Keyphase-based Bayesian Preference Elicitation for Conversational Recommendation

Abstract

Preference elicitation (PE) is a popular approach for interactive, conversational recommendation. A well-known PE approach is to perform a Bayesian update on the belief over a user's preferences based on their response to queries chosen by the PE system. However, existing methods often produce queries that require a user to pick preferred items from a short candidate list, which may be difficult if the user is unfamiliar with the items. In this paper, we propose an alternative PE approach that interacts with users through queries based on natural language keyphrase descriptions. Using Bayesian updating, the model incrementally updates a user's belief and chooses a candidate query among keyphrases. Notably, the model works with the latent representation of the user, item, and keyphrases, which support modeling their complex mutual interactions. Our empirical evaluation on two real-world datasets shows that the proposed query selection strategies effectively update user beliefs, leading to high-quality recommendations for a user with a minimal number of queries.

1 Introduction

In an era of pervasive Machine Learning and AI, interaction with search and recommendation systems has become increasingly interactive and dialog-based (e.g. Google Assistant, Amazon Alexa, Apple Siri). However, existing conversational assistants do not typically support the level of personalization commonly found in recommender systems and also lack a general ability to actively elicit preference information during an interactive session.

The field of preference elicitation (PE) historically does address the problem of how to strategically interact with a user to narrow down their session-based preferences. One common approach for preference elicitation is the traditional paradigm of using Bayesian updating from preference feedback to infer a posterior belief over a user's preferences. However, this Bayesian paradigm has rarely been revisited in the context of modern recommender systems with millions of items and a latent user-item representation, which has driven many recent technological advances in recommendation.

Recently, a few works have revisited PE in the era of latent recommendation methods. For example, [6] combine a Bayesian Matchbox recommender [13] with bandit-style query generation. And [14] introduces a continuous formulation of Expected Value of Information (EVOI) [9] as a differentiable network that can be optimized using gradient methods. Most of these previous works attempt to generate queries by asking about item preferences. However, the user may often lack the information to determine a preference for specific items (e.g., items the user has not seen). For this reason, it has been common in PE to query attributes of items instead of items themselves [12]. Furthermore, many previous works focus on the cold-start setting and do not explictly handle the incorporation of prior interaction histories in order to warm-start with an informed belief over user preferences.

In this paper, we revisit PE to develop alternative ways to adapt existing Bayesian methodology to work with a rich shared latent representation of users, items, and their natural language descriptions as expressed in online reviews or tags. Specifically, the proposed PE approach queries expressive language-based user preferences based on keyphrases mined from online tags. In our approach, we first leverage a user's tag data to generate a keyphrase embedding that is mapped to their embedded representation in a standard latent user-item recommendation approach. From the observed item interactions, we leverage Bayesian techniques to infer a posterior belief over a user's preferences, which can also generate a distribution over a user's predicted keyphrase usage. By actively selecting keyphrase preferences to query a user, we can update their latent belief representation and ultimately use this to provide session-based item recommendations to the user. In contrast to existing PE systems, we note that our proposed PE approach actively queries in the space of natural language keyphrases while updating a common posterior belief in user's preferences — all enabled through data that links historical user preferences for items with their tags.

We evaluate a variety of keyphrase query selection strategies for our PE system as well as the benefits of warm-starting the elicitation process with a user belief based on their historical interactions. Our empirical evaluation on two real-world datasets shows that the proposed query selection strategies effectively update user beliefs, leading to high-quality recommendations for a user with a minimal number of queries.

2 Preliminary

2.1 Notation

We begin by defining notation used throughout this paper:

- $R \in \mathbb{B}^{m\times n}$: A user-item implicit feedback matrix. m represents number of users, and n represents number of items. The value of an entry $r_j^{(i)}$ is either 1 (observed interaction) or 0 (no interaction).
- $Y \in \mathbb{Z}^{m\times h}$: A user-keyphrase matrix. Keyphrases are user-generated metadata, typically a single word or a short phrase. The value of an entry $y_j^{(i)}$ represents the frequency of keyphrase j used by user i.
- $\bar{U} \in \mathbb{R}^{m\times d}$: User representation matrix. d represents the number of latent dimensions. $\bar{\mathbf{u}}^{(i)}$ denotes the latent representation vector of user i learned from R.
- $X \in \mathbb{R}^{n\times d}$: Item representation matrix. $\mathbf{x}_j$ denotes the latent representation vector of the jth item learned from R.
- $K \in \mathbb{R}^{h\times d}$: Keyphrase representation matrix. h represents the number of keyphrases in the metadata. $\mathbf{k}_l$ denotes the latent representation vector of the lth keyphrase learned from Y.
- $P(\mathcal{U}^{(i)})$: Representation distribution of user i, which serves as our probabilistic belief of user i. User representation $\mathbf{u}$ is sampled from $P(\mathcal{U}^{(i)})$, i.e. $\mathbf{u} \sim P(\mathcal{U}^{(i)})$.
- $P(\mathcal{Y}_j|\mathbf{u})$: A conditional distribution of a multivalued discrete keyphrase random variable $\mathcal{Y}_j$ given user representation $\mathbf{u}$.

2.2 Preference Elicitation for Recommendation

In a static recommendation task (where data is not dynamically collected), latent factor models produce a recommendation score for each user and item through a simple dot product

$$\hat{r}_j^{(i)} = \mathbf{x}_j^T \bar{\mathbf{u}}^{(i)}, \tag{1}$$

where the user representation $\bar{\mathbf{u}}^{(i)}$ and item representation $\mathbf{x}_j$ are learned from historical interactions matrix R.

However, the preference of a user varies based on their daily session-based needs, which is often uncertain. Thus, recommendation approaches that maintain a user preference distribution (belief) $P(\mathcal{U}^{(i)})$ over user i based on their observed interactions [13, 10, 8] provide a way to incrementally maintain a belief over a user's immediate preferences.

Probabilistic representation of the user preference is particularly crucial for the conversational recommendation task, where a system gradually updates its belief of user preference by asking strategic questions. The updated preference belief can then be leveraged to produce personalized recommendations and refine follow-up questions. This is reflected in the workflow that we follow in this paper as outlined in Figure 1.

With this motivation, we now proceed to review four fundamental concepts of the Bayesian Preference Elicitation in the literature, following an established framework [3, 14]: namely, we review the user utility function, query type, belief update, and query strategy.

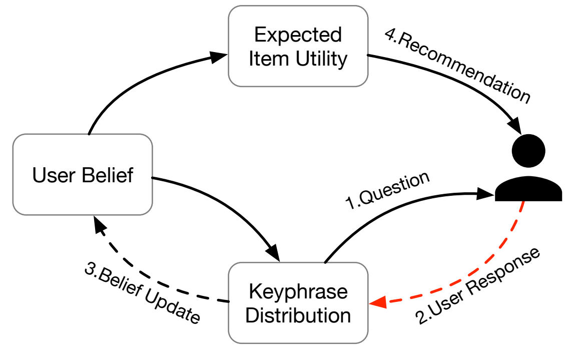

Figure 1: The workflow of our keyphrase-based Preference elicitation. From prior user beliefs, the system generates predictive frequency distributions for each keyphrase. These distributions are used to select a keyphrase-based query that not only minimizes the uncertainty of current user belief but also considers the user's current preference (1). From the user's response (2), the model incrementally updates the user's belief using Bayesian updating (3). After a few interactions, the system recommends items based on expected utility (4).

Utility Function

In the Preference Elicitation literature, the recommendation scoring function often acts as a utility function $u(\mathbf{x}_j; \bar{\mathbf{u}}^{(i)})$ that produces the utility of item j for user i based on the respective representations. As an concrete example, we can treat the scoring function in Equation 1 as a linear utility function

$$u(\mathbf{x}_j; \bar{\mathbf{u}}^{(i)}) = \mathbf{x}_j^T \bar{\mathbf{u}}^{(i)}.$$

In the probabilistic setting, since we maintain a user belief $P(\mathcal{U})$ instead of a deterministic vector $\bar{\mathbf{u}}$, the utility of an item j for a user must be computed in expectation:

$$EU(\mathbf{x}_j; P(\mathcal{U}^{(i)})) = \mathop{\mathbb{E}}_{P(\mathcal{U}^{(i)})}[\mathbf{x}_j^T \mathbf{u}] = \mathbf{x}_j^T \mathop{\mathbb{E}}_{P(\mathcal{U}^{(i)})}[\mathbf{u}] \tag{2}$$

Correspondingly, we recommend the item with maximum expected utility given $P(\mathcal{U})$ to the user as it also has highest recommendation score:

$$\mathbf{x}^*_{P(\mathcal{U}^{(i)})} = \arg\max_{\mathbf{x}_j} EU(\mathbf{x}_j; P(\mathcal{U}^{(i)})) \quad j \in \{1..n\} \tag{3}$$

Query Type

Asking precise questions is critical for the interactive recommender system to minimize cognitive burden on the user. In order to control the query quality, existing works often concentrate on specific query types $\mathcal{Q}$ (cf. [6, 8, 14, 16]). In particular, the pairwise comparison query [6, 8] asks *either-or* questions, whereas the slate query [14, 16] asks *multiple choice* questions. While the query type used in the literature varies based on the user belief updating mechanism, most existing works construct their preference queries on the item space. In other words, all questions are directly related to knowledge and preference over a subset of items.

Belief Update

Given a response $\mathbf{r}^q$ to the question $q \in \mathcal{Q}$, the interactive recommender system can update its estimation of user preferences by incorporating the new information through a

Bayesian update. Formally, the density of user belief $P(\mathbf{u})$ is updated as follows:

$$P_{t+1}(\mathbf{u}) = P(\mathbf{u}|r^q) = \frac{P(r^q|q,\mathbf{u})P_t(\mathbf{u})}{\int_{\mathcal{U}} P(r^q|q,\mathbf{u})P_t(\mathbf{u})d\mathbf{u}} \quad (4)$$

With each step of interaction between the user and the recommender system, the system should incrementally improve its certainty in the user's preferences [15].

Query Strategy

While collecting answers of any query (even random) should monotonically improve the certainty of utility estimation w.r.t. some items, we are interested in looking for an optimal query that contributes most to a user belief update at each query iteration and ultimately leads to effective recommendation.

Existing works in the literature suggested various criteria for good questions. For research under the Expected Value of Information (EVOI) paradigm [9, 8, 14, 4], the best query is the one that optimizes posterior expected utility. Based on the requirements of the tasks, the query selection process could be either myopic [8, 14] or sequential [4]. As a contrast, for methods based on Information Theory, the best query is the one that maximizes information gain. In particular, *active learning* style queries [6, 18] belong to this category and inspire our query strategy approach.

3 Methodology

As mentioned in Section 2.2, most of the existing works build up their query set $\mathcal{Q}$ on the item space. As a consequence, queries are limited to ask user about their preference on certain items. Such types of queries are undesirable for users with limited knowledge of items and impose a significant cognitive burden if there are a large number of items in queries or they are difficult to judge.

In this work, we suggest to perform Bayesian preference elicitation based on queries in a *keyphrase* space, where we use simple binary response queries. Compared to queries over the item space, which the user may not be familiar with, we assume that users are generally familiar with the meaning of keyphrase descriptions. However, querying in keyphrase space requires aligning a user's keyphrase preferences with their item preferences.

In the following subsections, we start by describing how we establish mappings among the three components of users, items, and keyphrases. Then, we introduce our approach to updating user beliefs based on their response to keyphrase-based queries.

3.1 Aligning the Latent Keyphrase Representation

We assume observations such as user-item and user-keyphrase interactions are generated from a user's latent preference representation, which can be learned through a data-driven approach. Mathematically, we minimize the following objective function:

$$\arg\min_{U} \sum_{i=1}^{m} \mathcal{L}(\mathbf{r}^{(i)}, f(\mathbf{u}^{(i)})) + \mathcal{L}(\mathbf{y}^{(i)}, g(\mathbf{u}^{(i)})) \quad (5)$$

where $\mathcal{L}(\cdot,\cdot)$ is an arbitrary loss function that penalizes the gap between the two input arguments.

Unfortunately, as the user latent representations are not observable, the optimization process needs to align the user representation between the two loss functions. In this work, we adopt the simple semantics that both functions f and g are linear, which is an assumption commonly used in state-of-the-art matrix factorization style methods [11, 13, 7]. In addition, because we ultimately want to focus on learning a latent user representation that is highly predictive of a user's item preferences, we focus initially on optimizing the latent user representation w.r.t. item preferences and then learn a mapping of this latent representation to a user's keyphrase usage.

Specifically, we decompose the user-item matrix R into two low-dimensional user and item latent matrices $\bar{U}$ and X through a truncated Single Value Decomposition (SVD) inspired by the PureSVD [7] recommendation method:

$$R \approx \bar{U}\Sigma\bar{V}^T \quad \textit{and} \quad X = \bar{V}\Sigma^T. \quad (6)$$

Then, we can minimize a linear regression objective to approximate the mapping from a user's latent representation to their observed keyphrase usage:

$$\arg\min_{K} \sum_{i} (\mathbf{y}^{(i)} - \bar{\mathbf{u}}^{(i)}K^T)^2 + \lambda\|K\|_F, \quad (7)$$

where we note that the parameters K are the keyphrase embedding matrix.

Since we optimize the Mean Squared Error objective with L2 regularization, this objective implicitly indicates that the conditional likelihood of keyphrase observations given the user's latent representation is a univariate Gaussian such that

$$P(\mathcal{Y}_l|\mathbf{u}^{(i)}) = \mathcal{N}(\mathbf{k}_j^T\mathbf{u}^{(i)}, \sigma^2) \quad (8)$$

where $\sigma \geq 0$.

The alignment described above will allow us to provide an initial belief of the user preference $P(\mathcal{U})$ as we describe in the next section. However, since our aim is to actively elicit feedback to keyphrase-based queries, we'll also need to update the belief in a users' preferences based on this feedback, which we discuss next.

3.2 Keyphrase-based Belief Update

Now we describe how the proposed system accepts the interactive feedback of users based on their responses to keyphrase-based preference queries.

Initial Prior Belief

Before interactive user belief updating based on preference feedback begins, the system should be initialized with some prior belief $P_0(\mathcal{U}^{(i)})$ for each user as follows:

$$P_0(\mathcal{U}^{(i)}) = \mathcal{N}(\bar{\mathbf{u}}_0^{(i)}, \Sigma_0^{(i)}). \quad (9)$$

Here, the vector $\bar{\mathbf{u}}_0^{(i)}$ and matrix $\Sigma_0^{(i)}$ represent the mean and the covariance matrix, respectively. We assume a prior with spherical covariance such that $\Sigma_0^{(i)} = \beta^2 I$, where $\beta \geq 0$. For the prior mean, we use the user embedding $\bar{\mathbf{u}}^{(i)}$ learned from Equation 6. At this point, we can compute the initial expected utility of an item $\mathbf{x}$ given this prior as follows:

$$EU(\mathbf{x}; P_0(\mathcal{U}^{(i)})) = \mathbf{x}^T \mathop{\mathbb{E}}_{P_0(\mathcal{U}^{(i)})}[\mathbf{u}] = \mathbf{x}^T\bar{\mathbf{u}}^{(i)} = \hat{r} \quad (10)$$

Query Response Handling

The proposed model performs Bayesian preference elicitation on the keyphrase space with a binary response, i.e., given a query q, the user response to the system is limited to the form of $r^q \in \{yes(\text{preferred}), no(\text{not preferred})\}$.

In order to handle the user response efficiently, we propose the following simple mechanism for instantiating the observed keyphrase frequency based on the binary response:

$$\tilde{y}_l = \begin{cases} \max_{l'} y_{l'} & r_l^q = yes \\ 0 & r_l^q = no \end{cases}, \quad \forall l' \in \{1..h\}. \tag{11}$$

If, for example, we had set the yes case to a value of 1 instead, this may have little effect on the Bayesian update if the keyphrase occurs frequently in the corpus; hence, we set the user's positive response to the maximum keyphrase frequency in the historical record to calibrate the magnitude of the user response with the keyphrase popularity in the data.

Belief Update

Once the system takes user response and corresponding keyphrase frequency $\tilde{y}_l$, we can then update the user's belief through Bayes' theorem. Formally, the posterior density of user preference belief is derived as follows:

$$P_{t+1}(\mathbf{u}) = P_{t+1}(\mathbf{u}|\mathbf{k}_l, \tilde{y}_l) \propto P_t(\tilde{y}_l|\mathbf{k}_l, \mathbf{u})P_t(\mathbf{u}) \tag{12}$$

Since both the likelihood and prior are Gaussian distributions and form a conjugate prior-likelihood pair, the posterior distribution is also Gaussian with the following form:

$$P_{t+1}(\mathcal{U}^{(i)}) = \mathcal{N}(\bar{\mathbf{u}}_{t+1}^{(i)}, \Sigma_{t+1}^{(i)}) \tag{13}$$

where

$$\bar{\mathbf{u}}_{t+1}^{(i)} = \Sigma_{t+1}^{(i)}({\Sigma_t^{(i)}}^{-1}\bar{\mathbf{u}}_t^{(i)} + \sigma^{-2}\tilde{y}_l\mathbf{k}_l) \tag{14}$$

and

$$\Sigma_{t+1}^{(i)} = ({\Sigma_t^{(i)}}^{-1} + \sigma^{-2}\mathbf{k}_l\mathbf{k}_l^T)^{-1}. \tag{15}$$

3.3 Query Selection

Now we describe how the predictive keyphrase distribution is utilized to select queries that not only minimize the uncertainty of a user's belief but also consider the user's most likely top preferences.

Predictive Keyphrase Distribution

Full Bayesian inference makes predictions by averaging over all likely explanations under the posterior distribution. We calculate the predictive distribution of keyphrase usage for user i as follows:

$$P_t^{(i)}(\mathcal{Y}_l) = \int_{\mathcal{U}^{(i)}} P(\mathcal{Y}_l|\mathbf{u})P_t(\mathbf{u})d\mathbf{u} \tag{16}$$

which yields the following Gaussian distributions:

$$P_t^{(i)}(\mathcal{Y}_l) = \mathcal{N}(\mathbf{k}_l^T\bar{\mathbf{u}}_t^{(i)}, \sigma^2 + {\mathbf{k}_l}^T\Sigma_t^{(i)}\mathbf{k}_l). \tag{17}$$

The mean and variance of this distribution respectively represent the predicted mean (expectation) of user i's usage frequency for keyphrase l and its confidence in this prediction.

Entropy Search

In the preference elicitation context, we aim to estimate a user's preferences by incrementally reducing the model uncertainty in the user belief $P(\mathcal{U}^{(t)})$. To this end, maximizing information gain in the form of

$$\mathcal{IG}_{t+1}^{(i)} = \max \mathcal{H}_t^{(i)} - \mathcal{H}_{t+1}^{(i)} \tag{18}$$

is probably the most well known approach [5], where the uncertainty of the user belief is an entropy term

$$\mathcal{H}_t = -\int_{\mathcal{U}^{(i)}} P_t(\mathbf{u}) \ln P_t(\mathbf{u})d\mathbf{u} \tag{19}$$

As $P_t(\mathcal{U}^{(i)})$ is a Gaussian distribution, entropy has an analytical closed-form solution such that:

$$\mathcal{H}_t^{(i)} = \frac{1}{2}\ln\left((2\pi e)^d|\Sigma_t^{(i)}|\right) \tag{20}$$

where the entropy is a function of covariance matrix $\Sigma_t^{(i)}$.

According to Equation 15 and the rank-one update property of determinants, we note the information gain could be compactly computed as:

$$\mathcal{IG}_{t+1}^{(i)} = \frac{d}{2}\ln\left(1 + \sigma^{-2} + \mathbf{k}_l^T\Sigma_t^{(i)}\mathbf{k}_l\right) \tag{21}$$

Overall, maximizing the information gain is equivalent to looking for the keyphrase l that maximizes the term $\mathbf{k}_l^T\Sigma_t^{(i)}\mathbf{k}_l$. In addition, according to Equation 17, this also intuitively indicates that we can select the query keyphrase whose distribution has the maximum variance (uncertainty) for the user.

Upper Confidence Bound

While entropy search is the (myopically) optimal solution to reduce overall uncertainty, it does not necessarily narrow down the top candidates for final recommendation that are the primary goal of the preference elicitation procedure.

To address this deficiency of entropy, the Upper Confidence Bound (UCB) [1] is an alternative query selection strategy. Compared to entropy search, the UCB function maintains exploitation $\mu_t^{(i)}(\mathcal{Y}_l)$ and exploration $\sigma_t^{(i)}(\mathcal{Y}_l)$ terms explicitly such that:

$$UCB_t^{(i)}(\mathcal{Y}_l) = \underbrace{\mathbf{k}_l^T\bar{\mathbf{u}}_t^{(i)}}_{\mu_t^{(i)}(\mathcal{Y}_l)} + \eta\underbrace{(\sigma^2 + {\mathbf{k}_l}^T\Sigma_t^{(i)}\mathbf{k}_l)}_{\sigma_t^{(i)}(\mathcal{Y}_l)} \tag{22}$$

where η is a relative weighting hyper-parameter. UCB strategically selects the keyphrase with the highest upper confidence bound in it's search for the highest predicted frequency keyphrases that should in turn provide information about a user's top item preferences. Algorithm 1 demonstrates the query selection strategy of UCB.

4 Experiments and Evaluation

In this section, we evaluate our proposed model through four experiments in order to answer the following two research questions:

- RQ1: Of the proposed keyphrase query selection strategies, which provides the fastest learning?
- RQ2: Does warm-starting with an initial belief have an impact on overall elicitation performance?

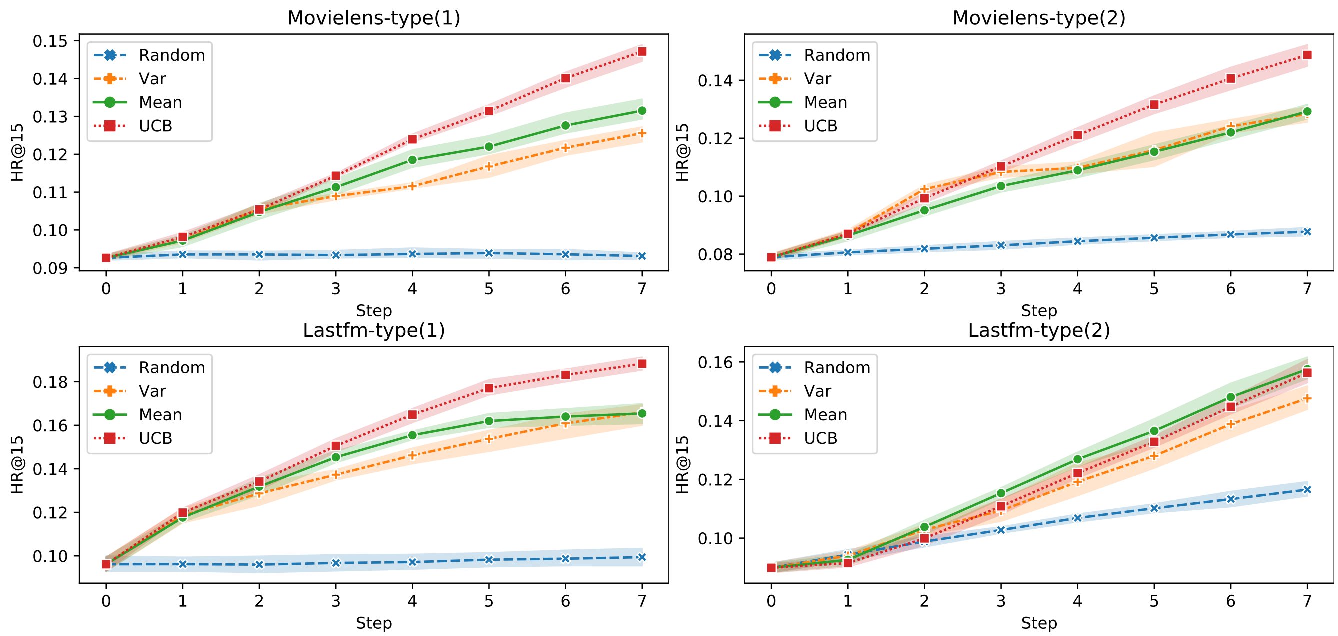

Figure 2: HR@15 comparison between query selection strategies during the conversation session on Movielens and Lastfm with 95% CI. Each type represents two different simulation models described in 4.2.

Algorithm 1 Query selection using UCB

$P_0(\mathcal{U})$: prior belief over $\mathcal{U}$
for elicitation process $t \in range(0, T)$ **do**
    Form $P_t(\mathcal{Y}_l)$ for unmeasured $\quad l \in \{1..h\}$
    Choose $\mathbf{k}_l$ s.t. $\arg\max_l UCB_t(\mathcal{Y}_l)$
    Using $P_t(\mathcal{U})$, $\mathbf{k}_l$, $\tilde{y}_l$, update belief to $P_{t+1}(\mathcal{U})$

Table 1: Summary of datasets.

Dataset	# Users	# Items	# keyphrases	# Interactions
Movielens	283,201	38,425	5,146	27,634,357
Lastfm	1,892	11,321	713	85,513

4.1 Datasets

We conduct experiments on two datasets: Hetrec-LastFM (LastFM) for music artist recommendation and MovieLens-20M (MovieLens) for movie recommendation. Both datasets have natural language-based keyphrase tag assignments of items provided by each user. For preprocessing, we only keep keyphrases that have been assigned by at least 10 users and 5 items to remove tags that may have user and item bias [2]. We partition the dataset into 5 folds and test the model using cross-validation. For every test, we split the remaining interaction in the ratio of 7:3 for the training and validation set. Table 1 shows a summary of the two datasets.

4.2 Simulation for Multi-step Elicitation

Because preference elicitation is a dynamic process, we simulate a conversational session for each observed interaction between users and items as done in prior elicitation and interactive recommendation work [8, 14, 17]. Specifically, given an observed user, item, keyphrase set triplet interaction $(\mathsf{u}, \mathsf{i}, \mathsf{k})$ in the test set, we treat i as the ground truth target item to seek and treat the keyphrase tags k as the corresponding keyphrases preferred by the user in this session. In simulation, user u will respond positively to the binary preference query of k, whereas queries for other keyphrases irrelevant to i (i.e., no user assigned those keyphrases to describe i) will receive a negative response. For keyphrases that other users used to describe i but not u, the appropriate simulated user response is not obvious. Thus we tested with two simulation models: (1) To eliminate the ambiguity of the user response, we exclude those keyphrases from the list of possible queries. (2) Users will respond positively to every item-

Table 2: Hyper-parameters tuned on the experiments.

name	Functionality	Value
σ	Variance for conditional distribution of keyphrase	0.2
Σ_0	Initial covariance matrix for weight distribution	$0.2I$
λ	L2 Regularization	1.0
η	Weight in UCB	1.0
d	Latent dimension	128

Table 3: Example keyphrase and top-4 similar keyphrases according to cosine distance as learned on each dataset.

Dataset	keyphrase	Top-4 Similar keyphrases (Cosine Similarity)
MovieLens	social commentary	loneliness, depressing, stylized, political
	conspiracy	thriller, assassination, cia, assassin
	futuristic	sci-fi, future, dystopia, artificial intelligence
	family	children, kids, christmas, holiday
Last.fm	j-pop	japanese, j-rock, jpop, anime
	country	modern country, country rock, oldies, great song
	hardcore	metalcore, deathcore, screamo, post-hardcore
	metal	hard rock, nu metal, heavy metal, alternative rock

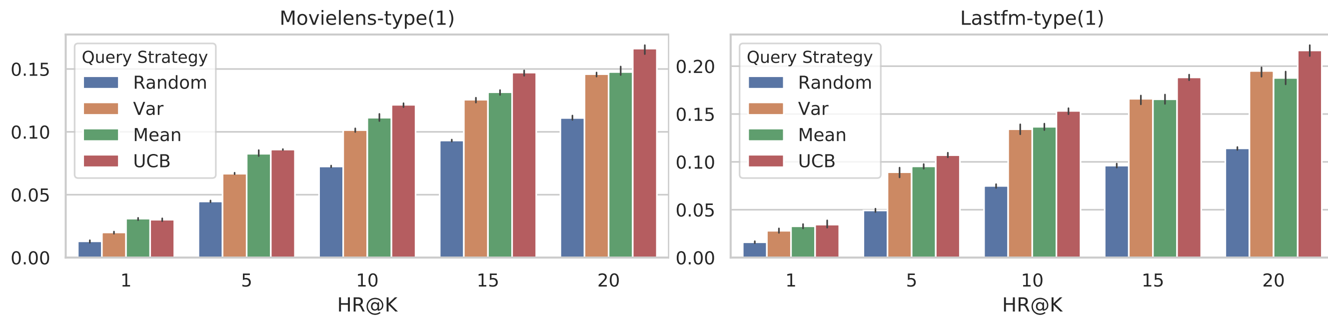

Figure 3: HR@k for different k values after 7th step on Movielens and Lastfm with 95% CI.

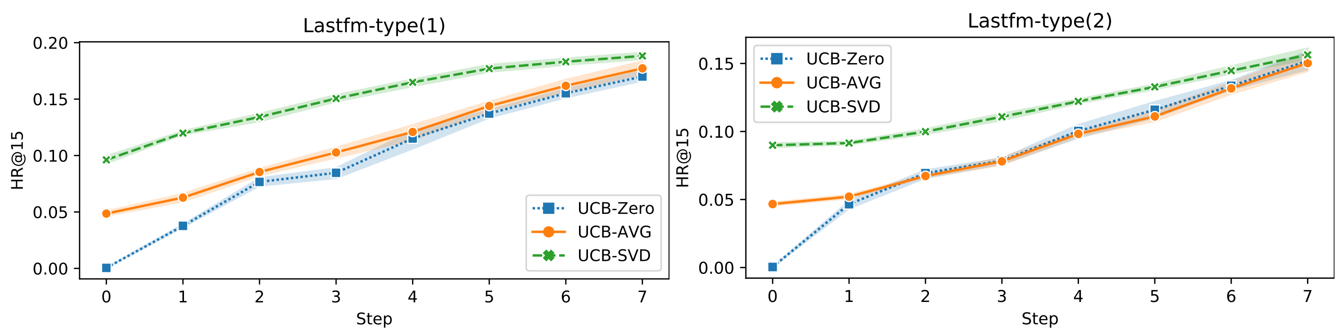

Figure 4: HR@15 comparison between different prior belief initialization during the conversation session Lastfm with 95% CI. Each type represents two different simulation models described in 4.2.

relevant keyphrase, assuming they know all attributes of i.

To rule out interactions with too few possible positive answers, we use interactions that have more than 3 keyphrases in LastFM and 5 keyphrases in MovieLens. Finally, to consider the semantic similarity between keyphrases, we assume that users respond positively to the keyphrases who have higher than 0.9 cosine similarity with k in the simulation model (2). Table 2 shows the hyperparameters used in these experiments. We measure recommendation quality using Hit Ratio(HR)@k with 95% confidence intervals (CI).

4.3 Performance Comparison

Figure 2 shows the HR@15 comparison between the following 4 query selection strategies: Random, exploitation-only (Mean), exploration-only (Var, a.k.a., entropy search), and UCB. We did not compare with EVOI here since it would require an extension of this work to develop a probabilistic user response model for keyphrase queries. Results indicate that while every query selection strategy leads to better recommendation quality, UCB results in the most stable performance improvements, solidly outperforming other methods.

Figure 3 shows a performance comparison on HR@k for different k after the 7th interaction. As previously, UCB generally outperforms for every k on each dataset. This validates our hypothesis that jointly choosing high uncertainty *and* highly promising keyphrases leads to user belief updates that most improve final recommendation performance.

4.4 Prior Belief Warm-start Initialization

Figure 4 shows the HR@15 comparison between different warm-start prior initialization strategies during the conversational session using UCB as the query strategy. UCB-SVD uses the user embedding $\bar{\mathbf{u}}$ as the prior mean — a personalized initialization that reflects user history. The other two methods are non-personalized: UCB-Zero initializes a prior mean with zero vector, while the average over all user vectors (the non-personalized "average user") is used as the initial mean in UCB-AVG. The graph clearly indicates that UCB-SVD outperforms non-personalized methods by starting with the most informed recommendation at Step 0; while UCB-SVD dominates for all time steps, the influence of the prior nearly disappears by step 7 suggesting that the UCB query strategy should also be effective for the cold-start setting.

4.5 Empirical Results on Keyphrase Embeddings

Table 3 shows a sample of empirical results on the keyphrase embedding similarity task. We set each row of K (learned from Equation 7) as the representation for the keyphrase and retrieve the most similar keyphrases via cosine similarity. While anecdotal, these results clearly demonstrate that the learned keyphrase embeddings capture semantic similarity.

5 Conclusion

We introduced a novel Bayesian Preference Elicitation framework for conversational recommendation that works with state-of-the-art latent recommendation methods and which actively queries expressive language-based keyphrase preferences instead of preferences over potentially unfamiliar items. Experiments show that our proposed model with a UCB query strategy and warm-start user prior can effectively and quickly provide strong recommendation performance with less communication compared to other strategies.

References

[1] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2002.

[2] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 265–274, 2019.

[3] Craig Boutilier. A pomdp formulation of preference elicitation problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 239–246, 2002.

[4] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, pages 239–246, 2002.

[5] Li Chen and Pearl Pu. Survey of preference elicitation methods. Technical report, SWISS FEDERAL INSTITUTE OF TECHNOLOGY IN LAUSANNE (EPFL, 2004.

[6] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 815–824, 2016.

[7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46, 2010.

[8] Shengbo Guo and Scott Sanner. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[9] R.A. Howard, J.E. Matheson, and Strategic Decisions Group. *Readings on the Principles and Applications of Decision Analysis: General collection*. Readings on the Principles and Applications of Decision Analysis. Strategic Decisions Group, 1983.

[10] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698, 2018.

[11] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[12] David W. Rajala. Review of "decisions with multiple objectives: Preferences and value trade-offs" by ralph l. keeney and howard raiffa. *IEEE Trans. Systems, Man, and Cybernetics*, 9(7):403, 1979.

[13] David H. Stern, Ralf Herbrich, and Thore Graepel. Matchbox: large scale online Bayesian recommendations. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pages 111–120, New York, NY, USA, 2009. ACM.

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

[15] Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, pages 101–108, 2009.

[16] Paolo Viappiani and Craig Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems 23 (NIPS)*. MIT press, 2010.

[17] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 177–186, 2018.

[18] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey O. Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. A cost-effective framework for preference elicitation and aggregation. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 446–456, 2018.