Efficient Explorative Key-Term Selection Strategies for Conversational Contextual Bandits

Zhiyong Wang[1], **Xutong Liu**[1], **Shuai Li**[2*], **John C.S. Lui**[1]

[1]The Chinese University of Hong Kong ,Hong Kong SAR, China
[2]Shanghai Jiao Tong University, Shanghai, China
zywang21@cse.cuhk.edu.hk, liuxt@cse.cuhk.edu.hk, shuaili8@sjtu.edu.cn, cslui@cse.cuhk.edu.hk

Abstract

Conversational contextual bandits elicit user preferences by occasionally querying for explicit feedback on key-terms to accelerate learning. However, there are aspects of existing approaches which limit their performance. First, information gained from key-term-level conversations and arm-level recommendations is not appropriately incorporated to speed up learning. Second, it is important to ask explorative key-terms to quickly elicit the user's potential interests in various domains to accelerate the convergence of user preference estimation, which has never been considered in existing works. To tackle these issues, we first propose "ConLinUCB", a general framework for conversational bandits with better information incorporation, combining arm-level and key-term-level feedback to estimate user preference in one step at each time. Based on this framework, we further design two bandit algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConLinUCB-MCR. We prove tighter regret upper bounds of our proposed algorithms. Particularly, ConLinUCB-BS achieves a regret bound of $O(\sqrt{dT\log T})$, better than the previous result $O(d\sqrt{T}\log T)$. Extensive experiments on synthetic and real-world data show significant advantages of our algorithms in learning accuracy (up to 54% improvement) and computational efficiency (up to 72% improvement), compared to the classic ConUCB algorithm, showing the potential benefit to recommender systems.

1 Introduction

Nowadays, recommender systems are widely used in various areas. The learning speed for traditional online recommender systems is usually slow since extensive exploration is needed to discover user preferences. To accelerate the learning process and provide more personalized recommendations, the conversational recommender system (CRS) has been proposed (Christakopoulou, Radlinski, and Hofmann 2016; Christakopoulou et al. 2018; Sun and Zhang 2018; Zhang et al. 2018; Li et al. 2021; Gao et al. 2021). In a CRS, a learning agent occasionally asks for the user's explicit feedback on some "key-terms", and leverages this additional conversational information to better elicit the user's preferences (Zhang et al. 2020; Xie et al. 2021).

Despite the recent success of CRS, there are crucial limitations in using conversational contextual bandit approaches to design recommender systems. These limitations include: (a) The information gained from key-term-level conversations and arm-level recommendations is not incorporated properly to speed up learning, as the user preferences are essentially assumed to be the same in these two stages but are estimated separately (Zhang et al. 2020; Xie et al. 2021; Wu et al. 2021); (b) Queries using traditional key-terms were restrictive and not explorative enough. Specifically, we say a key-term is "explorative" if it is under-explored so far and the system is uncertain about the user's preferences in its associated items. Asking for the user's feedback on explorative key-terms can efficiently elicit her potential interests in various domains (e.g., sports, science), which means we can quickly estimate the user preference vector in all directions of the feature space, thus accelerating the learning speed. Therefore, it is crucial to design explorative key-term selection strategies, which existing works have not considered.

Motivated by the above considerations, we propose to design conversational bandit algorithms that (i) estimate the user's preferences utilizing both arm-level and key-term-level interactions simultaneously to properly incorporate the information gained from both two levels and (ii) use effective strategies to choose explorative key-terms when conducting conversations for quick user preference inference.

To better utilize the interactive feedback from both recommendations and conversations, we propose ConLinUCB, a general framework for conversational bandits with possible flexible key-term selection strategies. ConLinUCB estimates the user preference vector by solving *one single optimization problem* that minimizes the mean squared error of both arm-level estimated rewards and key-term-level estimated feedback simultaneously, instead of separately estimating at different levels as in previous works. In this manner, the information gathered from these two levels can be better combined to guide the learning.

Based on this ConLinUCB framework, we design two new algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConLinUCB-MCR.

- ConLinUCB-BS makes use of a barycentric spanner containing linearly independent vectors, which can be an efficient exploration basis in bandit problems (Amballa, Gupta, and Bhat 2021). Whenever a conversation is al-

[*]Corresponding author.

lowed, ConLinUCB-BS selects an explorative key-term uniformly at random from a precomputed barycentric spanner $\mathcal{B}$ of the given key-term set $\mathcal{K}$.

- ConLinUCB-MCR applies in a more general setting when the key-term set can be time-varying, and it can leverage interactive histories to choose explorative key-terms adaptively. Note that in the bandit setting, we often use confidence radius to adaptively evaluate whether an arm has been sufficiently explored, and the confidence radius of an arm will shrink whenever it is selected (Lattimore and Szepesvári 2020). This implies that an explorative key-term should have a large confidence radius. Based on this reasoning, ConLinUCB-MCR selects the most explorative key-terms with maximal confidence radius when conducting conversations.

Equipped with explorative conversations, our algorithms can quickly elicit user preferences for better recommendations. For example, if the key-term *sports* is explorative at round t, indicating that so far the agent is not sure whether the user favors items associated with *sports* (e.g., basketball, volleyball), it will ask for the user's feedback on *sports* directly and conduct recommendations accordingly. In this manner, the agent can quickly find suitable items for the user. We prove the regret upper bounds of our algorithms, which are better than the classic ConUCB algorithm.

In summary, our paper makes the following contributions:

- We propose a new and general framework for conversational contextual bandits, ConLinUCB, which can efficiently incorporate the interactive information gained from both recommendations and conversations.
- Based on ConLinUCB, we design two new algorithms with explorative key-term selection strategies, ConLinUCB-BS and ConUCB-MCR, which can accelerate the convergence of user preference estimation.
- We prove that our algorithms achieve tight regret upper bounds. Particularly, ConLinUCB-BS achieves a bound of $O(\sqrt{dT\log T})$, significantly better than the previous $O(d\sqrt{T}\log T)$ in the conversational bandits literature.
- Experiments on both synthetic and real-world data validate the advantages of our algorithms in both learning accuracy (up to 54% improvement) and computational efficiency (up to 72% improvement)[1].

2 Related Work

Our work is most closely related to the research on conversational contextual bandits.

Contextual linear bandit is an online sequential decision-making problem where at each time step, the agent has to choose an action and receives a corresponding reward whose expected value is an unknown linear function of the action (Li et al. 2010; Chu et al. 2011; Abbasi-Yadkori, Pál, and Szepesvári 2011; Wu et al. 2016). The objective is to collect as much reward as possible in T rounds.

Traditional linear bandits need extensive exploration to capture the user preferences. To speed up online recommendations, the idea of conversational contextual bandits was first proposed in (Zhang et al. 2020), where conversational feedback on key-terms is leveraged to assist the user preference elicitation. They propose the ConUCB algorithm with a theoretical regret bound of $O(d\sqrt{T}\log T)$. Some follow-up works try to improve the performance of ConUCB with the help of additional information, such as self-generated key-terms (Wu et al. 2021), relative feedback (Xie et al. 2021), and knowledge graph (Zhao et al. 2022). Unlike these works, we adopt the same problem settings as ConUCB and improve the underlying mechanisms without relying on additional information. Yet one can use the principles of efficient information incorporation and explorative conversations proposed in this work to enhance these works when additional information is available, which is left as an interesting future work.

3 Problem Settings

This section states the problem setting of conversational contextual bandits. Suppose there is a finite set $\mathcal{A}$ of arms. Each arm $a \in \mathcal{A}$ represents an item to be recommended and is associated with a feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$. Without loss of generality, the feature vectors are assumed to be normalized, i.e., $\|\boldsymbol{x}_a\|_2 = 1, \forall a \in \mathcal{A}$. The agent interacts with a user in $T \in \mathbb{N}_+$ rounds, whose preference of items is represented by an *unknown* vector $\boldsymbol{\theta}^* \in \mathbb{R}^d$, $\|\boldsymbol{\theta}^*\|_2 \leq 1$.

At each round $t = 1, 2, ..., T$, a subset of arms $\mathcal{A}_t \subseteq \mathcal{A}$ are available to the agent to choose from. Based on historical interactions, the agent selects an arm $a_t \in \mathcal{A}_t$, and receives a corresponding reward $r_{a_t,t} \in [0, 1]$. The reward is assumed to be a linear function of the contextual vectors

$$r_{a_t,t} = \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}^* + \epsilon_t\,, \tag{1}$$

where ϵ_t is 1-sub-Gaussian random noise with zero mean.

Let $a_t^* \in \arg\max_{a\in\mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}^*$ denote an optimal arm at t. The learning objective is to minimize the cumulative regret

$$R(T) = \sum_{t=1}^{T} \boldsymbol{x}_{a_t^*}^\top \boldsymbol{\theta}^* - \sum_{t=1}^{T} \boldsymbol{x}_{a_t}^\top \boldsymbol{\theta}^*. \tag{2}$$

The agent can also occasionally query the user's feedback on some conversational key-terms to help elicit user preferences. In particular, a "key-term" is a keyword or topic related to a subset of arms. For example, the key-term *sports* is related to the arms like basketball, football, swimming, running, etc.

Suppose there is a finite set $\mathcal{K}$ of key-terms. The relationship between arms and key-terms is given by a weighted bipartite graph $(\mathcal{A}, \mathcal{K}, \boldsymbol{W})$, where $\boldsymbol{W} \triangleq [w_{a,k}]_{a\in\mathcal{A},k\in\mathcal{K}}$ represents the relationship between arms and key-terms, i.e., a key-term $k \in \mathcal{K}$ is associated to an arm $a \in \mathcal{A}$ with weight $w_{a,k} \geq 0$. We assume that each key-term k has positive weights with some related arms (i.e., $\sum_{a\in\mathcal{A}} w_{a,k} > 0$, $\forall k \in \mathcal{K}$), and the weights associated with each arm sum up to 1, i.e., $\sum_{k\in\mathcal{K}} w_{a,k} = 1$, $a \in \mathcal{A}$. The feature vector of

[1] Codes are available at https://github.com/ZhiyongWangWzy/ConLinUCB.

a key-term k is given by $\tilde{\boldsymbol{x}}_k = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The key-term-level feedback on the key-term k at t is defined as

$$\tilde{r}_{k,t} = \tilde{\boldsymbol{x}}_k^\top \boldsymbol{\theta}^* + \tilde{\epsilon}_t \, , \tag{3}$$

where $\tilde{\epsilon}_t$ is assumed to be 1-sub-Gaussian random noise. One thing to stress is that in the previous works (Zhang et al. 2020; Wu et al. 2021; Xie et al. 2021; Zhao et al. 2022), the *unknown* user preference vector $\boldsymbol{\theta}^*$ is essentially assumed to be the same at both the arm level and the key-term level.

To avoid affecting the user experience, the agent should not conduct conversations too frequently. Following (Zhang et al. 2020), we define a function $b : \mathbb{N}_+ \to \mathbb{R}_+$, where $b(t)$ is increasing in t, to control the conversation frequency of the agent. At each round t, if $b(t) - b(t-1) > 0$, the agent is allowed to conduct $q(t) = \lfloor b(t) - b(t-1) \rfloor$ conversations by asking for user's feedback on $q(t)$ key-terms. Using this modeling arrangement, the agent will have $b(t)$ conversational interactions with the user up to round t.

4 Algorithms and Theoretical Analysis

This section first introduces ConLinUCB, a framework for conversational bandits with better information incorporation, which is general for "any" key-term selection strategies. Based on ConLinUCB, we further propose two bandit algorithms, ConLinUCB-BS and ConLinUCB-MCR, with explorative key-term selection strategies.

To simplify the exposition, we merge the ConLinUCB framework, ConLinUCB-BS and ConLinUCB-MCR in Algorithm 1. We also theoretically give regret bounds of our proposed algorithms.

4.1 General ConLinUCB Algorithm Framework

In conversational bandits, it is common that the *unknown* preference vector $\boldsymbol{\theta}^*$ is essentially assumed to be the same at both arm level and key-term level (Zhang et al. 2020; Xie et al. 2021; Wu et al. 2021). However, all existing works treat $\boldsymbol{\theta}^*$ differently at these two levels. Specifically, they take two different steps to estimate user preference vectors at the arm level and key-term level, and use a discounting parameter $\lambda \in (0, 1)$ to balance learning from these two levels' interactions. In this manner, the contributions of the arm-level and key-term-level information to the convergence of estimation are discounted by λ and $1 - \lambda$, respectively. Therefore, such discounting will cause waste of observations, indicating that information at these two levels can not be fully leveraged to accelerate the learning process.

Algorithm 1: General ConLinUCB framework

```
Input: graph($(\mathcal{A}, \mathcal{K}, \boldsymbol{W})$), conversation frequency
       function $b(t)$, key-term selection strategy $\boldsymbol{\pi}$.
Initialization: $\boldsymbol{M}_0 = \beta \boldsymbol{I}$, $\boldsymbol{b}_0 = \boldsymbol{0}$.
for $t$ = 1 to $T$ do
    if $b(t) - b(t-1) > 0$ then
        $q(t) = \lfloor b(t) - b(t-1) \rfloor$;
        while $q(t) > 0$ do
            Select a key-term $k \in \mathcal{K}$ using the
              specified key-term selection strategy $\boldsymbol{\pi}$
              (e.g., Eq. (8) for ConLinUCB-BS and
              Eq. (9) for ConLinUCB-MCR), and
              query the user's preference over it;
            Receive the user's feedback $\tilde{r}_{k,t}$;
            $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top$;
            $\boldsymbol{b}_t = \boldsymbol{b}_{t-1} + \tilde{\boldsymbol{x}}_k \tilde{r}_{k,t}$;
            $q(t) -= 1$;
        end
    else
        $\boldsymbol{M}_t = \boldsymbol{M}_{t-1}$, $\boldsymbol{b}_t = \boldsymbol{b}_{t-1}$;
    end
    $\boldsymbol{\theta}_t = \boldsymbol{M}_t^{-1} \boldsymbol{b}_t$;
    Select $a_t = \arg\max_{a \in \mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}_t + \alpha_t \|\boldsymbol{x}_a\|_{\boldsymbol{M}_t^{-1}}$;
    Ask the user's preference on arm $a_t$ and receive
      the reward $r_{a_t,t}$ ;
    $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \boldsymbol{x}_{a_t} \boldsymbol{x}_{a_t}^\top$;
    $\boldsymbol{b}_t = \boldsymbol{b}_{t-1} + \boldsymbol{x}_{a_t} r_{a_t,t}$;
end
```

To handle the above issues, we propose a general framework ConLinUCB, for conversational contextual bandits. In this framework, in order to fully leverage interactive information from two levels, we simultaneously estimate the user preference vector by solving *one single optimization problem* that minimizes the mean squared error of both arm-level estimated rewards and key-term-level estimated feedback. Specifically, in ConLinUCB, at round t, the user preference vector is estimated by solving the following linear regression

$$\begin{aligned} \boldsymbol{\theta}_t = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} & \sum_{\tau=1}^{t-1} (\boldsymbol{x}_{a_\tau}^\top \boldsymbol{\theta} - r_{a_\tau,\tau})^2 \\ & + \sum_{\tau=1}^{t} \sum_{k \in \mathcal{K}_\tau} (\tilde{\boldsymbol{x}}_k^\top \boldsymbol{\theta} - \tilde{r}_{k,\tau})^2 + \beta \|\boldsymbol{\theta}\|_2^2 \, , \end{aligned} \tag{4}$$

where $\mathcal{K}_\tau$ denotes the set of key-terms asked at round τ, and the coefficient $\beta > 0$ controls regularization. The closed-form solution of this optimization problem is $\boldsymbol{\theta}_t = \boldsymbol{M}_t^{-1} \boldsymbol{b}_t$, where

$$\begin{aligned} \boldsymbol{M}_t &= \sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau} \boldsymbol{x}_{a_\tau}^\top + \sum_{\tau=1}^{t} \sum_{k \in \mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top + \beta \boldsymbol{I} \, , \\ \boldsymbol{b}_t &= \sum_{\tau=1}^{t-1} \boldsymbol{x}_{a_\tau} r_{a_\tau,\tau} + \sum_{\tau=1}^{t} \sum_{k \in \mathcal{K}_\tau} \tilde{\boldsymbol{x}}_k \tilde{r}_{k,\tau} . \end{aligned} \tag{5}$$

To balance exploration and exploitation, ConLinUCB selects arms using the upper confidence bound (UCB) strategy

$$a_t = \arg\max_{a \in \mathcal{A}_t} \boldsymbol{x}_a^\top \boldsymbol{\theta}_t + \alpha_t \|\boldsymbol{x}_a\|_{\boldsymbol{M}_t^{-1}} \triangleq \hat{R}_{a,t} + C_{a,t} \, , \tag{6}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{M}} = \sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$, $\hat{R}_{a,t}$ and $C_{a,t}$ denote the estimated reward and confidence radius of arm a at round t, and

$$\alpha_t = \sqrt{2\log(\frac{1}{\delta}) + d\log(1 + \frac{t + b(t)}{\beta d}) + \sqrt{\beta}}\,, \quad (7)$$

which comes from the following Lemma 1.

The ConLinUCB algorithm framework is shown in Alg. 1. The key-term-level interactions take place in line 3-14. At round t, the agent first determines whether conversations are allowed using $b(t)$. When conducting conversations, the agent asks for the user's feedback on $q(t)$ key-terms and uses the feedback to update the parameters. Line 15-20 summarise the arm-level interactions. Based on historical interactions, the agent calculates the estimated $\boldsymbol{\theta}^*$, selects an arm with the largest UCB index, receives the corresponding reward, and updates the parameters accordingly. ConLinUCB only maintains one set of covariance matrix $\boldsymbol{M}_t$ and regressand vector $\boldsymbol{b}_t$, containing the feedback from both arm-level and key-term-level interactions. By doing so, ConLinUCB better leverages the feedback information than ConUCB. Note that ConLinUCB is a general framework with the specified key-term selection strategy $\boldsymbol{\pi}$ to be determined.

4.2 ConLinUCB with Key-Terms from Barycentric Spanner (ConLinUCB-BS)

Based on the ConLinUCB framework, we propose the ConLinUCB-BS algorithm with an explorative key-term selection strategy. Specifically, ConLinUCB-BS selects key-terms from the barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$, which is an efficient exploration basis in online learning (Amballa, Gupta, and Bhat 2021), to conduct explorative conversations. Below is the formal definition of the barycentric spanner for the key-term set $\mathcal{K}$.

Definition 1 (Barycentric Spanner of $\mathcal{K}$) *A subset $\mathcal{B} = \{k_1, k_2, ..., k_d\} \subseteq \mathcal{K}$ is a barycentric spanner for $\mathcal{K}$ if for any $k \in \mathcal{K}$, there exists a set of coefficients $\boldsymbol{c} \in [-1, 1]^d$, such that $\tilde{\boldsymbol{x}}_k = \sum_{i=1}^{d} \boldsymbol{c}_i \tilde{\boldsymbol{x}}_{k_i}$.*

We assume that the key-term set $\mathcal{K}$ is finite and $\{\tilde{\boldsymbol{x}}_k\}_{k\in\mathcal{K}}$ span $\mathbb{R}^d$, thus the existence of a barycentric spanner $\mathcal{B}$ of $\mathcal{K}$ is guaranteed (Awerbuch and Kleinberg 2008).

Corresponding vectors in the barycentric spanner are linearly independent. By choosing key-terms from the barycentric spanner, we can quickly explore the *unknown* user preference vector $\boldsymbol{\theta}^*$ in various directions. Based on this reasoning, whenever a conversation is allowed, ConLinUCB-BS selects a key-term

$$k \sim \text{unif}(\mathcal{B}), \quad (8)$$

which means sampling a key-term k uniformly at random from the barycentric spanner $\mathcal{B}$ of $\mathcal{K}$. ConLinUCB-BS is completed using the above strategy as $\boldsymbol{\pi}$ in the ConLinUCB framework (Alg. 1). As shown in the following Lemma 1 and Lemma 2, in ConLinUCB-BS, the statistical estimation uncertainty shrinks quickly. Additionally, since the barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$ can be precomputed offline, ConLinUCB-BS is computationally efficient, which is vital for real-time recommendations.

4.3 ConLinUCB with Key-Terms Having Max Confidence Radius (ConLinUCB-MCR)

We can further improve ConLinUCB-BS in the following aspects. First, ConLinUCB-BS does not apply in a more general setting where the key-term set $\mathcal{K}$ varies over time since it needs a precomputed barycentric spanner $\mathcal{B}$ of $\mathcal{K}$. Second, as the selection of key-terms is independent of past observations, ConLinUCB-BS does not fully leverage the historical information. For example, suppose the agent is already certain about whether the user favors *sports* based on previous interactions. In that case, it does not need to ask for the user's feedback on the key-term *sports* anymore. To address these issues, we propose the ConLinUCB-MCR algorithm that (i) is applicable when the key-term set $\mathcal{K}$ varies with t and (ii) can adaptively conduct explorative conversations based on historical interactions.

In multi-armed bandits, confidence radius is used to capture whether an arm has been well explored in the interactive history, and it will shrink whenever the arm is selected. Motivated by this, if a key-term has a large confidence radius, it means the system has not sufficiently explored the user's preferences in its related items, indicating that this key-term is explorative. Based on this reasoning, ConLinUCB-MCR selects key-terms with maximal confidence radius to conduct explorative conversations adaptively. Specifically, when a conversation is allowed at t, ConLinUCB-MCR chooses a key-term as follow

$$k \in \arg\max_{k\in\mathcal{K}_t} \alpha_t \left\|\tilde{\boldsymbol{x}}_k\right\|_{\boldsymbol{M}_t^{-1}}\,, \quad (9)$$

where α_t is defined in Eq. (7) and $\mathcal{K}_t \subseteq \mathcal{K}$ denotes the possibly time-varying key-terms set available at round t. ConLinUCB-MCR is completed using the above strategy (Eq. (9)) as $\boldsymbol{\pi}$ in ConLinUCB (Alg. 1).

4.4 Theoretical Analysis

We give upper bounds of the regret for our algorithms. As a convention, the conversation frequency satisfies $b(t) \leq t$, so we assume $b(t) = b \cdot t$, $b \in (0, 1)$. We leave the proofs of Lemma 1-2 and Theorem 3-4 to the Appendix due to the space limitation.

The following lemma shows a high probability upper bound of the difference between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}^*$ in the direction of the action vector $\boldsymbol{x}_a$ for algorithms based on ConLinUCB.

Lemma 1 *At $\forall t$, for any $a \in \mathcal{A}$, with probability at least $1 - \delta$ for some $\delta \in (0, 1)$*

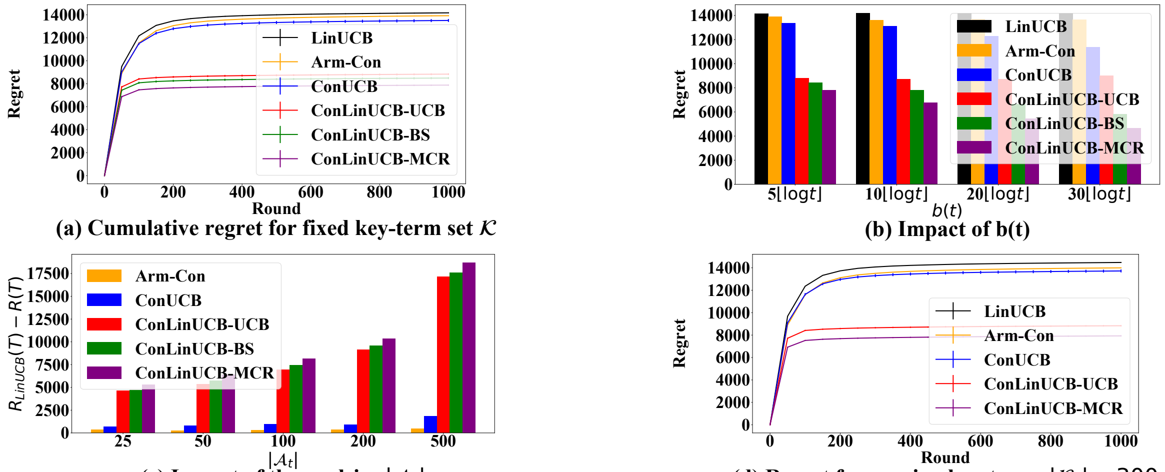
$$\left|\boldsymbol{x}_a^\top(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)\right| \leq \alpha_t \left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}} = C_{a,t},$$

where $\alpha_t = \sqrt{2\log(\frac{1}{\delta}) + d\log(1 + \frac{t+b(t)}{\beta d}) + \sqrt{\beta}}$.

For a barycentric spanner $\mathcal{B}$ of the key-term set $\mathcal{K}$, let

$$\lambda_{\mathcal{B}} := \lambda_{\min}(\boldsymbol{E}_{k\sim\text{unif}(\mathcal{B})}[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^\top]) > 0\,, \quad (10)$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the augment. We can get the following Lemma that gives a high probability upper bound of $\left\|\boldsymbol{x}_a\right\|_{\boldsymbol{M}_t^{-1}}$ for ConLinUCB-BS.



(a) Cumulative regret for fixed key-term set $\mathcal{K}$

(b) Impact of b(t)

(c) Impact of the poolsize $|\mathcal{A}_t|$

(d) Regret for varying key-terms $|\mathcal{K}_t| = 300$

Figure 1: Experimental results on synthetic dataset

Lemma 2 *For ConLinUCB-BS, $\forall a \in \mathcal{A}$, at $\forall t \geq t_0 = \frac{256}{b\lambda_{\mathcal{B}}^2}\log(\frac{128d}{\lambda_{\mathcal{B}}^2\delta})$, with probability at least $1-\delta$ for $\delta \in (0, \frac{1}{8}]$*

$$\|\boldsymbol{x}_a\|_{\boldsymbol{M}_t^{-1}} \leq \sqrt{\frac{2}{\lambda_{\mathcal{B}} bt}}\,.$$

The following theorem gives a high probability regret upper bound of our ConLinUCB-BS.

Theorem 3 *With probability at least $1-\delta$ for some $\delta \in (0, \frac{1}{4}]$, the regret $R(T)$ of ConLinUCB-BS satisfies*

$$R(T) \leq 4\sqrt{\frac{2}{b\lambda_{\mathcal{B}}}}\sqrt{T}\left(\sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{(1+b)T}{\beta d})} + \sqrt{\beta}\right) + \frac{256}{b\lambda_{\mathcal{B}}^2}\log(\frac{256d}{\lambda_{\mathcal{B}}^2\delta}) + 1\,.$$

Recall the regret bound of ConUCB (Zhang et al. 2020) is

$$R(T) \leq 2\sqrt{2Td\log(1+\frac{\lambda(T+1)}{(1-\lambda)d})}\left(\sqrt{\frac{1-\lambda}{\lambda}} + \sqrt{\frac{1-\lambda}{\lambda\beta}}\sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{bT}{\beta d})} + \sqrt{2\log(\frac{2}{\delta}) + d\log(1+\frac{\lambda T}{(1-\lambda)d})}\right),$$

which is of $O(d\sqrt{T}\log T)$. The regret bound of ConLinUCB-BS given in Theorem 3 is of $O(\sqrt{dT\log T})$, much better than ConUCB by reducing a multiplicative $\sqrt{d\log T}$ term. Since d and $\log T$ are usually large in practice, this is a significant improvement.

Next, the following theorem gives a high-probability regret upper bound of ConLinUCB-MCR.

Theorem 4 *With probability at least $1-\delta$ for some $\delta \in (0, 1)$, the regret $R(T)$ of ConLinUCB-MCR satisfies*

$$R(T) \leq 2\sqrt{2Td\log(1+\frac{T+1}{\beta d})} \times \left(\sqrt{\beta} + \sqrt{2\log(\frac{1}{\delta}) + d\log(1+\frac{(b+1)T}{\beta d})}\right).$$

Note that the regret upper bound of ConLinUCB-MCR is smaller than ConUCB by reducing some additive terms.

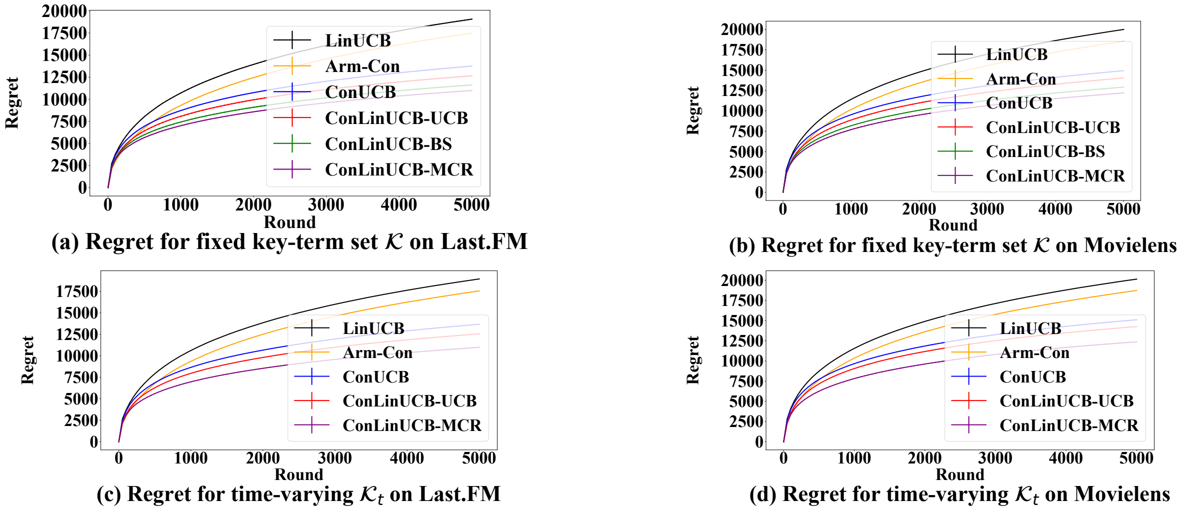
5 Experiments on Synthetic Dataset

In this section, we show the experimental results on synthetic data. To obtain the offline-precomputed barycentric spanner $\mathcal{B}$, we use the method proposed in (Awerbuch and Kleinberg 2008).

5.1 Experimental Settings

Generation of the Synthetic Dataset. We create a set of arms $\mathcal{A}$ with $|\mathcal{A}| = 5,000$ arms, and a set of key-terms $\mathcal{K}$ with $|\mathcal{K}| = 500$. We set the dimension of the feature space to be $d = 50$ and the number of users $N_u = 200$.

For each user preference vector $\boldsymbol{\theta}_u^*$ and each arm feature vector $\boldsymbol{x}_a$, each entry is generated by independently drawing from the standard normal distribution $\mathcal{N}(0, 1)$, and all these vectors are normalized such that $\|\boldsymbol{\theta}_u^*\|_2 = 1$, $\|\boldsymbol{x}_a\|_2 = 1$. The weight matrix $\boldsymbol{W} \triangleq [w_{a,k}]$ is generated as follows: First, for each key-term k, we select an integer $n_k \in [1, 10]$ uniformly at random, then randomly select a subset of n_k arms $\mathcal{A}_k$ to be the related arms for key-term k; second, for each arm a, if it is related to a set of n_a key-terms $\mathcal{K}_a$, we assign equal weights $w_{a,k} = \frac{1}{n_a}, \forall k \in \mathcal{K}_a$. Following (Zhang et al. 2020), the feature vector for each key-term k is computed using $\tilde{\boldsymbol{x}}_k = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The arm-level rewards and key-term-level feedback are generated following Eq. (1) and Eq. (3).



(a) Regret for fixed key-term set $\mathcal{K}$ on Last.FM

(b) Regret for fixed key-term set $\mathcal{K}$ on Movielens

(c) Regret for time-varying $\mathcal{K}_t$ on Last.FM

(d) Regret for time-varying $\mathcal{K}_t$ on Movielens

Figure 2: Experimental results on real-word datasets

Baselines. We compare with the following baselines:

- LinUCB (Li et al. 2010): A state-of-the-art contextual linear bandit algorithm that selects arms only based on the arm-level feedback without conversations.
- Arm-Con (Christakopoulou, Radlinski, and Hofmann 2016): A conversational bandit algorithm that conducts conversations on arms without considering key-terms, and uses LinUCB for arm selection.
- ConUCB (Zhang et al. 2020): The core conversational bandit algorithm that selects a key-term to minimize some estimation error when a conversation is allowed.
- ConLinUCB-UCB: An algorithm using a LinUCB-alike method as the key-term selection strategy in our proposed ConLinUCB framework, i.e., choose key-term $k \in \arg\max_{k \in \mathcal{K}_t} \tilde{\boldsymbol{x}}_k^\top \boldsymbol{\theta}_t + \alpha_t \left\| \tilde{\boldsymbol{x}}_k \right\|_{\boldsymbol{M}_t^{-1}}$ at round t.

5.2 Evaluation Results

This section first shows the results when the key-term set $\mathcal{K}$ is fixed. In this case, we evaluate the regret $R(T)$ for all algorithms, and we study the impact of the conversation frequency function $b(t)$ and the number of arms $|\mathcal{A}_t|$ available at each round t. When $\mathcal{K}$ varies with time, ConLinUCB-BS does not apply, and we compare the regret of other algorithms. Following (Zhang et al. 2020), we set $T = 1,000$, $b(t) = 5\lfloor \log(t+1) \rfloor$ and $|\mathcal{A}_t| = 50$, unless otherwise stated.

Cumulative Regret We run the experiments 10 times and calculate the average regret of all the users for each algorithm. We include $\pm std$ as the error bar, where std stands for the *standard deviation*. The results are given in Figure 1 (a). First, all other algorithms outperform LinUCB, showing the advantage of conversations. Further, with our proposed ConLinUCB framework, even if we use ConLinUCB-UCB with a simple LinUCB-alike key-term selection strategy, the performance is already better than ConUCB (34.91% improvement), showing more efficient information incorporation. With explorative conversations, ConLinUCB-BS and ConLinUCB-MCR achieve much lower regrets (37.00% and 43.10% improvement over ConUCB respectively), indicating better learning accuracy. ConLinUCB-MCR further leverages historical information to conduct explorative conversations adaptively, thus achieving the lowest regret.

Impact of Conversation Frequency Function $b(t)$ A larger $b(t)$ means the agent can conduct more conversations. We set $b(t) = f_q \cdot \lfloor \log t \rfloor$ and vary f_q to change the conversation frequencies, i.e., $f_q \in \{5, 10, 20, 30\}$. The results are shown in Figure 1 (b). With larger $b(t)$, our algorithms have less regret, showing the power of conversations. In all cases, ConLinUCB-BS and ConLinUCB-MCR have lower regrets than ConUCB, and ConLinUCB-MCR performs the best.

Impact of $|\mathcal{A}_t|$ We vary $|\mathcal{A}_t|$ to be 25, 50, 100, 200, 500. To clearly show the advantage of our algorithms, we evaluate the difference in regrets between LinUCB and other algorithms, i.e., $R_{\text{LinUCB}}(T) - R(T)$, representing the improved accuracy of the conversational bandit algorithms as compared with LinUCB. Note that the larger $|\mathcal{A}_t|$ is, the harder it is for the algorithm to identify the best arm. Results in Figure 1 (c) show that as $|\mathcal{A}_t|$ increases, the advantages of ConLinUCB-BS and ConLinUCB-MCR become more significant. Particularly, when $|\mathcal{A}_t|$=25, ConLinUCB-BS and ConLinUCB-MCR achieve 34.99% and 40.21% improvement over ConUCB respectively; when $|\mathcal{A}_t|$=500, ConLinUCB-BS and ConLinUCB-MCR achieve 50.36% and 53.77% improvement over ConUCB, respectively. In real applications, the size of arm set $|\mathcal{A}_t|$ is usually very large. Therefore, our proposed algorithms are expected to significantly outperform ConUCB in practice.

Cumulative Regret for Time-Varying $\mathcal{K}$ This section studies the case when only a subset of key-terms $\mathcal{K}_t \subseteq \mathcal{K}$ are available to the agent at each round t, where ConLinUCB-BS is not applicable as mentioned before. The number of key-terms available at each time t is set to be $|\mathcal{K}_t| = 300$. At round t, 300 key-terms are chosen uniformly at random

Alogrithm	Total time	Total time for selecting arms	Total time for selecting key-terms
ConUCB	11,297	5,217	6,080
ConLinUCB-UCB	5,738	3,060	2,678
ConLinUCB-MCR	4,821	3,030	1,791
ConLinUCB-BS	3,127	3,120	6

Table 1: Total runninng time (in seconds) of algorithms on Movielens with $T = 5,000$.

from $\mathcal{K}$ to form $\mathcal{K}_t$. We evaluate the regret of all algorithms except ConLinUCB-BS. The results are shown in Figure 1 (d). We can observe that ConLinUCB-MCR outperforms all baselines and achieves 43.02% improvement over ConUCB.

6 Experiments on Real-World Datasets

This section shows the experimental results on two real-world datasets, Last.FM and Movielens. The baselines, generations of arm-level rewards and key-term-level feedback, and the computation method of the barycentric spanner are the same as in the last section. We set $T = 5,000$, $b(t) = 5\lfloor\log(t + 1)\rfloor$ and $|\mathcal{A}_t| = 50$, unless otherwise stated.

6.1 Experiment Settings

Last.FM and Movielens Datasets Last.FM is a dataset for artist recommendations containing 186,479 interaction records between 1,892 users and 17,632 artists. Movielens is a dataset for movie recommendation containing 47,957 interaction records between 2,113 users and 10,197 movies.

Generation of the Data The data is generated following (Li et al. 2019; Zhang et al. 2020; Wu et al. 2021). We treat each music artist and each movie as an arm. For both datasets, we extract $|\mathcal{A}| = 2,000$ arms with the most assigned tags by users and $N_u = 500$ users who have assigned the most tags. For each arm, we keep at most 20 tags that are related to the most arms, and consider them as the associated key-terms of the arm. All the kept key-terms associated with the arms form the key-term set $\mathcal{K}$. The number of key-terms for Last.FM is $|\mathcal{K}| = 2,726$ and that for Movielens is $5,585$. The weights of all key-terms related to the same arm are set to be equal. Based on the interactive recordings, the user feedback is constructed as follows: if the user has assigned tags to the item, the feedback is 1, otherwise the feedback is 0. To generate the feature vectors of users and arms, following (Li et al. 2019), we construct a feedback matrix $\boldsymbol{F} \in \mathbb{R}^{N_u \times N}$ based on the above user feedback, and decompose it using the singular-value decomposition (SVD): $\boldsymbol{F} = \boldsymbol{\Theta}\boldsymbol{S}\boldsymbol{X}^\top$, where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_{\boldsymbol{u}}^*)$, $u \in [N_u]$ and $\boldsymbol{X} = (\boldsymbol{x}_a)$, $a \in [N]$. We select $d = 50$ dimensions with highest singular values in $\boldsymbol{S}$. Following (Zhang et al. 2020), feature vectors of key-terms are calculated using $\tilde{\boldsymbol{x}}_k = \sum_{a\in\mathcal{A}} \frac{w_{a,k}}{\sum_{a'\in\mathcal{A}} w_{a',k}} \boldsymbol{x}_a$. The arm-level rewards and key-term-level feedback are then generated following Eq. (1) and Eq. (3).

6.2 Evaluation Results

This section first shows the results on both datasets in two cases: $\mathcal{K}$ is fixed and $\mathcal{K}$ is varying with time t. We also compare the running time of all algorithms on the Movielens dataset, since it has more key-terms than Last.FM.

Cumulative Regret We run the experiments 10 times and calculate the average regret of all the users over $T = 5,000$ rounds on the fixed generated datasets. We also include $\pm std$ as the error bar. For the time-varying key-term sets case, we set $|\mathcal{K}_t| = 1,000$ and randomly select $|\mathcal{K}_t|$ key-terms from $\mathcal{K}$ to form $\mathcal{K}_t$ at round t. Results on Last.FM and Movielens for fixed key-term set are shown in Figure 2 (a) and Figure 2 (b). On both datasets, the regrets of ConLinUCB-BS and ConLinUCB-MCR are much smaller than ConUCB (13.28% and 17.12% improvement on Last.FM, 13.08% and 16.93% improvement on Movielens, respectively) and even the simple ConLinUCB-UCB based on our ConLinUCB framework outperforms ConUCB. Results on Last.FM and Movielens for varying key-term sets are given in Figure 2 (c) and Figure 2 (d). ConLinUCB-MCR performs much better than ConUCB on both datasets (19.66% and 17.85% improvement on Last.FM and Movielens respectively).

Running Time We evaluate the running time of all conversational bandit algorithms on the representative Movielens dataset. For clarity, we report the total running time for selecting arms and key-terms. We set $T = 5,000$ and the results are shown in Table 1. It is clear that our algorithms cost much less time in both key-term selection and arm selection than ConUCB. Specifically, the improvements of total running time over ConUCB are 72.32% for ConLinUCB-BS and 57.32% for ConLinUCB-MCR. The main reason is our algorithms estimate the *unknown* user preference vector in one step, whereas ConUCB does it in two steps as mentioned before. For ConLinUCB-BS, time costed in key-term selection is almost negligible, since it randomly chooses key-terms from the precomputed barycentric spanner.

7 Conclusion

In this paper, we introduce ConLinUCB, a general framework for conversational bandits with efficient information incorporation. Based on this framework, we propose ConLinUCB-BS and ConLinUCB-MCR, with explorative key-term selection strategies that can quickly elicit the user's potential interests. We prove tight regret bounds of our algorithms. Particularly, ConLinUCB-BS achieves a bound of $O(\sqrt{dT\log T})$, much better than $O(d\sqrt{T}\log T)$ of the classic ConUCB. In experiments, our algorithms dramatically outperform ConUCB. For future work, it would be interesting to consider the settings with knowledge graphs (Zhao et al. 2022), hierarchy item trees (Song et al. 2022), relative feedback (Xie et al. 2021) or different feedback selection strategies (Letard et al. 2020, 2022), and use our framework and principles to improve the performance of existing algorithms.

Acknowledgements

The corresponding author Shuai Li is supported by National Natural Science Foundation of China (62006151) and Shanghai Sailing Program. The work of John C.S. Lui was supported in part by the RGC's GRF 14215722.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

Amballa, C.; Gupta, M. K.; and Bhat, S. P. 2021. Computing an Efficient Exploration Basis for Learning with Univariate Polynomial Features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6636–6643.

Awerbuch, B.; and Kleinberg, R. 2008. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1): 97–114.

Christakopoulou, K.; Beutel, A.; Li, R.; Jain, S.; and Chi, E. H. 2018. Q&R: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 139–148.

Christakopoulou, K.; Radlinski, F.; and Hofmann, K. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 815–824.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214. JMLR Workshop and Conference Proceedings.

Gao, C.; Lei, W.; He, X.; de Rijke, M.; and Chua, T.-S. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2: 100–126.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Letard, A.; Amghar, T.; Camp, O.; and Gutowski, N. 2020. Partial Bandit and Semi-Bandit: Making the Most Out of Scarce Users' Feedback. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 1073–1078. IEEE.

Letard, A.; Amghar, T.; Camp, O.; and Gutowski, N. 2022. COM-MABs: From Users' Feedback to Recommendation. In *The International FLAIRS Conference Proceedings*, volume 35.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Li, S.; Chen, W.; Li, S.; and Leung, K.-S. 2019. Improved Algorithm on Online Clustering of Bandits. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, 2923–2929. AAAI Press. ISBN 9780999241141.

Li, S.; Lei, W.; Wu, Q.; He, X.; Jiang, P.; and Chua, T.-S. 2021. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *ACM Transactions on Information Systems (TOIS)*, 39(4): 1–29.

Song, Y.; Sun, S.; Lian, J.; Huang, H.; Li, Y.; Jin, H.; and Xie, X. 2022. Show Me the Whole World: Towards Entire Item Space Exploration for Interactive Personalized Recommendations. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 947–956.

Sun, Y.; and Zhang, Y. 2018. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*, 235–244.

Wu, J.; Zhao, C.; Yu, T.; Li, J.; and Li, S. 2021. Clustering of Conversational Bandits for User Preference Learning and Elicitation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2129–2139.

Wu, Q.; Wang, H.; Gu, Q.; and Wang, H. 2016. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 529–538.

Xie, Z.; Yu, T.; Zhao, C.; and Li, S. 2021. Comparison-based Conversational Recommender System with Relative Bandit Feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1400–1409.

Zhang, X.; Xie, H.; Li, H.; and CS Lui, J. 2020. Conversational contextual bandit: Algorithm and application. In *Proceedings of The Web Conference 2020*, 662–672.

Zhang, Y.; Chen, X.; Ai, Q.; Yang, L.; and Croft, W. B. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*, 177–186.

Zhao, C.; Yu, T.; Xie, Z.; and Li, S. 2022. Knowledge-aware Conversational Preference Elicitation with Bandit Feedback. In *Proceedings of the ACM Web Conference 2022*, 483–492.