Bayesian Preference Elicitation with Keyphrase-Item Coembeddings for Interactive Recommendation

HOJIN YANG, University of Toronto, Canada

SCOTT SANNER*, University of Toronto, Canada

GA WU†, Borealis AI, Canada

JIN PENG ZHOU, University of Toronto, Canada

Interactive (a.k.a. conversational) recommendation systems provide the potential capability to *personalize* interactions with increasingly prevalent dialog-based AI assistants. In the conversational recommendation setting, a user often has long-term preferences inferred from previous interactions along with ephemeral session-based preferences that need to be efficiently elicited through minimal interaction. Historically, Bayesian preference elicitation methods have proved effective for (i) leveraging prior information to incrementally estimate uncertainty in user preferences as new information is observed, and for (ii) supporting active elicitation of preference feedback to quickly zero in on the best recommendations in a session. Previous work typically focused on eliciting preferences in the space of items or a small set of attributes; in the dialog-based setting, however, we are faced with the task of eliciting preferences in the space of natural language while using this feedback to determine a user's preferences in item space. To address this task in the era of modern, latent embedding-based recommender systems, we propose a method for coembedding user-item preferences with keyphrase descriptions (i.e., not explicitly known attributes, but rather subjective judgments mined from user reviews or tags) along with a closed-form Bayesian methodology for incrementally estimating uncertainty in user preferences based on elicited keyphrase feedback. We then combine this framework with well-known preference elicitation techniques that can leverage Bayesian posteriors such as Upper Confidence Bounds, Thompson Sampling, and a variety of other methods. Our empirical evaluation on real-world datasets shows that the proposed query selection strategies effectively update user beliefs, leading to high-quality recommendations with a minimal number of keyphrase queries.

CCS Concepts: • **Information systems → Collaborative filtering**.

Additional Key Words and Phrases: preference elicitation, interactive recommender system, personalized recommendation

ACM Reference Format:

Hojin Yang, Scott Sanner, Ga Wu, and Jin Peng Zhou. 2021. Bayesian Preference Elicitation with Keyphrase-Item Coembeddings for Interactive Recommendation. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '21), June 21–25, 2021, Utrecht, Netherlands.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3450613.3456814

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). Much of the practical use

*Affiliate to Vector Institute of Artificial Intelligence, Toronto

†The majority of the work was done while the author was at the University of Toronto.

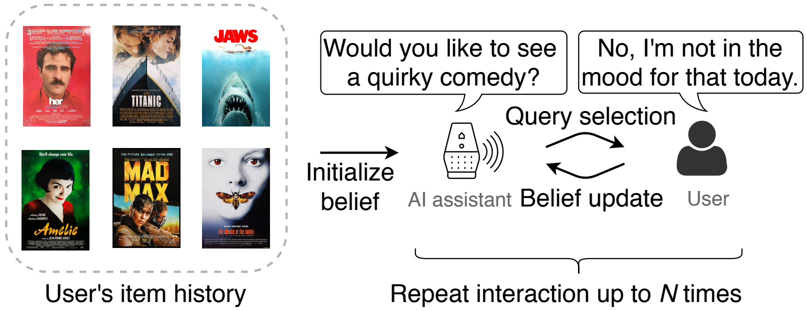

Fig. 1. The workflow of our keyphrase-based preference elicitation approach. First, the system creates a prior belief over a latent representation of user preferences based on their item interaction history. Using this prior, the system actively elicits a user's preference over a keyphrase-based query and performs a Bayesian belief update to the latent user representation based on their response. The system may continue with further rounds of PE or terminate with the best recommendation in expectation w.r.t. current user beliefs.

of conversational assistants has been focused on conversational search, which is primarily concerned with information retrieval to address a user query (i.e. user-initiated), such as inquiring about the weather or requesting songs to play. 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 natural language preference information during an interactive session (i.e. system-initiated) [12, 22].

Historically, the field of preference elicitation (PE) [6, 7] has addressed the problem of how to strategically interact with a user to narrow down their session-based preferences. One common approach is Bayesian preference elicitation that maintains a posterior belief over a user's preferences based on their response to the system's preference queries [4, 14]. Previous PE works typically focused on a system's query in the space of items [10, 28] or a small set of attributes [14, 23]. In the dialog-based setting, however, we are faced with the task of eliciting preferences in the space of natural language while using this feedback to determine a user's preferences in item space.

To this end, we propose a novel PE approach that queries expressive language-based user preferences based on keyphrases mined from online reviews or tags.[1] In contrast to previous work, our framework allows the system to ask subjective keyphrase preference queries as opposed to simply querying explicitly known properties (i.e., attributes). Perhaps more importantly though, compared to item-based queries where a user often lacks the information to determine a preference for previously unseen items (e.g. comparing two movies the user has not watched before), users are generally more comfortable judging language-based descriptions [24].

In order to facilitate this keyphrase approach to PE in the era of modern, latent embedding-based recommender systems [15, 20], we propose a method for *coembedding* user-item preferences with keyphrase descriptions along with a Bayesian methodology for incrementally estimating uncertainty in user preferences. By actively selecting *keyphrase* preferences to query a user, we can efficiently and incrementally update their latent belief representation in closed-form and ultimately use this to provide session-based item recommendations as outlined in Figure 1. In contrast to existing PE systems, our proposed PE approach actively queries in the space of natural language keyphrases while updating a

[1]In this work we simplify language interaction to keyphrases. Historically, the term **conversational recommendation** refers to any sequential recommendation process involving interaction between the recommender system and user [34]. This includes our work and hence our specific use of the term *conversational* for our interactive preference elicitation approach to recommendation.

common posterior belief in a user's latent item and keyphrase preferences — all enabled through historical data that links users, items, and keyphrase descriptions.

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 prior based on their historical interactions. Our empirical evaluation on three real-world datasets shows the proposed query selection strategies effectively update user beliefs, leading to high-quality recommendations with a minimal number of queries.

2 RELATED WORK

Bayesian Preference Elicitation has proved popular due to (i) its natural ability to model uncertainty in user preferences and incorporate long-term preference priors, (ii) its (often) closed-form and efficient incremental belief update when new preference information is observed, and (iii) its ability to support active learning methods that can leverage posterior uncertainty in user preferences. In seminal work on Bayesian PE, Boutilier [4] proposed a partially observed Markov decision process (POMDP) framework with a Bayesian belief state update over user preferences and a methodology for approximately optimal sequential query strategies. Noting the intractability of such sequentially optimal solutions and even approximations thereof, Guo and Sanner [14] instead proposed a heuristic, but more efficient myopic strategy for Bayesian PE based on pairwise comparison queries.

While traditional PE methods tended to use explicit feature representations, recent works have revisited **PE using latent user and item embedding techniques** that have driven many recent technological advances in recommendation [15, 20]. For example, Sepliarskaia et al. [28] propose a PE system that generates a static preference questionnaire to optimize the expectation of the loss function of the underlying latent factor model. In a more online adaptive approach, Christakopoulou et al. [10] combine a Bayesian Matchbox matrix factorization recommender [31] with bandit-style query generation for PE. In a different vein, Vendrov et al. [33] combine a latent factor model with a continuous formulation of Expected Value of Information (EVOI) [16] in a differentiable network optimized using efficient gradient methods. However, we remark that all of these previous Bayesian PE approaches focus on queries in the item space, whereas we are faced with the task of eliciting preferences in the space of natural language keyphrases.

On the other hand, a number of recent methods have aimed to **combine natural language interaction with deep learning-based recommender** and even **reinforcement learning**. For example, Li et al. [18] propose a hierarchical RNN-based language model [30] combined with AutoRec [27] and Pointer Networks [13] for conversational movie recommendations, though they focus on language generation, without explicitly optimizing a PE strategy. In a similar vein, Sun and Zhang [32] focus on reinforcement learning, dialog state tracking, and natural language generation models for conversational recommendation interactions, but do not provide explicit PE optimization functionality. Finally, while Zhang et al. [37] do handle natural language preference feedback in a recommendation system with interactions optimized through reinforcement learning, they do not focus on active PE methods, but rather on an alternative conversational recommendation feedback approach known as *critiquing* [5, 25, 26] — here the user initiates critiques on one or more aspects of the currently recommended item. In contrast to these works, while we do intend to focus on natural language interaction with interactive (conversational) recommendation in this paper, we focus explicitly on system-initiated keyphrase-based active PE query strategies, which appear to be overlooked in the existing literature.

3 BACKGROUND

3.1 Notation

We begin by defining notation used throughout this paper:

- $R \in \mathbb{Z}^{m\times n}$: A user-item feedback matrix. m represents the number of users, and n represents the number of items. In the implicit feedback case, the value of an entry 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 represents the frequency of keyphrase l used by user i.
- $X \in \mathbb{R}^{n\times d}$: 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.
- $P(\mathcal{U}^{(i)})$: The distribution over the latent embedding of user i, which serves as our probabilistic belief in the preferences of user i. The user representation $\mathbf{u}$ is sampled from $P(\mathcal{U}^{(i)})$, i.e. $\mathbf{u} \sim P(\mathcal{U}^{(i)})$.
- $P(\mathcal{R}_j^{(i)}|\mathbf{u}, \mathbf{x}_j)$: A conditional distribution of a scalar item utility continuous random variable $\mathcal{R}_j^{(i)}$ given user representation $\mathbf{u}$ and item representation $\mathbf{x}_j$.
- $P(\mathcal{Y}_l^{(i)}|\mathbf{u}, \mathbf{k}_l)$: A conditional distribution of a scalar keyphrase preference continuous random variable $\mathcal{Y}_l^{(i)}$ given user representation $\mathbf{u}$ and keyphrase representation $\mathbf{k}_l$.

3.2 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 scoring function f

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

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

However, in interactive (conversational) recommendation tasks, a user's preferences are often uncertain initially and can vary from session to session. For handling dynamically evolving user preference beliefs, previous works [14, 19, 31] explicitly model a probabilistic distribution $P(\mathcal{U}^{(i)})$ over user i's preferences. By actively querying the user, the recommender system can incrementally maintain an updated belief over a user's immediate preferences.[2] This updated preference belief can then be leveraged to produce personalized recommendations and refine follow-up questions.

In the literature, the aforementioned approach is referred to as *Bayesian Preference Elicitation* (Bayesian PE). We now proceed to review four fundamental concepts of Bayesian PE, following an established framework [4, 33]: namely, the user utility function, query type, belief update, and query strategy.

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

$$u(\mathbf{x}_j; \mathbf{u}^{(i)}) = \mathbf{x}_j^T \mathbf{u}^{(i)}. \tag{2}$$

[2]One caveat of traditional PE is that it often aims to constrain the range of user beliefs u to a unimodal distribution even though the user may have a multimodal distribution of preferences. Since our work is based on modern embedding-based matrix factorization models for recommendation, it is possible for our belief representation to simultaneously provide high probability to different latent dimensions representing distinct user interests.

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

$$\mathop{\mathbb{E}}_{P(\mathcal{U}^{(i)})}[u(\mathbf{x}_j;\mathbf{u})] = \int_{\mathbf{u}\sim\mathcal{U}^{(i)}} P(\mathbf{u})u(\mathbf{x}_j;\mathbf{u})d\mathbf{u} \tag{3}$$

Correspondingly, the item with maximum expected utility given $P(\mathcal{U}^{(i)})$ is considered as it has the highest recommendation score:

$$\mathbf{x}^*_{P(\mathcal{U}^{(i)})} = \arg\max_{\mathbf{x}_j} \mathop{\mathbb{E}}_{P(\mathcal{U}^{(i)})}[u(\mathbf{x}_j;\mathbf{u})] \quad j \in \{1..n\} \tag{4}$$

3.2.2 Query Type. Asking precise questions is critical for the interactive recommender system to minimize the cognitive burden on a user. In order to control the query quality, existing works often concentrate on specific query types Q (cf. [10, 14, 28, 33, 35]). In particular, the pairwise comparison query [10, 14, 28] asks *either-or* questions, whereas the slate query [33, 35] 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, queries are directly related to knowledge and preference over a subset of items.

3.2.3 Belief Update. Given a user response $\mathbf{a}^q$ to the question $q \in Q$, the interactive recommender system can update its estimation of user preferences by incorporating the new information through *Bayesian updating*. Formally, the density of user belief $P(\mathcal{U}^{(i)} = \mathbf{u})$, or $P(\mathbf{u})$ in short, is updated to its posterior density $P(\mathbf{u}|a^q, q)$ as follows:

$$P(\mathbf{u}|a^q, q) = \frac{P(a^q|q,\mathbf{u})P(\mathbf{u})}{\int_{\mathcal{U}^{(i)}} P(a^q|q,\mathbf{u})P(\mathbf{u})d\mathbf{u}} \tag{5}$$

The system calibrates the belief distribution through a *sequential learning process* [34]. With each step of interaction between the user and the recommender system, the system should incrementally improve its certainty in the user's preferences. That is to say, the belief will be updated as the dialog interaction between user and system unfolds, and we use $P_t(\mathcal{U}^{(i)})$ to denote the user's belief after an interaction at time step t.

3.2.4 Query Strategy. While collecting user responses for an arbitrary preference query should typically improve the certainty of utility estimation for at least one item, we are interested in looking for an optimal query that contributes most to a user belief update at each iteration that ultimately leads to a good final recommendation in expectation.

Existing works in the literature suggested various criteria for good questions. For research under the *Expected Value of Information* (EVOI) paradigm [4, 14, 16, 33], 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 [14, 33] or sequential [4]. In contrast, for methods based on Information Theory, the best query is the one that maximizes information gain. In particular, *active learning* style queries [10, 39] belong to this category and inspire our query strategy approach.

4 METHODOLOGY

As mentioned above, most of the existing works build up their query set Q on the item space. Such queries pose judgment difficulty for users with limited knowledge of the items and impose a significant cognitive burden if there are a large number of queried items or they are difficult to distinguish.

In this work, we leverage a Bayesian preference elicitation model based on queries in a *keyphrase* space with simple binary response queries. Compared to queries over the item space that the user may not be familiar with, users are generally more comfortable judging keyphrase descriptions [24]. Hence, we provide a novel methodology for performing

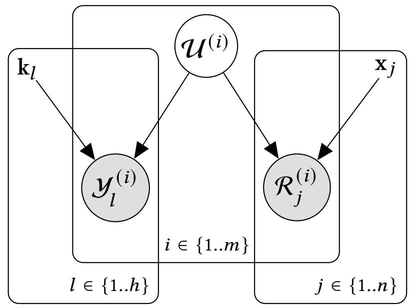

Fig. 2. Probabilistic graphical model underlying our proposed PE system. Both item utility variable $\mathcal{R}$ and Keyphrase preference variable $\mathcal{Y}$ are conditioned on user latent variable $\mathcal{U}$. $\mathcal{R}$ and $\mathcal{Y}$ are shaded as they are observed variables, and the system would thus infer the posterior distribution of $\mathcal{U}$ conditioned on this evidence. $\mathbf{x}$ and $\mathbf{k}$ denote the latent representation for items and keyphrases, respectively. We use plate notation to represent replication of random variables over all users i, items j, and keyphrases l. In realistic settings, the system only observes a subset of $\mathcal{Y}$ and $\mathcal{R}$, thus it is important to strategically query a user's preferences to yield the best final recommendation.

efficient and closed-form belief updates on a user's latent preferences through a two-headed latent factorized Bayesian model (one head for item preference predictions, the other for keyphrase preference predictions) that allows us to co-embed item and keyphrase preferences; the resulting Bayesian model then conveniently dovetails with established methodology for preference elicitation strategies.

In the following subsections, we split our description of the proposed model into two phases. In phase 1, we describe a probabilistic graphical model that establishes a foundation for the keyphrase-based probabilistic elicitation. In phase 2, we introduce our PE approach. Specifically, we show 1) how to update the user belief by interacting with keyphrase-based queries and answers, and 2) how to propose questions to the user so that the user belief update is effective and efficient.

4.1 Phase 1: Probabilistic Model Representation

A fundamental assumption in modern embedding-based recommender systems is that rating and review observations are consequences of interacting latent factors relating to user preferences, item properties and the semantic meaning of keywords [15, 17]. We can formalize such assumptions in a latent probabilistic graphical model as shown in Figure 2.

In this generative graphical latent factor model, a latent user representation $\mathbf{u}^{(i)}$ and latent item representation $\mathbf{x}_j$ combine via inner product to generate a Normal distribution over observed feedbacks $\mathcal{R}_j^{(i)}$ for all items j[3]. Formally,

$$P(\mathcal{R}_j^{(i)}|\mathbf{u}^{(i)}, \mathbf{x}_j) = \mathcal{N}(\mathbf{x}_j^T \mathbf{u}^{(i)}, \beta_{\mathcal{R}}^{-1}), \tag{6}$$

where precision $\beta_{\mathcal{R}}$ is a constant.

Similarly, a latent user representation $\mathbf{u}^{(i)}$ and latent keyphrase representation $\mathbf{k}_l$ combine via inner product to generate a Normal distribution over the "strength" $\mathcal{Y}_l^{(i)}$ of each keyphrase l expressed by i's preferences. Formally,

$$P(\mathcal{Y}_l^{(i)}|\mathbf{u}^{(i)}, \mathbf{k}_l) = \mathcal{N}(\mathbf{k}_l^T \mathbf{u}^{(i)}, \beta_{\mathcal{Y}}^{-1}), \tag{7}$$

where precision $\beta_{\mathcal{Y}}$ is a constant.

[3]In this work we adopt a probabilistic linear model with Gaussian observation noise as (i) existing MF models also have implicit Guassian assumptions that aim to minimize mean squared error [20] and (ii) computationally convenient and closed-form for Bayesian inference.

We use a zero-mean spherical Gaussian for the prior latent user representation $\mathbf{u}^{(i)}$:

$$P(\mathcal{U}) = P_0(\mathcal{U}) = \mathcal{N}(\mathbf{0}, \beta_{\mathcal{U}}^{-1} \cdot I), \tag{8}$$

where precision $\beta_{\mathcal{U}}$ is a constant. Here, we omit superscript (i) as the uninformed belief is identical for every user. We also remark that this prior is identical to the system's belief $P_t(\mathcal{U})$ in the user's latent representation at time step $t = 0$.

Given the above Normal definitions, we can infer that the posterior distribution over a user i's latent representation $\mathbf{u}^{(i)}$ at any time step t (with corresponding evidence set) must be a multivariate Normal distribution:

$$P_t(\mathcal{U}^{(i)}) = \mathcal{N}(\bar{\mathbf{u}}_t^{(i)}, \Sigma_t^{(i)}), \tag{9}$$

where the vector $\bar{\mathbf{u}}_t^{(i)}$ and matrix $\Sigma_t^{(i)}$ respectively denote the mean vector and covariance matrix of the posterior.

While both $\mathbf{x}_j$ and $\mathbf{k}_l$ could be represented as random variables and updated along with the user belief, we note it is unnecessary in most practical settings since the semantic meaning of items and keyphrases are usually shared across users. Thus, in this work, we freeze $\mathbf{x}_j$ and $\mathbf{k}_l$ once they are generated by the coembedding process (discussed next) and thus focus solely on updating the posterior Normal user belief distribution $P_t(\mathcal{U}^{(i)})$ during the PE conversation. This update happens to be a simple, closed-form computation since all random variables are Normally distributed.

4.2 Item and Keyphrase Representation Coembedding

Here, we design a simple but efficient, closed-form coembedding algorithm to learn fixed item $\mathbf{x}_j$ and keyphrase $\mathbf{k}_l$ embeddings shared across all users as previously described.

We begin by noting that Figure 2 is a probabilistic graphical latent factor model in the style of state-of-the-art matrix factorization methods [11, 20, 31]. Indeed, there are actually two matrix factorizations implicit in this model and reflected in the means of Equations 6 and 7. Because we only need $\mathbf{x}_j$ and $\mathbf{k}_l$ as fixed vectors, it would almost suffice to simply compute respective matrix factorizations for the user-item data matrix and the user-keyphrase data matrix. The caveat is that we actually need to share a common user embedding $\mathbf{u}^{(i)}$ to obtain compatible coembeddings for both $\mathbf{x}_j$ and $\mathbf{k}_l$ with $\mathbf{u}^{(i)}$. However, this can be easily done in a two-step process.

Specifically, we first decompose the user-item matrix R into two low-dimensional user and item latent matrices $\dot{U}$ and X through a truncated Single Value Decomposition (SVD) inspired by the PureSVD [11] recommendation method:

$$R \approx \dot{U}\Sigma V^T \quad and \quad X = V\Sigma^T. \tag{10}$$

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

$$\arg\min_K \sum_i (\mathbf{y}^{(i)} - \dot{\mathbf{u}}^{(i)} K^T)^2 + \lambda \|K\|_F, \tag{11}$$

where we note that the parameters K are the keyphrase embedding matrix. Here, $\dot{\mathbf{u}}$ represent "anchor" embeddings to ensure mutual compatibility and alignment of item and keyphrase latent representations with user embeddings $\dot{\mathbf{u}}$.

4.3 Phase 2: Belief Updating

Now we describe how the Bayesian PE system refines user beliefs using both item history and keyphrase queries.

4.3.1 Item-based Bayesian Linear Regression. Before elicitation begins, the system should compute an informed posterior distribution for a user's long-term preferences (equivalently, a warm-start informed prior for PE) based on their previous item history in X. Let $\mathbf{r}^{(i)} \in \mathbb{Z}^m$ denote a vectorization of user i's feedback over all observed interactions in X (e.g., a vector of ordinal ratings for explicit feedback, or a binary vector in case of implicit feedback). Then the posterior density over the user's long-term preferences is

$$P_1(\mathbf{u}) = P(\mathbf{u}|X, \mathbf{r}^{(i)}) \propto P(\mathbf{r}^{(i)}|X, \mathbf{u})P_0(\mathbf{u}) \tag{12}$$

where the likelihood of $\mathbf{r}^{(i)}$ was given in Equation 6. 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:

$$P_1(\mathcal{U}^{(i)}) = \mathcal{N}(\bar{\mathbf{u}}_1^{(i)}, \Sigma_1^{(i)}) \tag{13}$$

where

$$\bar{\mathbf{u}}_1^{(i)} = \beta_{\mathcal{R}} {\Sigma_1^{(i)}}^{-1} X^T \mathbf{r}^{(i)} \tag{14}$$

and

$$\Sigma_1^{(i)} = \beta_{\mathcal{R}} X^T X + \beta_{\mathcal{U}} I. \tag{15}$$

4.3.2 Keyphrase-based Sequential Belief Updating. Starting with the previously computed posterior over long-term preferences, the system should now generate sequential preference elicitation queries for the current user session. Detailed query strategies are covered in the following section. Here, we describe how the system performs a Bayesian update of user beliefs based on user responses to *keyphrase*-based preference queries (i.e., not item preference feedback).

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 $a^q \in \{yes\,(\text{preferred}), no\,(\text{not preferred})\}$. While there may be benefits in allowing more elaborate natural language user responses, this significantly complicates the machine's interpretation of the user's intent. Hence, we intentionally focus on an unambiguous binary response setting in this work to focus our investigation on the ability to select informative keyphrases for preference elicitation.

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

$$\tilde{y}_l = \begin{cases} \tau & a_l^q = yes \\ 0 & a_l^q = no \end{cases}. \tag{16}$$

Here $\tau > 0$ is an integer hyperparameter chosen empirically to match the "strength" of item-based feedback.

Once the system receives a user response and corresponding keyphrase preference $\tilde{y}_l$, we can then update the user's belief as in 12:

$$P_{t+1}(\mathbf{u}) = P_{t+1}(\mathbf{u}|\mathbf{k}_l, \tilde{y}_l) \propto P_t(\tilde{y}_l|\mathbf{k}_l, \mathbf{u})P_t(\mathbf{u}), \text{where } t \geq 1 \tag{17}$$

Notably, the posterior distribution is also Gaussian and can be computed incrementally in closed-form as follows:

$$P_{t+1}(\mathcal{U}^{(i)}) = \mathcal{N}(\bar{\mathbf{u}}_{t+1}^{(i)}, \Sigma_{t+1}^{(i)}) \tag{18}$$

where

$$\bar{\mathbf{u}}_{t+1}^{(i)} = \Sigma_{t+1}^{(i)}({\Sigma_t^{(i)}}^{-1}\bar{\mathbf{u}}_t^{(i)} + \tilde{y}_l \beta_{\mathcal{Y}} \mathbf{k}_l) \tag{19}$$

and

$$\Sigma_{t+1}^{(i)} = ({\Sigma_t^{(i)}}^{-1} + \beta_{\mathcal{Y}} \mathbf{k}_l \mathbf{k}_l^T)^{-1}. \tag{20}$$

4.3.3 Relationship Between Precisions. The two likelihood precisions $\beta_{\mathcal{R}}$ and $\beta_{\mathcal{Y}}$ introduced earlier play a critical role in Bayesian updating, as those indicate the system's degree of confidence in how precise each observed datum is.

Technically, the larger the precision is, the more the system leverages the information when it updates user beliefs. In equation 19 and 20, for instance, $\beta_{\mathcal{Y}}$ is the weight of $\mathbf{k}_l$ and determines how much it would be reflected in $\bar{\mathbf{u}}_t^{(i)}$ and $\Sigma_{t+1}^{(i)}$. Similarly, $\beta_{\mathcal{R}}$ is the weight of X in equation 14 and 15.

For recommendation performance, it is crucial to set these hyperparameters to reflect the relative importance of each type of feedback. For example, $\beta_{\mathcal{Y}}$ could be set higher than $\beta_{\mathcal{R}}$ when the user's item history is out-of-date.

4.4 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.

4.4.1 Predictive Keyphrase Preference Dist. 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:

$$P_t^{(i)}(\mathcal{Y}_l) = \int_{\mathcal{U}^{(i)}} P(\mathcal{Y}_l|\mathbf{u}, \mathbf{k}_l) P_t(\mathbf{u}) d\mathbf{u} \tag{21}$$

which yields the following Gaussian distributions:

$$P_t^{(i)}(\mathcal{Y}_l) = \mathcal{N}(\mathbf{k}_l^T \bar{\mathbf{u}}_t^{(i)}, \beta_{\mathcal{Y}}^{-1} + \mathbf{k}_l{}^T \Sigma_t^{(i)} \mathbf{k}_l). \tag{22}$$

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

4.4.2 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

$$\mathcal{IG}_{t+1}^{(i)} = \max \mathcal{H}_t^{(i)} - \mathcal{H}_{t+1}^{(i)} \tag{23}$$

is probably the most well-known approach [9], where the uncertainty of the user belief is an entropy term

$$\mathcal{H}_t = -\int_{\mathcal{U}^{(i)}} P_t(\mathbf{u}) \ln P_t(\mathbf{u}) d\mathbf{u} \tag{24}$$

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

$$\mathcal{H}_t^{(i)} = \frac{1}{2} \ln\left((2\pi e)^d |\Sigma_t^{(i)}|\right) \tag{25}$$

where the entropy is a function of covariance matrix $\Sigma_t^{(i)}$. Here, d is the dimension of the latent space.

According to Equation 20 and the rank-one update property of determinants, we note the information gain could be compactly computed as:

$$\mathcal{IG}_{t+1}^{(i)} = \frac{d}{2} \ln\left(1 + \beta_{\mathcal{Y}} + \mathbf{k}_l^T \Sigma_t^{(i)} \mathbf{k}_l\right) \tag{26}$$

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 22, this also intuitively indicates that we can select the query keyphrase whose distribution has the maximum variance (uncertainty) for the user.

4.4.3 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:

$$UCB_t^{(i)}(\mathcal{Y}_l) = \underbrace{\mathbf{k}_l^T \bar{\mathbf{u}}_t^{(i)}}_{\mu_t^{(i)}(\mathcal{Y}_l)} + \eta \underbrace{(\beta_{\mathcal{Y}} + \mathbf{k}_l{}^T \Sigma_t^{(i)} \mathbf{k}_l)}_{\sigma_t^{(i)}(\mathcal{Y}_l)} \tag{27}$$

where η is a relative weighting hyper-parameter. Algorithm 1 demonstrates the query selection strategy of UCB.

As an alternative method, PE systems use Thompson sampling (TS) [8], a stochastic variant of UCB where the system draws a sample from each predictive distribution and chooses the maximum. Both UCB and TS tend to strategically select the keyphrase with the highest upper confidence bound in their search for the most preferred keyphrases that we conjecture should in turn provide information about a user's top item preferences.

Algorithm 1 Query selection using UCB

1: $P_1(\mathcal{U})$: prior belief over $\mathcal{U}$
2: **for** elicitation process $t \in range(0, T)$ **do**
3: Form $P_t(\mathcal{Y}_l)$ for unmeasured $l \in \{1..h\}$
4: Choose $\mathbf{k}_l$ s.t. $\arg\max_l UCB_t(\mathcal{Y}_l)$
5: Using $P_t(\mathcal{U})$, $\mathbf{k}_l$, $\tilde{y}_l$, update belief to $P_{t+1}(\mathcal{U})$

5 EXPERIMENTS AND EVALUATION

In this section, we evaluate our proposed model through four experiments over three real-world datasets in order to answer the following four research questions:

- **RQ1:** Of the proposed keyphrase query selection strategies, which learns user preferences the fastest?
- **RQ2:** Does warm-starting with the user's informed initial belief have an impact on overall elicitation performance?
- **RQ3:** How does the hyperparameter setting of the keyphrase likelihood precision $\beta_{\mathcal{Y}}$ (with a fixed item precision $\beta_{\mathcal{R}}$) affect performance?
- **RQ4:** Anecdotally, do the keyphrase embeddings learned from Equation 11 in our framework capture the semantic similarity between keyphrases in a way that should faciliate generalization from elicited feedback?

5.1 Datasets

We conduct experiments on three datasets: Hetrec-LastFM (LastFM) for music artist recommendation, MovieLens-20M (MovieLens) for movie recommendation, and Yelp for business recommendation. All datasets have natural language-based keyphrase description assignments of items provided by each user. MovieLens and LastFM datasets contain social tags, typically single words or short phrases, that users have assigned to individual movies or artists. For Yelp, we follow the preprocessing steps described in [17] to extract restaurant preference and keyphrase data for Toronto, Canada.

We only keep keyphrases that have been assigned by at least 10 users/items for the MovieLens dataset, and at least 3 users/items for datasets with relatively fewer users, i.e. Yelp and LastFM [2].

We evaluate and tuned hyperparameters using 5-fold cross-validation. For every non-test split, we divide the data in the ratio of 7:3 for the training and validation set. Table 1 provides a summary of the datasets.

Table 1. Summary of datasets.

Dataset	# Users	# Items	# keyphrases	# Interactions
MovieLens	283,201	38,378	4,004	27,631,505
LastFM	1,890	11,424	1,153	85,653
Yelp	2,191	7,356	234	148,569

Table 2. Hyperparameters tuned on the experiments.

Name	Functionality	MovieLens	LastFM	Yelp
$\beta_{\mathcal{U}}$	Uninformed user precision	0.0001	0.0001	0.0001
$\beta_{\mathcal{R}}$	Item precision	0.01	0.01	0.01
$\beta_{\mathcal{Y}}$	Keyphrase precision	10	0.01	0.01
λ	L2 Regularization	0.001	0.001	0.001
η	Weight in UCB	1	1	5
τ	Maximum value of user's past keyphrase preference	5	5	5
d	Latent dimension	200	50	50

5.2 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 [14, 32, 33, 38]. Specifically, given an observed user–item interaction (u,i) in the test set, we treat i as the ground truth target item to seek and treat its keyphrases 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 any k, whereas queries for irrelevant keyphrases will receive a negative response. The target item during evaluation simulations has never been seen by the user in the training data.[4]

For every experiment, we show algorithm performance over a trial length of 10 queries. It is important to note that this is not a suggestion that preference elicitation should proceed for 10 queries in practice, but rather that we are interested in empirically evaluating algorithm performance (and potentially diminishing returns over time) at all possible stopping points ranging from 0 queries up to 10 queries. We remark that in practice, external constraints like time limits or user-initiated directives will often terminate the interaction process.

Training data is used for co-embedding item and keyphrase representations and the initial warm-start user belief update based on item history. Table 2 shows the hyperparameters used in these experiments. We measure recommendation quality using Hit Ratio (HR)@k with 95% confidence intervals (CI).

5.3 RQ1: Performance Comparison

The left column of Figure 3 shows the HR@10 comparison between the following query selection strategies: exploitation-only (**Mean**, a.k.a., greedy search), exploration-only (**Var**, a.k.a., entropy search), Thompson Sampling (**TS**) and Upper Confidence Bound **UCB**. We also compare these with two baselines: **Random** query and keyphrase popularity (**POP**, keyphrase queries are selected in descending order of popularity according to frequency of usage).

Results indicate that on every dataset, UCB results in the most stable performance improvements, generally matching or outperforming *all* other methods. Random performs poorly as expected and TS shows similar performance to

[4]This underscores why we cannot compare to item-based queries. As argued in the introduction, it is difficult for a user to judge previously unseen items, while they are generally able to judge keyphrase item descriptions used for the preference queries in this work.

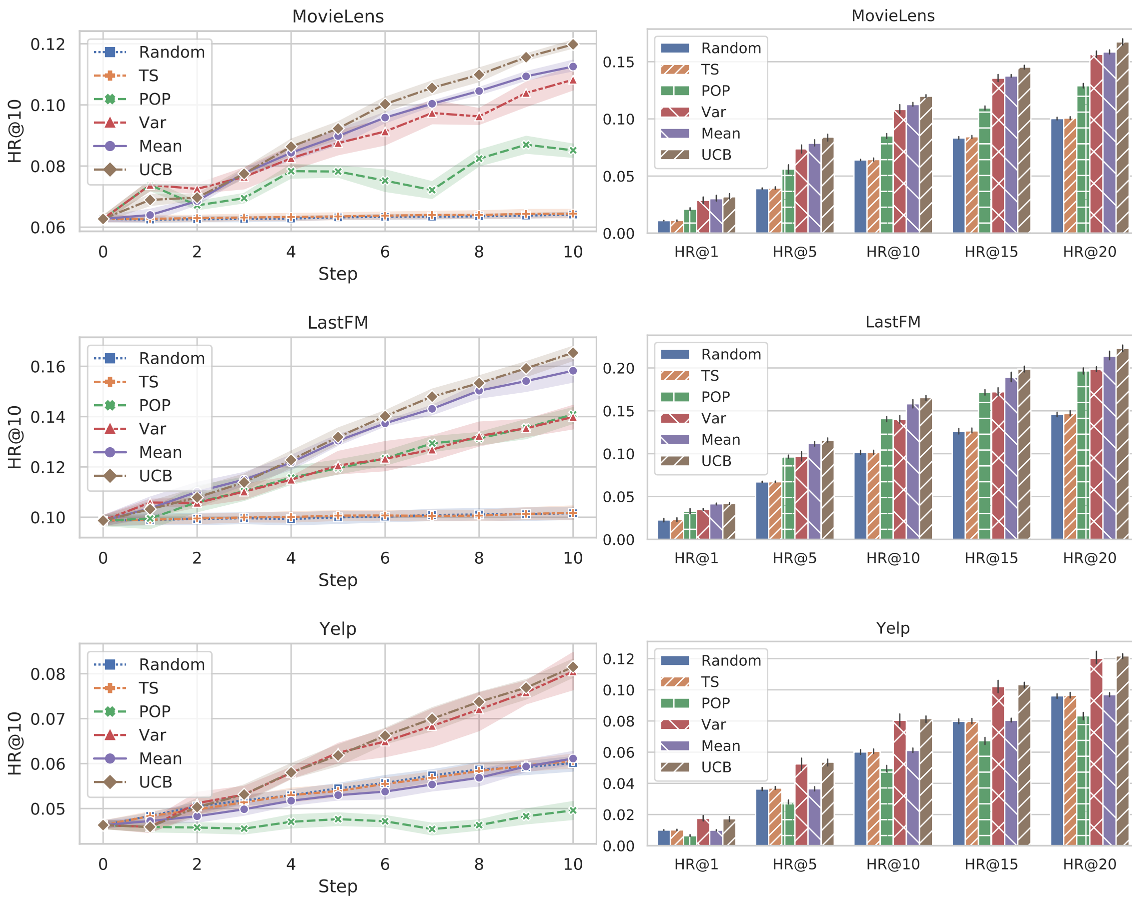

Fig. 3. **Left column**: HR@10 comparison between query selection strategies during the conversation session. **Right column**: Hit Ratio for $k \in [1, 5, 10, 15, 20]$ after 10th step.

Random on account of the inherently high posterior uncertainty in the presence of few keyphrase observations. The greedy exploitation-only strategy of Mean performs moderately well on MovieLens and LastFM (but not Yelp), while the exploration-only Var method performs well on Yelp. One possible hypothesis for the comparable performance of UCB and Var on Yelp is that since there is a well-known bias towards high ratings in Yelp, it is possible that the mean information used by UCB was relatively uninformative (nb. Mean did not outperform Random on Yelp) and hence only the variance leveraged by both methods was informative for exploration. It is also useful to note that the performance of a standard user-item matrix factorization model without any preference elicitation corresponds to step 0 (for all methods).

The right column of Figure 3 shows a performance comparison on HR@k for different k after the 10th interaction. As discussed 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 improve final recommendation performance the most.

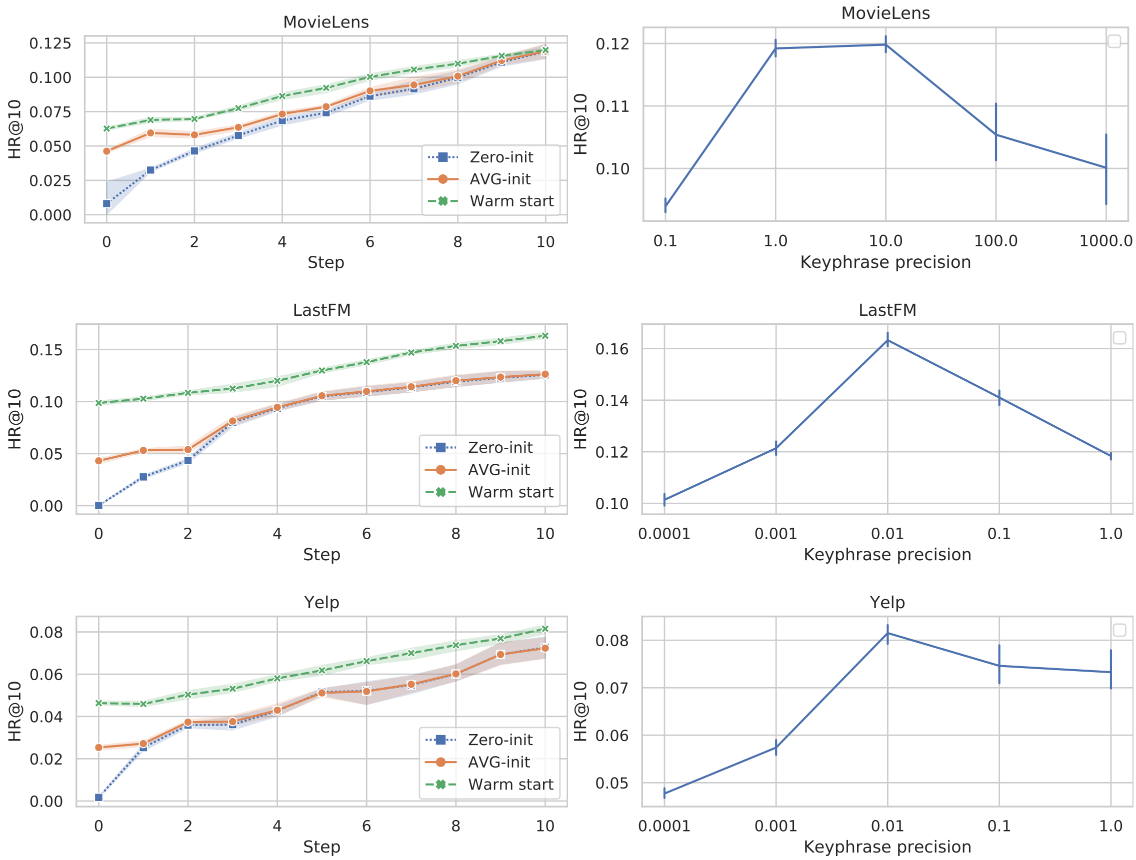

Fig. 4. **Left column**: HR@10 comparison between different prior belief initialization during the conversation session. **Right column**: HR@10 HR@10 comparison with different keyphrase precision β_y values with 95% CI.

5.4 RQ2: Prior Belief Warm-start Initialization

The left column of Figure 4 shows the HR@10 comparison between different warm-start prior initialization strategies during the interactive session using UCB as the query strategy. Warm-start uses an informed prior that reflects the user's long-term item preference history as previously described. The other two methods are non-personalized: Zero-init initializes the prior mean with a zero vector, while the average over all user vectors (the non-personalized "average user") is used as the initial mean in AVG-init. The graph clearly indicates that our proposed warm-start method outperforms non-personalized methods by starting with the most informed recommendation at Step 0; while the warm-start approach dominates for all time steps, the influence of the prior gradually decreases as the interaction between user and system unfolds; this suggests that the UCB query strategy should also be effective for the cold-start setting. In MovieLens, non-personalized approaches show almost the same performance by step 10 while the difference still exists in LastFM (and Yelp, though the difference is relatively small). This indicates that in Yelp and especially LastFM, the user's previous history still plays a significant role whereas preferences in MovieLens may be more ephemeral.

Table 3. Example keyphrase and top-5 similar keyphrases according to cosine distance as learned on each dataset.

Dataset	keyphrase	Top-5 Similar keyphrases (Cosine Similarity)
MovieLens	social commentary	loneliness, depressing, stylized, political, thought-provoking
	conspiracy	thriller, assassination, cia, assassin, spying
	futuristic	sci-fi, future, dystopia, artificial intelligence, dystopic future
	family	children, kids, christmas, holiday, dog
Last.fm	j-pop	japanese, j-rock, jpop, anime, kpop
	country	modern country, country rock, oldies, great song, country pop
	hardcore	metalcore, deathcore, screamo, post-hardcore, emocore
	metal	hard rock, nu metal, heavy metal, alternative rock, trash metal

5.5 RQ3: Tuning Keyphrase Precision for Belief Update

The right column of Figure 4 analyzes how different keyphrase precision $\beta_{\mathcal{Y}}$ values affect elicitation performance (vs. fixed item precision $\beta_{\mathcal{R}} = 0.01$) when using UCB. This plot indicates that tuning $\beta_{\mathcal{Y}}$ is critical for improving the effectiveness of the elicitation process. In MovieLens, it performs best when keyphrase precision is 1,000 times larger than item precision, but for LastFM and Yelp, the best performance is achieved when keyphrase and item precision have the same value. Compared to Yelp, in LastFM, performance decreases sharply as the keyphrase precision increases (or the item precision decreases relatively). This can be interpreted in the same vein as the result of Section 5.4, where previous item history plays a relatively important role in LastFM (and in Yelp, though not as important as in LastFM) compared to MovieLens.

5.6 RQ4: Anecdotal Evaluation of 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 11) 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 and hence should facilitate the generalization of elicited feedback over related keyphrase queries.

6 CONCLUSION

We introduced a novel keyphrase-based Bayesian preference elicitation (PE) framework for interactive (conversational) recommendation that coembeds user, items, and keyphrase descriptions in the same space. It allows efficient, closed-form Bayesian updating over user preferences for both items and keyphrases, thus enabling a variety of *keyphrase*-based Bayesian PE variants based on this underlying framework. Key results show that our proposed model with a UCB query strategy and warm-start prior from previous user history provides strong recommendation performance improvements with each round of keyphrase elicitation.

There are many interesting problems for future exploration in this keyphrase-based Bayesian PE framework. For example, we can investigate improvements to the Bayesian framework itself. While it would require a significant computational overhead in comparison to the current closed-form, efficient Bayesian updating framework, a deep Bayesian framework [3, 29] could provide a more expressive framework for Bayesian modeling of user preferences. In a different vein, we could investigate deeper knowledge-based methods such as knowledge graphs [21, 36] as a means of querying a large semantic variety of natural language feedback (e.g., favorite genres, actors, and directors for movies) and relating them all in a latent embedding space. While such extensions would be very challenging, they could further enhance future natural language PE recommendation interfaces based on the foundational keyphrase-based PE methods proposed in this work.

REFERENCES

[1] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *J. Mach. Learn. Res.* 3 (2002), 397–422. http://jmlr.org/papers/v3/auer02a.html

[2] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. 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*. 265–274. https://doi.org/10.1145/3331184.3331211

[3] Charles Blundell, Julien Cornebise, K. Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *ICML*.

[4] Craig Boutilier. 2002. 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*. 239–246. http://www.aaai.org/Library/AAAI/2002/aaai02-037.php

[5] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1996. Knowledge-based Navigation of Complex Information Spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1* (Portland, Oregon) *(AAAI'96)*. AAAI Press, 462–468.

[6] Urszula Chajewska, Lise Getoor, Joseph Norman, and Yuval Shahar. 2013. Utility Elicitation as a Classification Problem. *CoRR* abs/1301.7367 (2013). arXiv:1301.7367 http://arxiv.org/abs/1301.7367

[7] Urszula Chajewska, Daphne Koller, and Ronald Parr. 2000. Making Rational Decisions Using Adaptive Utility Elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 363–369.

[8] Olivier Chapelle and Lihong Li. 2011. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2249–2257. http://papers.nips.cc/paper/4321-an-empirical-evaluation-of-thompson-sampling.pdf

[9] Li Chen and Pearl Pu. 2004. *Survey of Preference Elicitation Methods.* Technical Report. SWISS FEDERAL INSTITUTE OF TECHNOLOGY IN LAUSANNE EPFL.

[10] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. 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*. 815–824. https://doi.org/10.1145/2939672.2939746

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

[12] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Association for Computational Linguistics, Melbourne, Australia, 2–7. https://doi.org/10.18653/v1/P18-5002

[13] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 140–149. https://doi.org/10.18653/v1/P16-1014

[14] Shengbo Guo and Scott Sanner. 2010. Real-time Multiattribute Bayesian Preference Elicitation with Pairwise Comparison Queries. In *Thirteenth International Conference on Artificial Intelligence and Statistics*.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[16] R.A. Howard and J.E Matheson. 1983. *Readings on the Principles and Applications of Decision Analysis: General collection.* Strategic Decisions Group. https://books.google.ca/books?id=KTy3AAAAIAAJ

[17] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A Ranking Optimization Approach to Latent Linear Critiquing in Conversational Recommender System. In *Proceedings of the 14th International ACM Conference on Recommender Systems (RecSys-20)*. Online.

[18] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 9725–9735.

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

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

[21] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104 (12 2015). https://doi.org/10.1109/JPROC.2015.2483592

[22] Gustavo Penha and C. Hauff. 2020. Challenges in the Evaluation of Conversational Search Systems. In *Converse@KDD*.

[23] Bilih Priyogi. 2019. Preference Elicitation Strategy for Conversational Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 824–825. https://doi.org/10.1145/3289600.3291604

[24] David W. Rajala. 1979. 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 (1979), 403. https://doi.org/10.1109/TSMC.1979.4310245

[25] James Reilly, Kevin McCarthy, Lorraine Mcginty, and Barry Smyth. 2004. Dynamic Critiquing. In *Advances in Case-Based Reasoning, 7th European Conference (ECCBR) 2004*. 37–50. https://doi.org/10.1007/978-3-540-28631-8_55

[26] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2005. Explaining Compound Critiques. *Artif. Intell. Rev.* 24, 2 (Oct. 2005), 199–220.

[27] S. Sedhain, A. Menon, S. Sanner, and L. Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on the World Wide Web (WWW-15)*. Florence, Italy.

[28] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *RecSys 2018: The ACM Conference on Recommender Systems*. ACM, 172–180.

[29] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, N. Sundaram, Md. Mostofa Ali Patwary, Prabhat, and R. Adams. 2015. Scalable Bayesian Optimization Using Deep Neural Networks. In *ICML*.

[30] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, ACM, New York, NY, USA.

[31] David H. Stern, Ralf Herbrich, and Thore Graepel. 2009. Matchbox: large scale online Bayesian recommendations. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web* (Madrid, Spain). ACM, New York, NY, USA, 111–120. https://doi.org/10.1145/1526709.1526725

[32] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. *CoRR* abs/1806.03277 (2018). arXiv:1806.03277 http://arxiv.org/abs/1806.03277

[33] Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. 2019. Gradient-based Optimization for Bayesian Preference Elicitation. *CoRR* abs/1911.09153 (2019). arXiv:1911.09153 http://arxiv.org/abs/1911.09153

[34] Paolo Viappiani and Craig Boutilier. 2009. 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*. 101–108. https://doi.org/10.1145/1639714.1639732

[35] Paolo Viappiani and Craig Boutilier. 2010. Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets. In *Advances in Neural Information Processing Systems 23 (NIPS)*. MIT press. http://www.cs.toronto.edu/~paolo/papers/bayesian-opt-sets_LONG.pdf

[36] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* PP (09 2017), 1–1. https://doi.org/10.1109/TKDE.2017.2754499

[37] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, and Changyou Chen. 2019. Text-Based Interactive Recommendation via Constraint-Augmented Reinforcement Learning. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 15214–15224. http://papers.nips.cc/paper/9657-text-based-interactive-recommendation-via-constraint-augmented-reinforcement-learning.pdf

[38] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1512–1520. https://doi.org/10.1145/3394486.3403202

[39] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey O. Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. 2018. 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*. 446–456. http://auai.org/uai2018/proceedings/papers/179.pdf