'} = 1, \exists i' \in \mathcal{I} : t_{u,i',g} = 1\}$ . Pair  $(u, i)$  belongs to  $\mathbf{T}_g$ , if user  $u$  applies tag  $g$  to item  $i$ , and belongs to  $\mathbf{T}_{\bar{g}}$ , if  $u$  applies a tag  $g' \neq g$  to  $i$  and tags another item  $i' \neq i$  with  $g$ . Tags are usually strictly sparser than ratings, so we assume  $\sum_g |\mathbf{T}_g| \ll |\mathbf{R}|$ .

**2.3 Concept Activation Vectors in RSs.** Modern ML models usually learn complex and non-transparent representations of concepts, an issue addressed by work on interpretable representations [42]. Kim et al. [28] develop an approach that tries to find a correspondence between internal state of an ML model (e.g., a DNN used for image classification)—in the form of a *concept activation vector (CAV)*—and a human-interpretable concept.

Göpfert et al. [21] apply CAVs to identify the semantics of tags w.r.t. the item representation learned by CF models in RSs. We adopt this approach in our work: for a tag  $g \in \mathcal{T}$ , we attempt to find a CAV  $\phi_g \in \mathcal{X}$  in the embedding space, and use it to determine if the CF model has learned a representation of (the attribute corresponding to)  $g$ .<sup>3</sup> We train the CF model and learn CAVs separately, similar to methods that build attribute models on top of embeddings for cold-start [16, 38], and in contrast to those that jointly train attribute models [31, 48]. This separation is motivated by the hypothesis that if a tag is useful for understanding user preferences (and thus for PE), the CF model will have learned a representation of it [21]. After training a two-tower model, we use its item tower  $\phi_I$  to learn CAVs. For a tag  $g \in \mathcal{T}$ , we construct a training set  $\mathbf{D}_g$  in which positive instances ( $y = +1$ ) are items in  $\mathbf{T}_g$  and negatives ( $y = -1$ ) are those in  $\mathbf{T}_{\bar{g}}$ . We then learn the CAV  $\phi_g$  for  $g$  by learning a (regularized) logistic regressor:

$$\mathcal{L}(\phi_g; \mathbf{D}_g) = \sum_{(i,y) \in \mathbf{D}_g} \log(1 + e^{-y\phi_g^\top \phi_I(i)}) + \frac{\lambda}{2} \phi_g^\top \phi_g. \quad (1)$$

The induced CAV  $\phi_g$  is the normal to the separating hyperplane of this classifier, and offers a directional semantics for (the attribute corresponding to) tag  $g$  in the item embedding space. Specifically, its dot product with  $\phi_I(i)$ , or  $g$ -score  $c_g(i) = \phi_g^\top \phi_I(i)$ , quantifies the degree to which item  $i$  satisfies tag  $g$ , (e.g., how violent or funny a movie is). We use the  $g$ -score to measure the *quality* (or usefulness) of CAV  $\phi_g$ , defining  $Q(\phi_g; \mathbf{D}_g)$  to be the fraction of the item pairs  $\{(i_1, i_2) \mid (i_1, +1) \in \mathbf{D}_g, (i_2, -1) \in \mathbf{D}_g\}$  for which  $c_g(i_1) \geq c_g(i_2)$ . Given sparsity and noise in the tag data, we may be more confident in some CAVs than in others. We discuss CAV uncertainty below.

**2.4 Preference Elicitation.** Traditional CF-based RSs uncover user preferences indirectly, using indirect feedback (e.g., clicks, ratings, purchases). *Interactive RSs*, by contrast, actively elicit a user’s preference by asking her questions, responses to which are used to refine its estimate (or *belief*) about her utility function and improve quality of future recommendations [23]. At suitable points, an interactive RS selects a *query*  $q$  from a *query space*  $\mathcal{Q}$  that reveals some aspects of the user’s preferences. For example, in item-based PE, users may be asked to rate a particular item [10, 51], in which case  $\mathcal{Q} = \mathcal{I}$ , or to choose an item from a slate  $S$  that she most prefers [45], in which case  $\mathcal{Q}$  is the set of all subsets of  $\mathcal{I}$  of size  $|S|$ . In attribute-based PE [14], a user can express her preferences with more abstract semantic attributes. We describe several types of elicitation queries in Section 3.

<sup>3</sup>While a tag  $g$  is simply a *token* used by users to label items, its corresponding soft attribute is a *property* of the items themselves, which can take a real-value. For conciseness, we sometimes refer to the semantics (or representation or CAV) of a tag  $g$  as shorthand for the semantics of the attribute corresponding to  $g$ .

**2.5 Related Work.** Our work assumes an underlying CF method, and can be applied to matrix factorization methods [40] or more recent DNN-based approaches [49, 50]. A number of approaches have been proposed for interpreting tags or attributes in RSs, often using tag or review data [19, 31, 32]. Some works focused on learning tag semantics w.r.t. a learned recommendation model, including those that jointly learn both, or that derive tag meanings using a pre-trained CF model [16, 34, 37]. The model of Göpfert et al. [21], which we employ in this work, is of the latter form.

The literature on PE is substantial. Most related to our work are those which use Bayesian methods for eliciting user preferences [2, 8]. They maintain a *belief state* over user preferences and refine it using PE queries. Item-based methods include, say, active CF methods that ask users to explicitly rate specific items [10, 51], and item-selection methods that ask users to compare two items or state which item is most preferred within some set [9, 11, 45]. Attribute-based techniques include, for instance, example-critiquing schemes [14, 46]. Most work in PE deals with hard attributes, while we focus on soft attributes. One exception is the work of Radlinski et al. [35], who develop a methodology for relating soft attribute usage in conversational RSs to user preferences.

### 3 QUERIES AND USER RESPONSES

We first describe two novel *query* types that we develop as well as the item queries. These queries can be used to elicit information about a user’s preferences for specific items or their (hard or soft) attributes. For each type, we develop *response models* that relate a user’s (stochastic) response to their underlying preferences, using the CAV semantics to interpret the user’s behaviour w.r.t. attributes. For ease of exposition, we assume that the CAV semantics is *known* by both the RS and the user, but derive updates when there is uncertainty in the CAV semantics below.

**3.1 Attribute Queries.** An *attribute query*  $q = (S, g)$  consists of a *slate* of items  $S$  and a tag  $g$ . The RS presents the slate to the user and asks her if she prefers items, relative to those in  $S$ , that are more/less—i.e., exhibit a greater degree of— $g$ ’s attribute (e.g., “Would you prefer movies that are more/less thought-provoking than those in  $S$ ?”). The user responds to positively  $\rho = +1$  (i.e., more) or negatively  $\rho = -1$  (i.e., less). Slate  $S$  can also be viewed as a recommendation set, not just a query (see Section 5).

Our user response models for attribute queries assume user  $u$  relates items in  $S$  to her “most preferred” or *target* item given her underlying utility function  $\phi_u$ . While  $u$  would like to guide the RS to her most preferred item  $\operatorname{argmax}_{i \in \mathcal{I}} \phi_u^\top \phi_I(i)$ , this would assume an unrealistic level of familiarity with available items  $\mathcal{I}$  by  $u$ . Instead, we consider a model where  $u$  targets a hypothetical *ideal* item,  $\phi_{I,u}^* \in \operatorname{argmax}_{\phi \in \Gamma} \phi_u^\top \phi$ , w.r.t. some mildly constrained space  $\Gamma \subset \mathcal{X}$  unrelated to  $\mathcal{I}$ .<sup>4</sup> In lieu of a detailed familiarity model for  $u$ , we capture some familiarity with  $\mathcal{I}$  by constraining her *target* item as follows:  $\phi_{I,u}^* \in \operatorname{argmax}_{\phi \in \Gamma} \phi_u^\top \phi$ , s.t.  $\|\phi\|_2 \leq \max_{i \in \mathcal{I}} \|\phi_I(i)\|_2$ , whose solution is

$$\phi_{I,u}^* = \max_{i \in \mathcal{I}} \|\phi_I(i)\|_2 \cdot \phi_u / \|\phi_u\|_2. \quad (2)$$

The target item  $\phi_{I,u}^*$  may fall outside of  $\mathcal{I}$ , but this allows the user to direct the RS more meaningfully.

For attribute query  $q = (S, g)$ , in our *mean-slate response model*,  $u$  compares her target item  $\phi_{I,u}^*$  with the mean embedding of items in  $S$ ,  $\phi_{I,\bar{S}} = \frac{1}{|\bar{S}|} \sum_{i \in \bar{S}} \phi_I(i)$ . If the target exhibits more of attribute  $g$  than the mean slate, i.e., the target’s  $g$ -score  $c_g(\phi_{I,u}^*)$  is greater than  $c_g(\phi_{I,\bar{S}})$ , then  $u$  responds positively ( $\rho = +1$ ). User response is estimated by the following *probit* model [12, 43]:

$$P(\rho = +1 \mid q, \phi_u) = \Phi \left( c_g \left( \phi_{I,u}^* - \phi_{I,\bar{S}} \right) / \sigma_g \right), \quad (3)$$

<sup>4</sup>For example, a user will not target a movie that is simultaneously maximally funny, though-provoking, and scary.

where  $\epsilon_g$  is a zero-mean Gaussian noise with variance  $\sigma_g^2$ , and  $\Phi(\cdot)$  is the standard Gaussian CDF.

In our second *mean-probability response model*,  $u$  compares the  $g$ -score of her target with that of each item  $i \in S$ , and responds using the weighted average of differences:

$$P(\rho = +1 \mid q, \phi_u) = \frac{1}{|S|} \sum_{i \in S} \lambda_i \cdot \Phi \left( c_g \left( \phi_{I,u}^* - \phi_I(i) \right) / \sigma_g \right), \quad (4)$$

where  $\lambda_i > 0$  is a response weight s.t.  $\sum_{i \in S} \lambda_i = 1$ . While mean-slate model is intuitive, the flexibility of mean-probability may better capture nuances in user behavior (see *IpA* below).

**3.2 Item Queries.** An *item query* (or choice query)  $q = S$  presents a slate  $S$  to  $u$  and asks her which item is preferred [5, 45]. User response  $\rho = i$  is given by a standard *multinomial logit* model w.r.t.  $\phi_u$  [5, 12]:

$$P(\rho = i \mid q, \phi_u) = \frac{\exp(\phi_I(i)^\top \phi_u / T)}{\sum_{j \in S} \exp(\phi_I(j)^\top \phi_u / T)}, \quad (5)$$

where  $T$  is a temperature parameter.

**3.3 Item-plus-Attribute Queries.** *Item-plus-Attribute (IpA) queries* combine attribute and item queries, and have the same form  $q = (S, g)$  as attribute queries. The user is first asked to select her preferred item  $i_S^*$  in  $S$  (per item queries), then asked to critique  $i_S^*$  w.r.t.  $g$  (per attribute queries). The user responds with  $\rho = (i, +1)$  or  $\rho = (i, -1)$  by comparing  $\phi_{I,u}^*$  to  $i_S^*$  (rather than the slate) via  $g$ -scores, and otherwise responds as in attribute queries:

$$\begin{aligned} &P(\rho_1 = i \mid S, \phi_u) \times P(\rho_2 = y \mid g, \phi_u, \rho_1 = i) \\ &= \frac{\exp(\phi_I(i)^\top \phi_u / T)}{\sum_j \exp(\phi_I(j)^\top \phi_u / T)} \times \Phi \left( y \cdot c_g \phi_{I,u}^* - \phi_I(i) / \sigma_g \right). \end{aligned} \quad (6)$$

**3.4 CAV Uncertainty.** In contrast to hard attributes, the semantics of soft attributes are typically estimated using sparse, noisy data (e.g., as in our use of tag data above). As a result, CAVs constructed for different tags may be characterized by varying degrees of uncertainty. For instance, if CAV  $\phi_g$  for tag  $g$  is based on a large amount of tag data with little disagreement in usage, the RS should be more confident in  $\phi_g$  than in the CAV  $\phi_{g'}$  for a tag  $g'$  whose data is sparser or less consistent. Abstractly, we assume that the RS has a *CAV belief (distribution)*  $P_g(\phi_g \mid \mathbf{D}_g)$  reflecting this uncertainty (where  $\mathbf{D}_g$  is the tag data used to train  $\phi_g$ ). We do not require a specific mechanism for generating this belief, but Bayesian logistic regression (augmenting Eq. 1) [26] or Bayesian learning-to-rank [29] are suitable methods.

The response models for attribute and IpA queries above depend on the CAV/semantics of the attribute/tag in question. In what follows, we assume that user  $u$  responds using some true, underlying CAV  $\phi_g$ .<sup>5</sup> To reflect this additional uncertainty in a user response, the RS interprets it w.r.t. its CAV belief. Specifically,  $P(\rho \mid q, \phi_u)$  in Eqs. 3 and 6 requires taking an expectation over possible CAVs  $\phi_g$  w.r.t.  $P_g(\phi_g \mid \mathbf{D}_g)$ . These response probabilities are then used to update RS beliefs about  $u$ 's utility (Section 4) and to choose queries (Section 5) when the CAVs are uncertain.

Specifically,  $P(\rho \mid q, \phi_u)$  in Eqs. 3, 4 and 6 requires taking an expectation over possible CAVs  $\phi_g$  (or equivalently, scoring functions  $c_g$ ) w.r.t.  $P_g(\phi_g \mid \mathbf{D}_g)$ . These response probabilities are then used to update beliefs about  $u$ 's utility (Section 4, by Bayesian methods) and selecting optimal queries (Section 5) when (some or all) the CAVs are uncertain. See Appendix A for further details.

<sup>5</sup>Different users may have different interpretations of an attribute. Such *subjectivity* can be uncovered using CAVs [21], but we do not consider this here.

#### 4 USER BELIEF STATE MODELING

Bayesian methods generally maintain a *belief state* or distribution over a user’s utility function [9, 44]. In our setting, the belief state for user  $u$  is initially  $P_U(u)$ , and is refined as  $u$  responds to PE queries. Given response  $\rho$  to query  $q$ , we update our belief about  $u$  in the standard Bayesian fashion:

$$P_U(u | \rho, q) := P(\phi_u | \rho, q) \propto P(\rho | q, \phi_u) P_U(u).$$

The prior  $P_U(u)$  is the Gaussian user embedding learned by our two-tower model (Section 2). Let  $\mathcal{H}^{(K)} = \{(q^{(k)}, \rho^{(k)})\} \cup \mathcal{H}^{(K-1)}$  with  $\mathcal{H}^{(0)} = \emptyset$ , be a history of  $K$  queries and user responses. Given  $\mathcal{H}^{(K)}$ , the RS’s posterior for  $u$  is

$$P_U(u | \mathcal{H}^{(K)}) = P(\phi_u | \mathcal{H}^{(K)}) \propto P_U(u) \prod_{k=1}^K P(\rho^{(k)} | q^{(k)} \phi_u). \quad (7)$$

This assumes conditional independence of responses given  $\phi_u$ . Unlike the prior, generally the posterior is not Gaussian, and thus, we use the following two methods to tractably approximate it.

##### 4.1 Parameterized Posterior

We can approximate the posterior with some parameterized distribution  $P(u; \theta)$ . To do so, we sample from the true unnormalized posterior using Metropolis-Hastings [15] or Hamiltonian Monte Carlo (HMC). These samples  $\{\phi_{u,i}\}_{i=1}^n$  are then used to estimate parameter  $\theta$  by maximizing the log-likelihood

$$\begin{aligned} L(\theta | \mathcal{H}^{(K)}) &= \sum_{i=1}^n \log P(\phi_{u,i} | \mathcal{H}^{(K)}; \theta) \\ &= \sum_{i=1}^n \log P(\rho^{(K)} | q^{(K)}, \phi_{u,i}; \theta) + \log P(\phi_{u,i} | \mathcal{H}^{(K-1)}; \theta). \end{aligned} \quad (8)$$

We test two different variants of the sampling and posterior update methods in this scheme. In the first *batch* method, we generate a large set of samples and use them to update the posterior offline, i.e., for each query no additional sample of posterior belief is generated before the posterior is updated. In the second *iterative* method, we generate fewer samples from our initial posterior, update the posterior with these samples, and then re-generate new samples with this updated posterior. This process is repeated several times for each query. While the iterative method is more computationally expensive, our experiments (Section 6, Figure 5) demonstrate its improved data efficiency over the batched counterpart.

##### 4.2 Gaussian Posterior

The posterior can be “assumed” to be Gaussian  $\mathcal{N}(\hat{\phi}_{\mu,U}(u), \hat{\phi}_{\sigma,U}(u))$  using the Laplace approximation [7, 30, 47]. For this, we treat the posterior mean  $\hat{\phi}_{\mu,U}(u)$  as the mode of the true log-posterior, i.e.,

$$\begin{aligned} \hat{\phi}_{\mu,U}(u) &\in \operatorname{argmax}_{\phi \in \mathbb{R}^d} \left[ \sum_{k=1}^K \log P(\rho^{(k)} | q^{(k)}, \phi) \right. \\ &\quad \left. - \frac{1}{2} (\phi - \phi_{\mu,U}(u))^\top (\phi_{\sigma,U}(u)^\top \phi_{\sigma,U}(u))^{-1} (\phi - \phi_{\mu,U}(u)) \right]. \end{aligned} \quad (9)$$

This optimization may not be convex, but its local optima can be found efficiently because its gradient can be written in closed-form.  $\hat{\phi}_{\mu,U}(u)$ . In Figure 2 in Section 6, we numerically compare the performance of the Gaussian posterior with the parameterized (categorical) posterior. The discrete posterior model outperforms its Gaussian counterpart with both

attribute and IpA response models. We therefore use the parameterized posterior as the default belief state model for all PE methods described below.

### 4.3 CAV Uncertainty

When the CAV used by an attribute or IpA query is uncertain, the response probabilities used in our belief state updates (Eqs. 7, 9 or 8) are computed using expectations over CAV samples w.r.t. its belief distribution  $P_g(\phi_g | \mathbf{D}_g)$  (see Section 3.4). Similarly, we update the belief about  $u$  with responses:

$$P_U(u | \rho, \tilde{q}) := P(\phi_u | \rho, \tilde{q}) \propto P(\rho | \tilde{q}, \phi_u) P_U(u).$$

Given any query-response pair  $(\tilde{q}, \rho)$  under the current belief state  $P_U(u)$ , the posterior belief is updated by Bayes rule:

$$P(\rho | \tilde{q}, \phi_u) = \frac{P(\rho | \tilde{q}, \phi_u) P(\phi_u)}{\int_{\phi_u} P(\rho | \tilde{q}, \phi_u) P(\phi_u) d\phi_u}. \quad (10)$$

## 5 QUERY OPTIMIZATION

In this section, we develop methods to optimize the choice of queries. A key objective is to find queries whose responses quickly refine the posterior to improve recommendation quality. However, the RS may also want to use queries whose slates contain “good” recommendations w.r.t. the current posterior, not just “good” information for posterior update, since the user may select/consume one of these items at any time. We address the trade-off between information gathering and recommendation quality below.<sup>6</sup>

### 5.1 Pure Preference Elicitation

In *pure preference elicitation (PPE)*, we focus on the *information* a query response provides about user  $u$ 's utility (and ignore the predicted utility of items in  $S$ ). We consider several *acquisition functions (AFs)* to find the query with the most useful information w.r.t. reducing uncertainty in the belief state  $P_U(u)$ .<sup>7</sup>

**Random Query Selection:** This is a natural baseline that samples queries uniformly at random from  $\mathcal{Q}$ .

**Entropy:** To make posterior beliefs more informative, one can use conditional Shannon entropy [17]:

$$H(\rho | q, \mathcal{H}) = -\mathbb{E}_{\rho | q, P_U(u | \mathcal{H})} [\log P(\rho | q, P_U(u | \mathcal{H}))]. \quad (11)$$

Computing the expected posterior entropy for a given query in closed-form is generally hard, but one can estimate it by taking an expectation over all sampled posterior user responses. This metric measures the amount of latent user information that remains after a PE query  $q$ .

**Mutual Information:** We can measure the information content of  $q$  using *mutual information (MI)* between  $u$ 's target  $\phi_{I,u}^*$  and her response  $\rho$ , given  $q$  and  $\mathcal{H}$ :

$$MI(\phi_{I,u}^*; \rho | q, \mathcal{H}) := H(\rho | q, \mathcal{H}) - \mathbb{E}_{\phi_{I,u}^* | \mathcal{H}} [H(\rho | q, \phi_{I,u}^*)]. \quad (12)$$

We can also express the distribution over target items as  $P(\phi_{I,u}^* | \mathcal{H}) = \int P(\phi_{I,u}^*, \phi_u | \mathcal{H}) d\phi_u = \mathbb{E}_{\phi_u \sim P_U} [P(\phi_{I,u}^* | \phi_u)]$ .

Using (2), the second term is equal to  $\mathbb{E}_{\phi_{I,u}^* | \mathcal{H}} [H(\rho | q, \phi_{I,u}^*)] = \mathbb{E}_{\phi_u \sim P_U(u | \mathcal{H})} [H(\rho | q, \max_{i \in \mathcal{I}} \frac{\|\phi_I(i)\|_2 \cdot \phi_u}{\|\phi_u\|_2})]$ , which is

<sup>6</sup>For item queries without attributes, the optimal recommendation and query slates are identical under several natural response models, resolving this tension [45].

<sup>7</sup>An ideal AF would be a *policy* that optimizes the *sequence* of queries [9, 24]. However, such a sequential formulation is generally intractable; thus, we consider *myopic* approaches and optimize the AF w.r.t. the immediate query only (as is common in PE [11]).

estimated by sampling  $\phi_u \sim P_U(u | \mathcal{H})$ . We choose a query that minimizes this measure to make our posterior model correlated with  $u$ 's target.

**Expected Value of Information (EVOI).** The EVOI acquisition function [9, 11, 45] measures the improvement in  $u$ 's expected utility resulting from a response to query  $q$ . The EVOI of  $q$  (given  $\mathcal{H}$ ) is

$$EVOI(q | \mathcal{H}) = PEU(q | \mathcal{H}) - EU^*(P_U(u | \mathcal{H})), \quad (13)$$

where

$$EU^*(P_U(u)) = \max_{i \in \mathcal{I}} \mathbb{E}_{\phi_u \sim P_U(u)} [\phi_u^\top \phi_I(i)] \quad (14)$$

and  $PEU(q | \mathcal{H})$  is the *posterior expected utility* of  $q$ ,

$$PEU(q | \mathcal{H}) = \sum_{\rho} P(\rho | q, \mathcal{H}) \cdot EU^*(P_U(u | \mathcal{H} \cup \{(\rho, q)\})). \quad (15)$$

EVOI extracts information that offers maximum expected improvement in recommendation quality. It can be also used to decide when to stop elicitation (e.g., once it falls below some tolerance or exceeds the cost of interaction). The direct computation of PEU can be expensive, so we approximate it by sampling from the prior  $P_U(u | \mathcal{H})$ . With CAV uncertainty, the response probabilities used in PEU and other AFs are computed using expectation over CAV. The details are deferred to App. A.

**CAV Uncertainty.** With CAV uncertainty, response probabilities used in PEU and other AFs are computed using expectation over CAVs as discussed above. For example, the EVOI of a soft attribute query  $\tilde{q}$  (given  $\mathcal{H}$ ) is:

$$EVOI(\tilde{q} | \mathcal{H}) = PEU(\tilde{q} | \mathcal{H}) - EU^*(P_U(u | \mathcal{H})). \quad (16)$$

where  $EU^*(P_U(u))$  is still the same as Eq (10) but  $PEU(\tilde{q} | \mathcal{H})$  is the *posterior expected utility* of w.r.t.  $\tilde{q}$ ,

$$PEU(\tilde{q} | \mathcal{H}) = \sum_{\rho} P(\rho | \tilde{q}, \mathcal{H}) \cdot EU^*(P_U(u | \mathcal{H} \cup \{(\rho, \tilde{q})\})). \quad (17)$$

Other acquisition functions can also be extended analogously but their details will be omitted for the sake of brevity.

## 5.2 Blended Elicitation & Recommendation

In *blended PE and recommendation (BPER)*, we include the quality of the slate  $S$  used in query  $q$  when assessing  $q$ , not just its information value. BPER blends two objectives, *information gathering*  $IG(q|\mathcal{H})$  and *recommendation quality*  $RQ(q|\mathcal{H})$ .  $IG(q|\mathcal{H})$  measures the utility information extracted by  $q$  and can use any pure PE AF (e.g., Entropy, MI, EVOI).  $RQ(q|\mathcal{H})$  measures slate quality (i.e., expected utility) of  $S$ :  $RQ(q|\mathcal{H}) := \sum_{i \in S} \mathbb{E}_{\phi_u \sim P_U(u | \mathcal{H})} [\phi_u^\top \phi_I(i)]$ . To balance the two, we define the BPER AF as  $\gamma IG(q|\mathcal{H}) + (1 - \gamma) RQ(q|\mathcal{H})$ , where  $\gamma \in [0, 1]$ . We expect an RS to focus on IG earlier in an interaction sequence to learn about  $u$ , and gradually shift focus to RQ. While decaying  $\gamma$  seems natural, experiments suggest that a well-tuned *constant*  $\gamma$  suffices, since IG tends to decrease over time as the posterior converges to  $u$ 's true embedding, after which the BPER AF would focus more on recommendation.

## 5.3 Query Optimizers

Query selection requires optimizing the chosen AF over query space  $\mathcal{Q}$ . The size of  $\mathcal{Q}$  depends linearly on the number of tags  $|\mathcal{T}|$  and combinatorially on the number of items  $|I|$ , where usually  $|\mathcal{T}| \ll |I|$ . Optimal tags can be found by searching over  $\mathcal{T}$ , while slate optimization is demanding when  $I$  is large, hence requires approximation. We consider



two approaches to slate-attribute selection: optimizing slate  $S$  (or an item inside  $S$ ) and then tag  $g$ ; and optimizing both jointly.

**Thompson Sampling.** Slate  $S$  is constructed using sequential TS [20]:  $S_0 = \emptyset$  and  $S_j = S_{j-1} \cup \operatorname{argmax}_{i \in \mathcal{I} \setminus S_{j-1}} \phi_u^\top \phi_I(i)$ , where  $\phi_u \sim P_U(u | \mathcal{H})$  and  $j \in \{1, \dots, |S|\}$ . Given  $S$ ,  $g$  is randomly selected.

**Sequential Greedy.** Each item in the slate is selected greedily over the item set (excluding items that have already been added). The first item in  $S$  is the “best” item; at each subsequent step, we alternate between updating  $g$  and the next item in  $S$  using the AF until  $S$  is complete.

**Random Search.** We randomly generate a certain number of slates-attribute pairs, then apply the AF to select the query with the highest score. With CAV uncertainty, we compute score of each query by averaging over sampled CAVs.

**Continuous Relaxation.** Inspired by gradient-based PE in [44], we relax the combinatorial query selection problem into a continuous one and solve it using first- or second-order methods. Given this continuous representation of  $q$ , we project it back onto the true query space. To recover  $S$  and  $g$ , we project using Euclidean distance. The main challenge in continuous optimization lies in the discontinuous nature of our AFs, which require maximizing over the discrete item set. We circumvent this with a normalization assumption (see Appendix A.3). With CAV uncertainty, since tag  $g$  is a random variable, we have to represent  $P_g$  by a multivariate normal distribution and apply the re-parameterization trick (see Appendix A.2). We project  $P_g$  using KL-divergence.

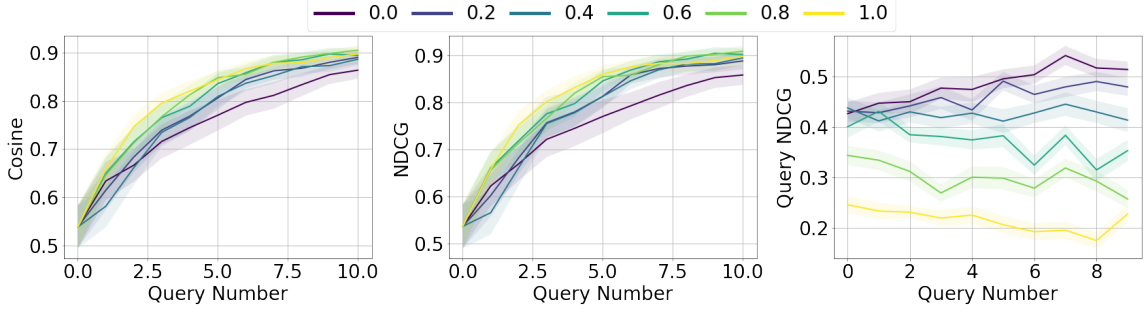
## 6 EMPIRICAL RESULTS

We conduct experiments to test the effectiveness of our PE methods, combining various query types, belief update methods, query selection schemes, and slate optimization methods. We evaluate our algorithms in three domains: (i) a simple synthetic environment; (ii) a complex simulated environment similar to that used by Göpfert et al. [21]; and (iii) one derived from the MovieLens 20M dataset [22]. The simulated domains provide us with ground-truth user utility and attribute semantics to allow precise evaluation. We describe the environments, our evaluation metrics, then a set of experimental studies in each of the three domains.

**Synthetic Environment.** We set  $|\mathcal{I}| = 1000$ ,  $|\mathcal{T}| = 10$ , with item embeddings  $\phi_I(\cdot)$  and CAV vectors  $\phi_g$  sampled from a  $d$ -dimensional ( $d = 5$ ) Gaussian  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . Each user is represented by a Gaussian distribution  $P_U$  with a random mean vector and covariance matrix. User response noise is  $\sigma_g = 0.1$  for all  $g \in \mathcal{T}$ .

**RecSim NG Environment.** Following Göpfert et al. [21], we use the RecSim NG [33] environment to construct a user model to generate ratings and tags, and then use the resulting dataset for learning CAVs and for PE. With  $|\mathcal{U}| = 25,000$  users and  $|\mathcal{I}| = 10,000$  items, we represent each user and item in a  $d$ -dimensional embedding space ( $d = 25$ )—each item dimension reflects a latent “attribute” and each user dimension a utility for that item attribute. User ratings for items are generated by a staged sampling process which approximates the user/item-embedding dot product (plus noise and rating discretization). The number of ratings given by users follows a power law distribution to ensure ratings sparsity. Five of the 25 latent attributes are taggable ( $|\mathcal{T}| = 5$ ). Users can only tag rated items and are more likely to tag higher-rated items. For any tag  $g$ , a *fixed* threshold  $\tau_g$  gives the probability of  $u$  tagging the item; we set  $\tau_g = 0.5$  for soft attributes. We train CAVs for each tag with logistic regression using the item-tag data. Average CAV quality (accuracy) is 0.909 on the test set, and Spearman correlation between predicted and ground-truth tags is 0.570. Noise is  $\sigma_g = 0.25$  for all  $g \in \mathcal{T}$ . The temperature  $T$  in the item response model is 0.5.

**MovieLens 20M.** We also evaluate our methods on the more realistic MovieLens 20M dataset [22]. There are 465K tag-instances in which 138K users applied tags to 27K movies. Tags mainly represent movie genres (e.g., action, drama) or more subjective descriptions (e.g., quirky, funny). Following the methodology of [21], we split ratings and tag data

Fig. 1. IG, RQ ( $\gamma$ ) tradeoff, BPER scheme (Synthetic).

into train and test sets such that any user-item pair is present exactly in one of these sets. We generate  $d$ -dimensional ( $d = 50$ ) user and item embeddings using alternating least-squares (ALS) and train CAVs on this latent space. Due to item-tag sparsity, we train CAVs only for the 164 most-frequently used tags (w.r.t. unique users, items). Average CAV test quality is 0.727. User response noise is again  $\sigma_g = 0.25$  and the temperature in the item response model is  $T = 0.5$ .

**Metrics.** We use three metrics to evaluate the performance of our PE methods. (i) **Cosine** is the cosine similarity (or alignment [39]) between the mean user posterior embedding and true user embedding. After  $k$  queries, it is given by 
$$\text{Cosine} = \frac{\phi_u^\top \mathbb{E}_{\phi \sim P_U(\cdot | \mathcal{H}^{(k)})}[\phi]}{\|\phi_u\|_2 \cdot \|\mathbb{E}_{\phi \sim P_U(\cdot | \mathcal{H}^{(k)})}[\phi]\|_2}$$
. (ii) **NDCG** is the normalized discounted cumulative gain [27] between the true top  $|S|$  items and the top  $|S|$  items estimated using the posterior. (iii) **Query NDCG** is a variant of NDCG where the slate  $S^{(k)}$  that is *presented* to the user in the  $k$ 'th query replaces the *a posteriori* optimum. We use this measure primarily with the BPER scheme to study the trade off between IG and RQ.

**Implementation Details.** We implement our PE algorithms using Tensorflow [1] which offers automatic differentiation for both Laplace approximation and continuous relaxation. We use Tensorflow Probability [18] for HMC in parameterized posterior updates and for probability/likelihood computation.

**Experiment 1: IG-RQ Tradeoff with Synthetic Data.** We use the synthetic environment to assess the IG-RQ tradeoff and its effect on PE performance. We initialize the RS with the prior user embedding distribution, after which it queries a simulated user 10 times with a slate of size 5 (and tag if needed). We test each PE method with 10 users, 5 runs each (different random seeds). We report the mean and standard deviation of our metrics over these 50 runs. Figure 1 shows the trade off between IG and RQ in the BPER scheme. Using IpA queries, EVOI as our AF, and random-search slate optimization, we run PE with  $\gamma$  ranging from 1 (pure elicitation) to 0 (pure recommendation). While tuning  $\gamma$  offers only a modest improvement in elicitation quality (see cosine and NDCG), PE with smaller  $\gamma$  generally places higher quality items on the slate during elicitation, with quality improving with the number of queries (see Query NDCG). The nonlinear trade-off between IG and RQ induces a “sweet spot” at around  $\gamma = 0.5$ , and therefore we use this  $\gamma$  for all subsequent experiments.

**Experiment 2: Posterior Update Methods with RecSim NG.** In this experiment we compare the performance of the Gaussian posterior, updated via the closed-form Laplace approximation, with the parameterized (categorical) posterior, updated using HMC as specified in Section 4.2. Figure 2 presents the performance of different posterior update methods in attribute-based PE, tested in the Recsim NG environment. We compare the accuracy of belief states via cosine similarity between the current posterior embedding and the underlying ground-truth one. Clearly, parameterized posterior models outperform the Gaussian counterpart in both attribute and IpA response models.

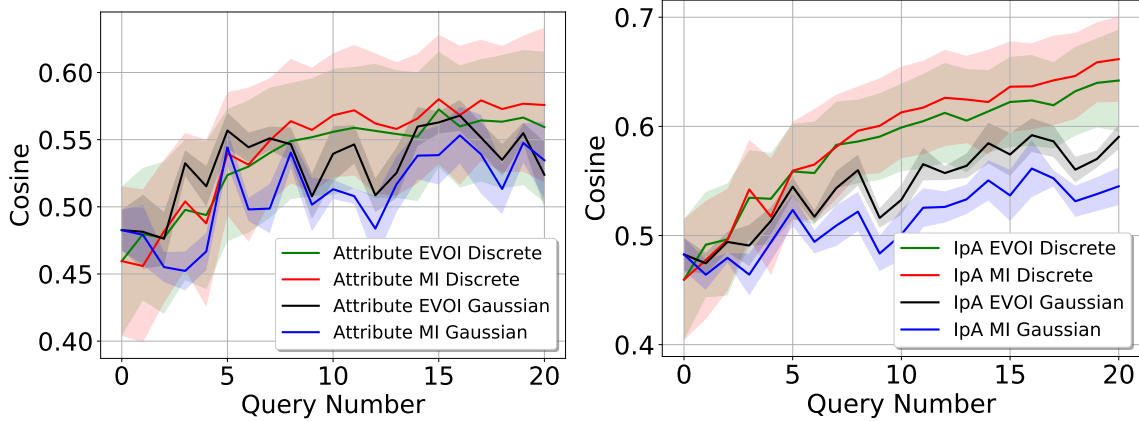


Fig. 2. Cosine similarity on RecSim NG with Gaussian (left) and parameterized (right) posterior.

**Experiment 3: Ablation Studies with RecSim NG.** Using the RecSim NG environment, we set the RS prior over user embeddings to be Gaussian (Section 2). The RS queries each user 20 times using slates of size 5 (for each query type). We test each PE method with 16 random users over 5 runs. We run several ablation studies to assess how different user query responses, query/slate optimizers, and AFs affect information gathering (IG) and recommendation quality (RQ). We focus on the cosine and NDCG metrics (as Query NDCG mainly measures the IG-RQ trade-off). To focus on the CAV/query-type interaction, our first four studies assume no CAV uncertainty. Figure 3 compares the different query types and AFs, while fixing the query optimization method to be random search over 100 queries. Among the query types, PE with IpA performs the best, followed by item then attribute queries. This is unsurprising since IpA elicits the most information. While attribute queries provide the simplest user feedback, PE with attribute queries is quite effective initially, achieving similar performance to item and IpA-based PE; though with only 5 taggable attributes, performance quickly saturates with more queries. EVOI is the most effective AF, outperforming MI and Entropy, and generating higher quality recommendations at each stage of the interaction (number of queries). This reflects the fact that EVOI-maximizing queries elicit user utility information directly aimed at improving recommendation quality.

In Figure 4, we assess the impact of various query optimizers on PE and recommendation quality. We use IpA queries, the EVOI AF, and BPER with  $\gamma = 0.5$ . We consider five query optimizers: (i) random search; (ii) continuous relaxation with first-order optimization; (iii) continuous relaxation with second-order optimization; (iv) sequential greedy (myopic w.r.t. BPER AF); and (v) Thompson sampling (TS). We also use a baseline in which both the slate and tag are selected uniformly at random. Random requires the least computation, followed by sequential greedy, TS, and then the joint optimizers on slates and tags. The additional computation cost of random search and continuous relaxation offers significant gains in IG and RQ, with a 10–15% NDCG improvement over TS and sequential greedy (though TS and sequential greedy perform well early in the PE process). While joint-optimization-based methods still outperform random, random fares better than the greedy methods, TS, and sequential greedy, which is surprising given its simplicity. We conjecture this is because the RecSim NG environment is quite simple, and the inherent item diversity on random slates itself helps reduce belief state entropy.

In Figure 5, we compare the batch and iterative posterior update methods (Section 4.1.1). In both RecSim NG and MovieLens datasets, the iterative method outperforms batch across multiple optimization methods. Intuitively, while

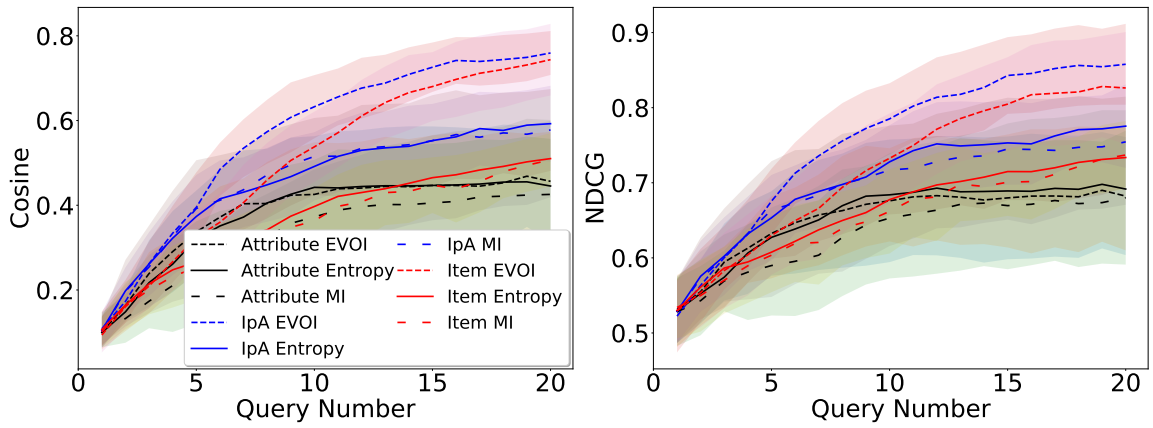


Fig. 3. Comparing query types and AFs (RecSim NG).

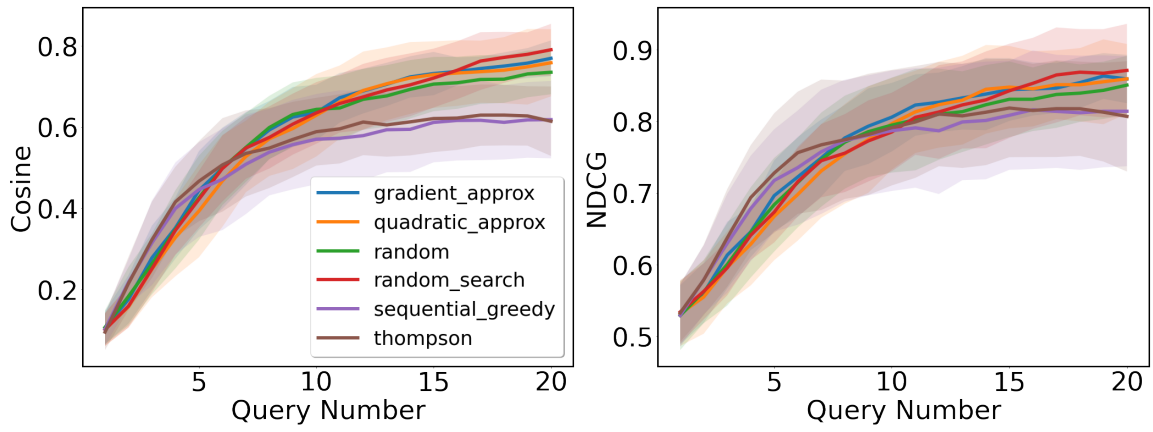


Fig. 4. Comparing query optimizers (RecSim NG).

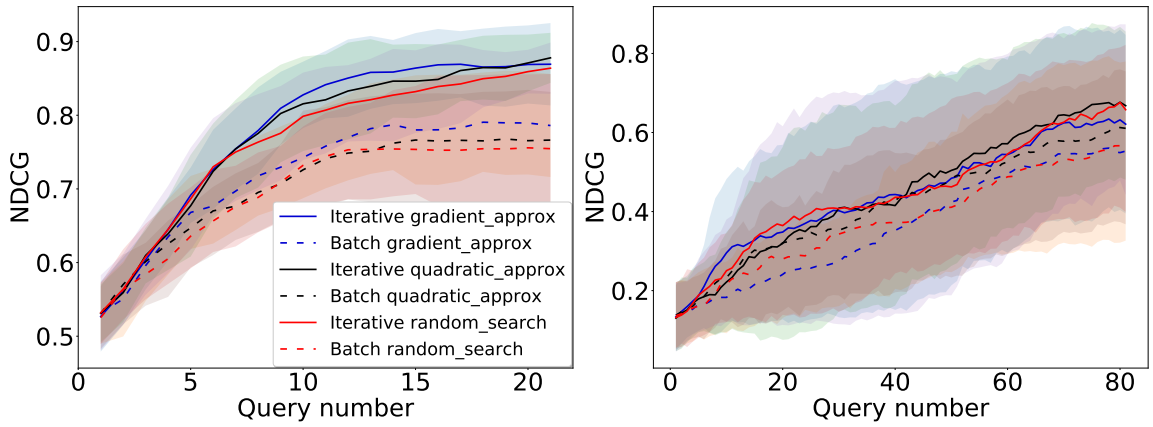


Fig. 5. Batch vs. iterative posterior update. Left: RecSim NG. Right: Movielens

information from both the prior  $P_U(u)$  and the query  $q$  contribute to the posterior update, only the responses to  $q$  provide new information. When the prior is far from the true user utility, especially at the start of training, the posterior update is dominated by information from  $q$ ; thus, by updating the posterior iteratively, new information collected from each query response (in HMC generation of new posterior samples) compounds. As a result, the iterative method generally has much better sample efficiency. Since user queries are expensive, we adopt the iterative method in all subsequent experiments.

We next study the effect of using our CAV-based semantics on the performance of the PE algorithms. Our PE framework does not require using CAVs for soft attributes—it can work with any semantics discovery method that generates some form of  $g$ -score to quantify the degree to which an item  $i$  satisfies a tag  $g$ . We compare *PITF* (*pairwise interaction tensor factorization*) [38] as an alternative semantics. PITF is a tensor factorization method, originally developed for personalized tag prediction, which outputs a predicted tag  $y_{u,i,g}$  for each user  $u \in \mathcal{U}$ , item  $i \in \mathcal{I}$ , and tag  $g \in \mathcal{T}$ . To learn the PITF semantics,  $\phi_g$ , for a tag  $g$ , we fit a linear regressor to the dataset  $\{(\phi_I(i), y_{u,i,g})\}_{u,i}$ , where  $u$  and  $i$  are sampled from  $\mathcal{U}$  and  $\mathcal{I}$ .<sup>8</sup> In Figure 6, we compare the performance of PE algorithms that use the CAV and PITF representations with attribute and IpA queries, random search, and the EVOI AF. In all regimes, PE with CAVs performs better than with PITF, with a more significant advantage in cosine similarity (which reflects better ability to estimate the user’s utility). This corroborates the main motivation for using CAVs, which is the ability to better represent semantic attributes that are especially predictive of users’ preferences.

Finally, we consider the impact of CAV uncertainty on our PE algorithms. In contrast to the experiments above, we model the noise in CAV discovery by injecting various degrees of uncertainty into the CAV model  $P_g(\phi_g | \mathbf{D}_g)$  for each tag  $g$ , where the more certain tags have lower (co-)variance in their CAV models. We assume  $P_g$  is a multivariate normal distribution with mean  $\mu_g$  and covariance matrix  $\Sigma_g$ . To explicitly model the fact that CAVs will have various degrees of uncertainty, the mean of  $P_g$  is the (deterministic) CAV learned via logistic regression, while the co-variance matrix is  $\sigma_g^2 \mathcal{I}$ , where the standard deviation  $\sigma_g$  is randomly chosen from a set of  $|\mathcal{T}|$  values, ranging from 0.01 to 1 evenly spread on a  $\log_{10}$  scale. For queries using attribute  $g$ , a “true” CAV vector  $\phi_g$  is sampled from  $P_g$  to power the user’s responses (Eqs. 3, 4 and 6).

We test whether modeling the CAV uncertainty in belief update and query optimization improves IG and RQ. Figure 7 shows PE results of an experiment using IpA queries and three joint optimizers (random search, continuous relaxation with first- or second-order optimization). We see that modeling CAV uncertainty offers significant gain in IG and RQ, with a 10-15% NDCG improvement over PE methods that update their beliefs by treating the mean CAV as “certain”. Ignoring uncertainty (and CAV learning error) can generate over-confident error-prone belief updates.

**Experiment 4: MovieLens 20M.** To evaluate PE with MovieLens 20M data, similar to the critiquing setup in [21], we create “ground-truth” users, each of whom has rated at least 50 movies. Given the large number of ratings, their corresponding embeddings are relatively stable and accurate, so we treat them as the *ground truth utility* for these users, which is used to generate query responses in our experiments. We sample 16 such test users for PE. The RS belief state for each user is initialized with a *cold-start prior* whose mean is the average embedding over all users. We use slates of 5 movies, set  $\gamma$  to 0.5, and use EVOI AF and parameterized posteriors.

We first evaluate the impact of CAV uncertainty. Similar to the RecSim NG experiment above, for each attribute  $g$  we assume the RS has a known multivariate-normal distribution  $P_g$  with mean  $\mu_g$  and covariance  $\sigma_g^2 \mathcal{I}$  that are generated as above to capture CAV uncertainty. We test how modeling the CAV uncertainty in belief update and query

<sup>8</sup>We use <https://github.com/yamaguchiyuto/pitf/> to train the PITF model.

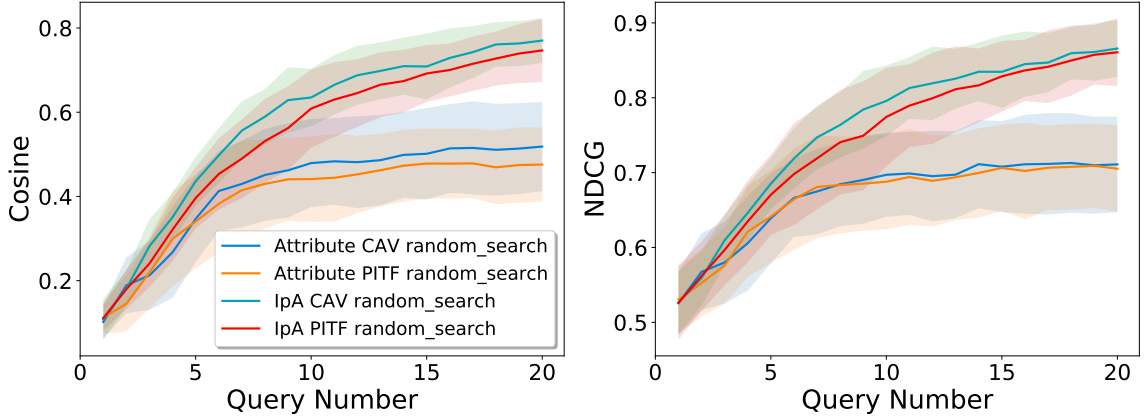


Fig. 6. CAV vs. PITF attribute semantics (RecSim NG).

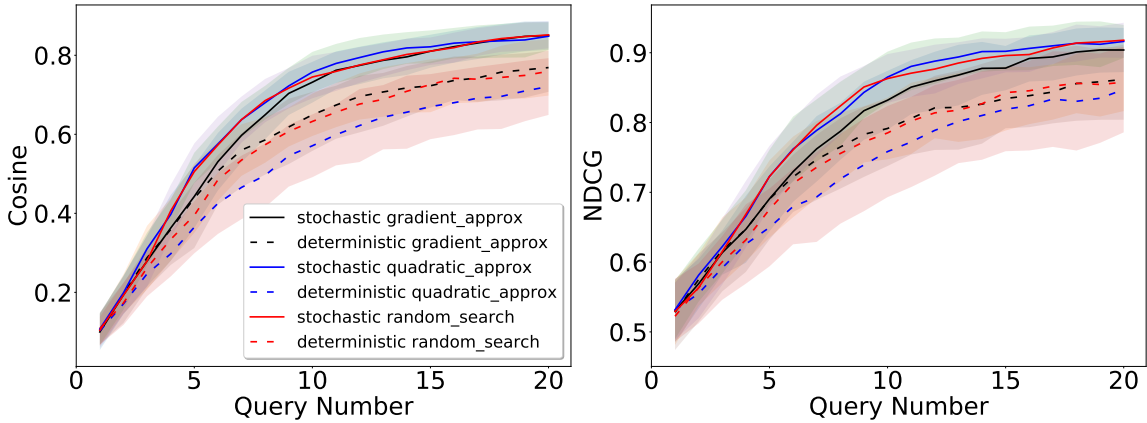


Fig. 7. PE with CAV uncertainty modeling (RecSim NG).

optimization impacts IG and RQ. Figure 8 shows PE results using IpA queries and three joint optimizers. Again, we see that modeling CAV uncertainty in PE improves both IG and RQ, with an up to 10–20% improvement in NDCG compared to PE using “deterministic” CAVs. This demonstrates the need to model noise in the estimation of the CAV semantics for soft attributes in PE, particularly in real-world domains as reflected in MovieLens.

We next validate the performance of different query optimizers in the MovieLens setting. Figure 9 compares different optimizers w.r.t. IG and RQ. While the RS can learn about user preferences and improve its recommendations with most PE methods, the joint optimizers provide better recommendations (slates with greater NDCG across any number of queries) by extracting more useful information (greater cosine). As in RecSim NG, PE with random queries performs relatively well, likely due to two factors: (i) With an inaccurate prior model (e.g., during the early phases of PE), a random query of diverse items may effectively reduce belief-state entropy; (ii) With a combinatorial space of decision variables (slates), none of the optimization strategies converge to local optima (w.r.t. BPER) that are much better than the

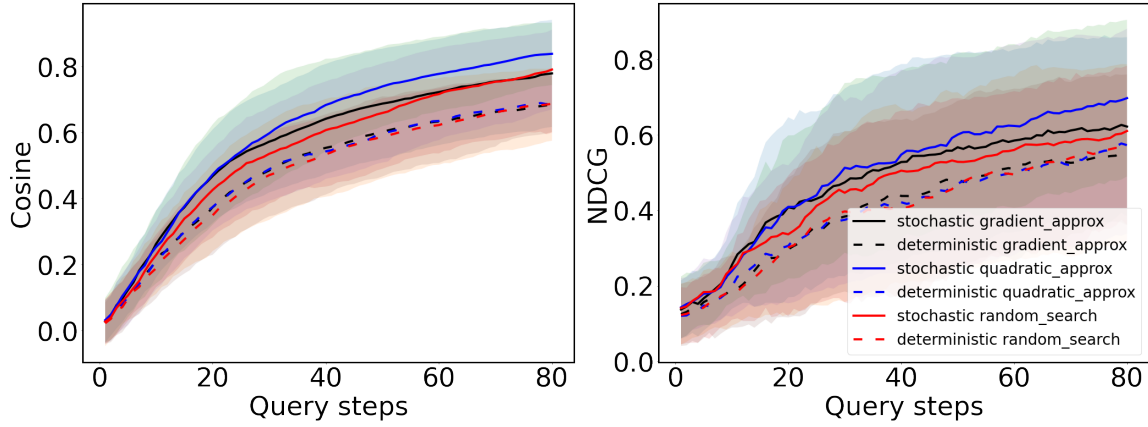


Fig. 8. PE with CAV uncertainty modeling (MovieLens).

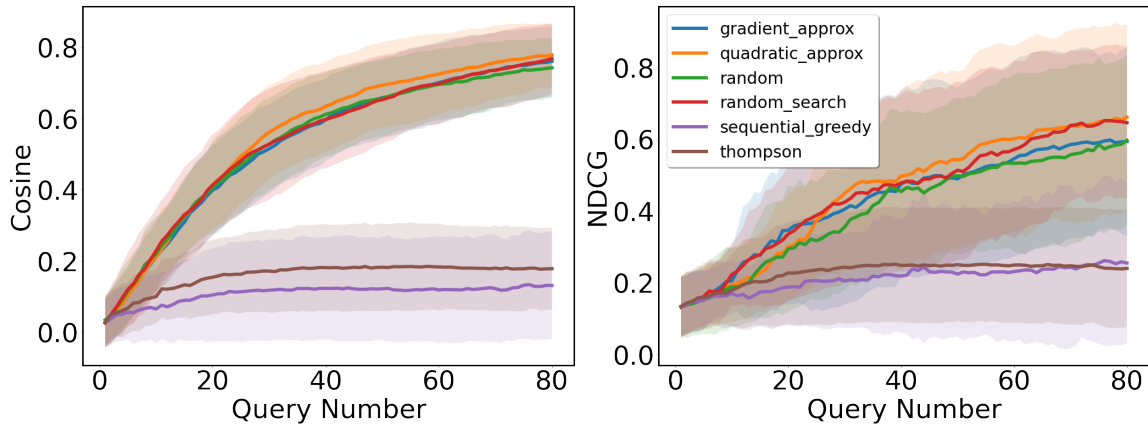


Fig. 9. Comparing query optimizers (MovieLens).

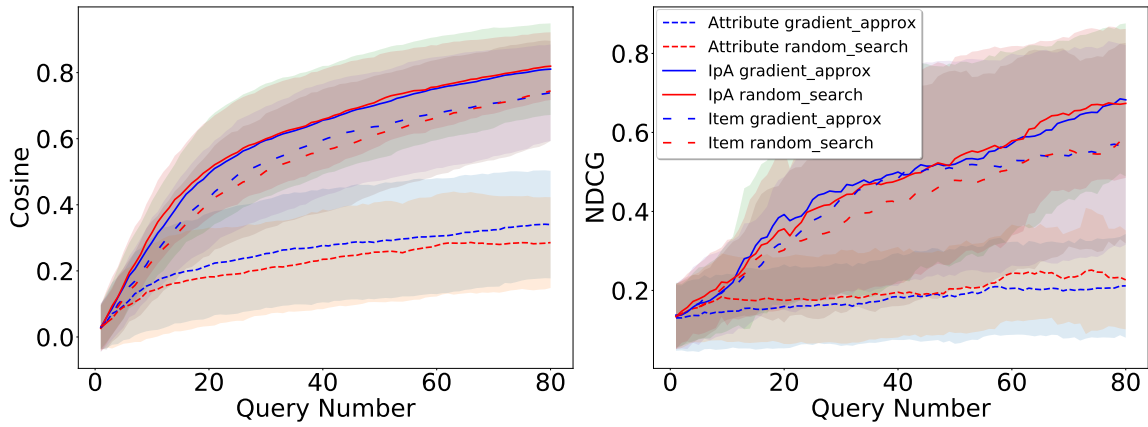


Fig. 10. Comparing query types (MovieLens).

random query. This suggests that smartly switching between random and jointly optimized queries may be “practically optimal” in terms of computational complexity and PE, but detailed studies are left for future work.

Finally, we explore different query types with both random search and gradient approximation. Figure 10 shows that PE with IpA queries is much more effective than PE with simpler queries, an observation consistent with that seen in the RecSim NG setting. With the additional information collected at each PE step, IpA queries offer a 10–20% NDCG gain over item queries. PE with attribute queries performs worst w.r.t. both IG and RQ because, without allowing users to express nuanced preferences through item selection, it may fail to capture preferences regarding important but untaggable latent attributes.

## 7 CONCLUSIONS AND FUTURE WORK

We have proposed a framework for preference elicitation in interactive recommenders that allows users to navigate item space or critique recommendations using soft attributes. Our techniques exploit concept-activation vectors (CAVs) to uncover the semantics of soft attributes w.r.t. the RS’s item representation, and to support Bayesian updates of the RS’s beliefs about a user’s underlying preferences. We considered uncertainty in the semantics of soft attributes and develop scalable, continuous relaxation methods for query optimization. We studied a variety of query types, both item and attribute-based, and response models, developed belief update and query optimization methods (i.e., acquisition functions) for each, and demonstrated the efficacy of these methods on both simulated and MovieLens 20M data. Our work has fundamental implications for designing the next-generation of recommenders (e.g., dialogue-based RSs), since our methods allows RSs to interact and learn about users more naturally with soft attributes.

There are a number of avenues for future research. With insufficient tag data, CAV noise may render PE ineffective, requiring active learning methods for attribute semantics—integrating these into the PE process should prove valuable. *Subjectivity*, where users may use terms in different ways [21, 36], means methods to elicit a user’s “personalized” semantics for an attribute should further enhance PE. Multi-modal user belief model, reflecting more diverse preferences, are of interest. While our current PE methods optimize for myopic information gathering and recommendation, multi-step lookahead PE methods that use reinforcement learning remain an important challenge. Finally, experiments with real users are needed to further evaluate our methods with more diverse user preferences, attribute semantics, and responses.

## REFERENCES

- [1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. 2016. TensorFlow: A System for Large-scale Machine Learning. *arXiv preprint arXiv:1605.08695* (2016).
- [2] Loïc Adam and Sébastien Destercke. 2021. Possibilistic preference elicitation by minimax regret. In *Proceedings of the Thirty-seventh Conference on Uncertainty in Artificial Intelligence (UAI-21) (Proceedings of Machine Learning Research, Vol. 161)*. AUAI Press, 718–727.
- [3] G. Adomavicius and A. Tuzhilin. 2005. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005).
- [4] K. Balog, F. Radlinski, and A. Karatzoglou. 2021. On Interpretation and Measurement of Soft Attributes for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [5] M. Ben-Akiva, S. Lerman, S. Lerman, et al. 1985. *Discrete choice analysis: theory and application to travel demand*. Vol. 9. MIT press.
- [6] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, and E. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 46–54.
- [7] E. Biyik, N. Huynh, M. Kochenderfer, and D. Sadigh. 2020. Active Preference-Based Gaussian Process Regression for Reward Learning. In *Proceedings of Robotics: Science and Systems*.
- [8] Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard. 2019. Incremental Elicitation of Rank-Dependent Aggregation Functions based on Bayesian Linear Regression. In *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. Macau, China, 2023–2029.
- [9] C. Boutilier. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of the 18th National Conference on Artificial Intelligence*. 239–246.



- [10] C. Boutilier, R. Zemel, and B. Marlin. 2003. Active Collaborative Filtering. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*. 98–106.
- [11] U. Chajewska, D. Koller, and R. Parr. 2000. Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*. 363–369.
- [12] B. Chaptini. 2005. *Use of discrete choice models with recommender systems*. Ph. D. Dissertation. MIT.
- [13] L. Chen and P. Pu. 2004. *Survey of Preference Elicitation Methods*. Tech Report. EPFL.
- [14] L. Chen and P. Pu. 2012. Critiquing-based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 125–150.
- [15] S. Chib and E. Greenberg. 1995. Understanding the Metropolis-Hastings algorithm. *The American statistician* 49, 4 (1995), 327–335.
- [16] D. Cohen, M. Aharon, Y. Koren, O. Somekh, and R. Nissim. 2017. Expediting Exploration by Attribute-to-feature Mapping for Cold-start Recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 184–192.
- [17] T. Cover. 1999. *Elements of information theory*. John Wiley & Sons.
- [18] J. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. Saurous. 2017. TensorFlow Distributions. (2017). <http://arxiv.org/abs/1711.10604>
- [19] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. 2010. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *2010 IEEE International Conference on Data Mining (ICDM-10)*. 176–185.
- [20] A. Gopalan, S. Mannor, and Y. Mansour. 2014. Thompson sampling for complex online problems. In *International conference on machine learning*. PMLR, 100–108.
- [21] C. Göpfert, Y. Chow, C. Hsu, I. Vendrov, T. Lu, D. Ramachandran, and C. Boutilier. 2022. Discovering Personalized Semantics for Soft Attributes in Recommender Systems Using Concept Activation Vectors. In *Proceedings of the Web Conference*. (Also arXiv 2202.02830).
- [22] F. Harper and J. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2016), 19:1–19:19.
- [23] C. He, D. Parra, and K. Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27.
- [24] H. Holloway and C. White. 2003. Question Selection for Multiattribute Decision-aiding. *European Journal of Operational Research* 148 (2003), 525–543.
- [25] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th International Conference on Data Mining*. 263–272.
- [26] T. Jaakkola and M. Jordan. 1997. A Variational Approach to Bayesian Logistic Regression Models and their Extensions. In *6th International Workshop on Artificial Intelligence and Statistics (AISTATS=97)*. 283–294.
- [27] K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.
- [28] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*. 2668–2677.
- [29] J. Kuo, P. Cheng, and H. Wang. 2009. Learning to Rank from Bayesian Decision Inference. In *18th ACM Conference on Information and Knowledge Management (CIKM-09)*. 827–836.
- [30] K. Li, M. Tucker, E. Biyik, E. Novoseller, J. Burdick, Y. Sui, D. Sadigh, Y. Yue, and A. Ames. 2021. ROIAL: Region of Interest Active Learning for Characterizing Exoskeleton Gait Preference Landscapes. In *International Conference on Robotics and Automation*.
- [31] K. Luo, S. Sanner, G. Wu, H. Li, and H. Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference*. 2535–2541.
- [32] J. McAuley, J. Leskovec, and D. Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *12th International Conference on Data Mining (ICDM-12)*. 1020–1025.
- [33] M. Mladenov, C. Hsu, V. Jain, E. Ie, C. Colby, N. Mayoraz, H. Pham, D. Tran, I. Vendrov, and C. Boutilier. 2020. Demonstrating Principled Uncertainty Modeling for Recommender Ecosystems with RecSim NG. In *Fourteenth ACM Conference on Recommender Systems*. 591–593.
- [34] P. Nema, A. Karatzoglou, and F. Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with  $\beta$ -VAE. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1356–1365.
- [35] F. Radlinski, K. Balog, B. Byrne, and K. Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *Proceedings of the Annual SIGDial Meeting on Discourse and Dialogue*.
- [36] Filip Radlinski, Craig Boutilier, Deepak Ramachandran, and Ivan Vendrov. 2022. Subjective Attributes in Conversational Recommendation Systems: Challenges and Opportunities. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-22)*. Washington, DC, 12287–12293.
- [37] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence (UAI-09)*. Montreal, 452–461.
- [38] S. Rendle and L. Schmidt-Thieme. 2010. Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. 81–90.
- [39] D. Sadigh, A. Dragan, S. Sastry, and S. Seshia. 2017. Active Preference-Based Learning of Reward Functions. In *Proceedings of Robotics: Science and Systems*.
- [40] R. Salakhutdinov and A. Mnih. 2007. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems* 20. 1257–1264.

- [41] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*. 285–295.
- [42] M. Sundararajan, A. Taly, and Q. Yan. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*. 3319–3328.
- [43] K. Train. 2009. *Discrete choice methods with simulation*. Cambridge university press.
- [44] I. Vendrov, T. Lu, Q. Huang, and C. Boutilier. 2020. Gradient-based optimization for Bayesian preference elicitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10292–10301.
- [45] P. Viappiani and C. Boutilier. 2010. Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets. In *Advances in Neural Information Processing Systems 23*. 2352–2360.
- [46] P. Viappiani, B. Faltings, and P. Pu. 2006. Preference-based Search using Example-Critiquing with Suggestions. *Journal of Artificial Intelligence Research* 27 (2006), 465–503.
- [47] C. Williams and C. Rasmussen. 2006. *Gaussian processes for machine learning*. MIT press.
- [48] G. Wu, K. Luo, S. Sanner, and H. Soh. 2019. Deep language-based critiquing for recommender systems. In *Proceedings of the Thirteenth ACM Conference on Recommender Systems*. 137–145.
- [49] J. Yang, X. Yi, D. Cheng, L. Hong, Y. Li, S. Wang, T. Xu, and E. Chi. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. In *Proceedings of the Web Conference*. 441–447.
- [50] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, and E. Chi. 2019. Sampling-bias-corrected Neural Modeling for Large Corpus Item Recommendations. In *Proceedings of the Thirteenth ACM Conference on Recommender Systems*. 269–277.
- [51] X. Zhao, W. Zhang, and J. Wang. 2013. Interactive Collaborative Filtering. In *Proceedings of the Twenty-Second ACM International Conference on Information and Knowledge Management*. 1411–1420.

## A ELICITATION WITH CAV UNCERTAINTY

### A.1 User Belief State Update and AFs

To incorporate the CAV uncertainty, we consider a soft attribute query  $\tilde{q} = (S, P_g)$ , where RS has a CAV belief  $P_g(\phi_g | \mathbf{D}_g)$  reflecting this uncertainty with training data  $\mathbf{D}_g$ . We assume the true CAV  $\phi_g$  is sampled from  $P_g$ . In this work, we do not update the CAV belief with user responses to CAV semantic elicitation. However, the RS has to compute the response probabilities for updating user belief state according to its CAV belief.

For a (soft) attribute query  $\tilde{q} = (S, P_g)$ , by referring to the user response model  $P(\rho | q, \phi_u)$  we define

$$\begin{aligned} P(\rho | \tilde{q}, \phi_u) &= \mathbb{E}_{\phi_g \sim P_g} P(\rho | (S, \phi_g), \phi_u) \\ &= \int_{\phi_g} P_g(\phi_g | \mathbf{D}_g) P(\rho | (S, \phi_g), \phi_u) d\phi_g \end{aligned} \quad (18)$$

as the probability of observing response  $\rho$  under a CAV belief  $P_g$ . We then update our belief about  $u$  with responses:

$$P_U(u | \rho, \tilde{q}) := P(\phi_u | \rho, \tilde{q}) \propto P(\rho | \tilde{q}, \phi_u) P_U(u).$$

Given any query-response pair  $(\tilde{q}, \rho)$  under the current belief state  $P_U(u)$ , the posterior belief is updated by Bayes rule:

$$P(\rho | \tilde{q}, \phi_u) = \frac{P(\rho | \tilde{q}, \phi_u) P(\phi_u)}{\int_{\phi_u} P(\rho | \tilde{q}, \phi_u) P(\phi_u) d\phi_u}. \quad (19)$$

We also need to consider the CAV uncertainty in computing acquisition functions. For example, the EVOI of  $\tilde{q}$  (given  $\mathcal{H}$ ) is:

$$EVOI(\tilde{q} | \mathcal{H}) = PEU(\tilde{q} | \mathcal{H}) - EU^*(P_U(u | \mathcal{H})). \quad (20)$$

where  $EU^*(P_U(u))$  is still the same as Eq (14) but  $PEU(\tilde{q} | \mathcal{H})$  is the *posterior expected utility* of w.r.t.  $\tilde{q}$ ,

$$PEU(\tilde{q} | \mathcal{H}) = \sum_{\rho} P(\rho | \tilde{q}, \mathcal{H}) \cdot EU^*(P_U(u | \mathcal{H} \cup \{(\rho, \tilde{q})\})). \quad (21)$$

Other acquisition functions can also be extended analogously but their details will be omitted for the sake of brevity.

### A.2 Continuous Relaxation for AFs

Recall that the acquisition function can be expressed as

$$F(q) = \gamma IG(q | \mathcal{H}) + (1 - \gamma) RQ(q | \mathcal{H}), \quad (22)$$

where

$$RQ(q | \mathcal{H}) := \sum_{i \in S} \mathbb{E}_{\phi_u \sim P_U(u | \mathcal{H})} [\phi_u^\top \phi_I(i)] \quad (23)$$

measures the recommendation quality and  $IG$  is the information gathering metric (i.e., Entropy, MI or EVOI). In the following we take EVOI as an example and show how we obtain an differentiable objective function. The same derivation applies to Entropy and MI. For  $IG=EVOI$ , computing  $q^* = \arg \max_q F(q)$  is equivalent to finding

$$q^* = \arg \max_q \{ \gamma PEU(q | \mathcal{H}) + (1 - \gamma) RQ(q | \mathcal{H}) \}. \quad (24)$$

PEU requires computing the expected belief w.r.t. possible responses  $\rho$ ,  $P_U(u | \mathcal{H} \cup \{(\rho, q)\})$ . This computation can be expensive, so we approximate PEU by sampling from the prior  $P_U(u | \mathcal{H})$ :

$$PEU'(q | \mathcal{H}) := \sum_{\rho} \max_{i \in \mathcal{I}} \mathbb{E}_{\phi_u \sim P_U(u | \mathcal{H})} [\phi_u^\top \phi_I(i) \cdot P(\rho | q, \phi_u)]. \quad (25)$$

With CAV uncertainty, the response probabilities used in PEU and other AFs are computed using expectation over CAV. The PEU function in Eq (25) can be further approximated by drawing  $m$  samples  $\{\phi_{u,j}\}_{j=1}^m$  from the posterior  $P_U(u | \mathcal{H})$ :

$$PEU(q | \mathcal{H}) \approx \frac{1}{m} \sum_{\rho} \max_{i \in \mathcal{I}} \left[ \sum_{j=1}^m \phi_{u,j}^\top \phi_I(i) P(\rho | q, \phi_{u,j}) \right]. \quad (26)$$

And similarly, the RQ function can be approximated by

$$RQ(q | \mathcal{H}) \approx \frac{1}{m} \sum_{i \in \mathcal{S}} \left[ \sum_{j=1}^m \phi_{u,j}^\top \phi_I(i) \right]. \quad (27)$$

In the following we use the concatenation of attribute vector  $g$  and item embeddings in  $S$  to represent  $q$ , immediately  $RQ(q | \mathcal{H})$  is differentiable w.r.t.  $q$ . However, computing  $PEU(q | \mathcal{H})$  requires enumerating over  $\mathcal{I}$  and thus is not differentiable. To derive a differentiable objective function  $F(q)$ , we assume  $\mathcal{I}$  is rich enough such that for each query-response pair  $(q, \rho)$  we can find  $i_{(q,\rho)}^* = \frac{\sum_{j=1}^m \phi_{u,j} P(\rho | q, \phi_{u,j})}{\|\sum_{j=1}^m \phi_{u,j} P(\rho | q, \phi_{u,j})\|} \max_{i \in \mathcal{I}} \|\phi_I(i)\|$  to maximize the RHS of Eq (26). Then Eq (26) can be further approximated by

$$PEU(q | \mathcal{H}) \approx \frac{\max_{i \in \mathcal{I}} \|\phi_I(i)\|}{m} \sum_{\rho} \left\| \sum_{j=1}^m \phi_{u,j} P(\rho | q, \phi_{u,j}) \right\|_2, \quad (28)$$

which is differentiable w.r.t.  $q$  because the response model  $P(\cdot | q, \cdot)$  is differentiable w.r.t.  $q$ .

Now we extend the continuous relaxation framework to account for queries with an uncertain CAV vector as its attribute. We may apply the re-parameterization trick to derive a continuous acquisition function. Suppose the CAV vector  $\phi_g$  in the attribute query  $\tilde{q} = (S, P_g)$  follows a Multivariate normal  $\mathcal{N}(\mu_g, \Sigma_g)$ . Then we can sample  $n$  queries  $\{q_i | q_i = (\{\phi_I(i)\}_{i \in S}, \phi_{g,i})\}_{i=1}^n$  by calculating  $\phi_{g,i} = \mu_g + L_g \epsilon_i$ , where  $\Sigma_g = L_g^\top L_g$  is the Cholesky decomposition of the covariance matrix, and  $\epsilon_i \sim \mathcal{N}(0, \mathcal{I}_d)$  are i.i.d. standard normal random vectors. Based on Eq (26), we can now introduce the uncertainty of CAVs in the following approximation of PEU

$$PEU(\tilde{q} | \mathcal{H}) \approx \frac{\max_{i \in \mathcal{I}} \|\phi_I(i)\|}{mn} \sum_{i=1}^n \sum_{\rho} \left\| \sum_{j=1}^m \phi_{u,j} P(\rho | q_i, \phi_{u,j}) \right\|_2,$$

where

$$\begin{aligned} P(\rho = +1 | q_i, \phi_u) &= P(\rho = +1 | S, \mu_g, \Sigma_g, \epsilon_i, \phi_u) \\ &= \Phi\left(\frac{(\mu_g + L_g \epsilon_i)^\top (\phi_{I,u}^* - \phi_{I,\bar{S}})}{\sigma_g}\right), \end{aligned} \quad (29)$$

Eq. (29) is differentiable w.r.t.  $q = (\{\phi_I(i)\}_{i \in S}, \mu_g, L_g)$  and we can thus obtain its gradient. Suppose  $(\hat{\phi}_{i \in S}, \hat{\mu}_g, \hat{L}_g)$  is a solution of the corresponding continuous optimization problem. The projection is done by picking the attribute with the smallest KL-divergence to  $\mathcal{N}(\hat{\mu}_g, \hat{L}_g^\top \hat{L}_g)$  and the slate of items with the minimum Euclidean distance to  $\hat{\phi}_{i \in S}$ .

### A.3 Gradient-based Query Optimization

Once we approximate the acquisition function in a continuous form  $F(q)$ , standard gradient-based method can be applied to find the optimal query  $q^* = \arg \max_q F(q)$ . In our empirical study we use both first and second-order optimization. First we generate an initial query  $q_0$  from random search with 20 trials and then apply either the following updates for a fixed number of steps:

(1) First-order optimization:

$$q_t = q_{t-1} + \eta \nabla F(q_{t-1}), 1 \leq t \leq T,$$

(2) Second-order optimization:

$$q_t = q_{t-1} + [\lambda I + \nabla^2 F(q_{t-1})]^{-1} \nabla F(q_{t-1}), 1 \leq t \leq T,$$

where  $\eta$  is the learning rate,  $\lambda$  is a regularization constant used to guarantee the non-singularity of the Hessian matrix  $\nabla^2 F$ . In our experiments we take  $T = 2, \eta = 1e - 3, \lambda = 1e - 4$ .

## B GRADIENT OF LOG-POSTERIOR

We are interested in the gradient of the following expression with respect to  $\phi_u$  (the embedding sample for the user):

$$-\frac{1}{2}(\phi_u - \phi_{\mu,U}(u))^\top (\phi_{\sigma,U}(u)^\top \phi_{\sigma,U}(u))^{-1} (\phi_u - \phi_{\mu,U}(u)) + \sum_{k=1}^K \log P(\rho^{(k)} | q^{(k)}, \phi_u) \quad (30)$$

For the first term that comes from the prior, the gradient is equal to  $-(\phi_{\sigma,U}(u)^\top \phi_{\sigma,U}(u))^{-1} (\phi_u - \phi_{\mu,U}(u))$ . The components of the second term (each log likelihood) depends on the type of query.

### B.1 Attribute Queries

*Mean Slate Model.* Here, we have

$$\log P(\rho^{(k)} | q^{(k)}, \phi_u) = \log \Phi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_{I,\bar{S}^{(k)}})}{\sigma_{g^{(k)}}} \right), \quad (31)$$

where  $\rho^{(k)}$  is either  $-1$  or  $1$ . We can write the gradient of this expression with respect to  $\phi_u$  as follows by using the relation between  $\phi_u$  and  $\phi_{I,u}^*$ :

$$\frac{\varphi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_{I,\bar{S}^{(k)}})}{\sigma_{g^{(k)}}} \right)}{\Phi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_{I,\bar{S}^{(k)}})}{\sigma_{g^{(k)}}} \right)} \rho^{(k)} z \left( \frac{\phi_{g^{(k)}}}{\|\phi_u\|_2} - \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_{I,\bar{S}^{(k)}}) \phi_u}{\|\phi_u\|_2^3} \right), \quad (32)$$

where  $\varphi$  is the standard normal pdf, and  $z = \max_{i \in \bar{I}} \|\phi_I(i)\|_2$ .

*Mean Probability Model.* We start with

$$\log P(\rho^{(k)} | q^{(k)}, \phi_u) = \log \frac{1}{|S^{(k)}|} \sum_{i \in S^{(k)}} \Phi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(i))}{\sigma_{g^{(k)}}} \right), \quad (33)$$

where  $\rho^{(k)}$  is either  $-1$  or  $1$ . Again by using the relation between  $\phi_u$  and  $\phi_{I,u}^*$ , we can write the gradient of this expression with respect to  $\phi_u$  as follows:

$$\frac{\sum_{i \in S^{(k)}} \varphi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(i))}{\sigma_{g^{(k)}}} \right) \rho^{(k)} z \left( \frac{\phi_{g^{(k)}}}{\|\phi_u\|_2} - \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(i)) \phi_u}{\|\phi_u\|_2^3} \right)}{\sum_{i \in S^{(k)}} \Phi \left( \rho^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(i))}{\sigma_{g^{(k)}}} \right)}. \quad (34)$$

## B.2 Item Queries

We ignore the temperature  $T$  for simplicity. Here, we have

$$\begin{aligned} \log P(\rho^{(k)} | q^{(k)}, \phi_u) &= \log \frac{\exp(\phi_u^\top \phi_I(\rho^{(k)}))}{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i))} \\ &= \phi_u^\top \phi_I(\rho^{(k)}) - \log \sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i)), \end{aligned} \quad (35)$$

where  $\rho^{(k)}$  is an item from  $S^{(k)}$ . We can write the gradient of this expression with respect to  $\phi_u$  as follows:

$$\phi_I(\rho^{(k)}) - \frac{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i)) \phi_I(i)}{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i))}. \quad (36)$$

## B.3 Item-Plus-Attribute Queries

We again start with the response model:

$$\begin{aligned} \log P(\rho^{(k)} | q^{(k)}, \phi_u) \\ = \log \frac{\exp(\phi_u^\top \phi_I(\rho_1^{(k)}))}{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i))} + \log \Phi \left( \rho_2^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(\rho_1^{(k)}))}{\sigma_{g^{(k)}}} \right), \end{aligned} \quad (37)$$

where  $\rho_1^{(k)} \in S^{(k)}$  and  $\rho_2^{(k)} \in \{-1, +1\}$ . We can write the gradient of this expression with respect to  $\phi_u$  simply as a sum of the two gradients we derived before:

$$\begin{aligned} \phi_I(\rho_1^{(k)}) - \frac{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i)) \phi_I(i)}{\sum_{i \in S^{(k)}} \exp(\phi_u^\top \phi_I(i))} + \frac{\varphi \left( \rho_2^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(\rho_1^{(k)}))}{\sigma_{g^{(k)}}} \right)}{\Phi \left( \rho_2^{(k)} \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(\rho_1^{(k)}))}{\sigma_{g^{(k)}}} \right)} \times \\ \frac{\rho_2^{(k)} z \left( \frac{\phi_{g^{(k)}}}{\|\phi_u\|_2} - \frac{\phi_{g^{(k)}}^\top (\phi_{I,u}^* - \phi_I(\rho_1^{(k)})) \phi_u}{\|\phi_u\|_2^3} \right)}{\sigma_{g^{(k)}}}. \end{aligned} \quad (38)$$