Contextual Combinatorial Bandit and its Application on Diversified Online Recommendation

Lijing Qin [†] Shouyuan Chen [‡] Xiaoyan Zhu [†]

Abstract

Recommender systems are faced with new challenges that are beyond traditional techniques. For example, most traditional techniques are based on similarity or overlap among existing data, however, there may not exist sufficient historical records for some new users to predict their preference, or users can hold diverse interest, but the similarity based methods may probably over-narrow it.

To address the above challenges, we develop a principled approach called *contextual combinatorial bandit* in which a learning algorithm can dynamically identify diverse items that interest a new user. Specifically, each item is represented as a feature vector, and each user is represented as an unknown preference vector. On each of n rounds, the bandit algorithm sequentially selects a set of items according to the item-selection strategy that balances *exploration* and *exploitation*, and collects the user feedback on these selected items. A reward function is further designed to measure the quality (e.g. relevance or diversity) of the selected set based on observed feedback, and the goal of the algorithm is to maximize the total rewards of n rounds. The reward function only needs to satisfy two mild assumptions that is general enough to accommodate a large class of nonlinear functions. To solve this bandit problem, we provide algorithm that achieves $\tilde{O}(\sqrt{n})$ regret after playing n rounds. Experiments conducted on real-wold movie recommendation dataset demonstrate that our approach can effectively address the above challenges and hence improve the performance of recommendation task.

1 Introduction

The multi-armed bandit (MAB) problem has been extensively studied in statistics and gained much popularity in the community of machine learning recently. It can be formulated as a sequential decision-making problem, where on each of n rounds a decision maker is presented with the choice of taking one of m arms (or actions), each having an unknown distribution of reward. The goal of the decision maker is to maximize the total expected rewards over the course of n rounds. When each arm is represented by a feature vector that can be observed by the decision maker, the problem is known as contextual bandit problem, which is recently used to develop recommender systems that adapt to user feedback.

User feedback is one kind of increasingly important source for online applications (e.g. Netflix project, Google news, Amazon) whose domains expand rapidly. Lots of new users don't have sufficient historical records, and hence are beyond the traditional recommendation technologies that anticipate a user's interest according to his/her past activities. Take movie recommendation for example, in contextual bandit setting, given a new user, we can repeatedly provide the user with one movie and collect his/her rating in multiple rounds. The algorithm helps decide which movie to recommend in the next round given the ratings in the previous rounds, i.e. whether we should try some new movies (exploration) or we should stick on the movies that the user has given high ratings so far (exploitation).

But in real-world scenario, recommender systems actually provide each user with a set of movies, rather than individual one. In this setting, not one simple arm but a set of arms (called a *super arm*) are played together on each round. The reward of a set of movies should not simply be the sum of ratings of each individual movie in this set. For example, we need a metric to qualify the diversity of the recommendation set to avoid redundant or over-specified recommendation lists. In this case, one can probably design a diversity promoting set function as the reward function of a super arm. The above problem can be described as a diversity promot-

[†] State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology. Dept. of Computer Science and Technology, Tsinghua University, Beijing, China. Emails: qinlj09@mail.tsinghua.edu.cn, zxy-dcs@tsinghua.edu.cn.

[‡] Dept. of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Email: sychen@cse.cuhk.edu.hk.

ing exploration/exploitation problem.

Motivated by the above challenges, we develop a principled approach called *contextual combinatorial bandit* to help predict user preference in recommender system. Specifically, movies are represented as feature vectors that can be regarded as contextual arms, and meanwhile users are represented as unknown preference variables, such that for users without enough historical records, a learning algorithm explores and exploits his/her preference by sequentially selecting sets of movies according to a diversity promoting reward function.

Note that the contextual combinatorial bandit is a general framework, and except for the above example setting, it can also be used in other applications by defining different reward functions, as long as the functions satisfy two mild assumptions that we will discuss in section 3.

In summary, the main contributions of this paper can be listed as follows: (i) We propose a general framework called contextual combinatorial bandit that can be used to address real-world recommendation challenges, i.e., user interest is unknown and meanwhile diverse. (ii) We develop C^{2}UCB, an efficient algorithm for contextual combinatorial bandit, for which we show a $\tilde{O}(\sqrt{n})$ regret bound after playing n rounds[1]. (iii) We apply the contextual combinatorial bandit approach on online diversified movie recommendation application. As evaluation, we conduct experiments on well-known movie recommendation dataset. The experiment results demonstrate that our approach significantly improves the performance of recommendation task.

2 Related Work

Most traditional recommendation techniques focus on learning user preference according to users' historical records [12, 23, 6]. However, recent studies show that historical records may not well represent user interest [24, 25]. On one hand, some users may not provide sufficient records, in which case it is crucial to predict user preference dynamically according to user feedback [16, 17]. On the other hand, users can hold diverse interest, and thus recommendation techniques should not only aim at increasing relevance, but also consider improving diversity of recommended results [25, 22].

To recommend items to users without sufficient historical records, several studies formulate this task as a multi-armed bandit problem [14, 15, 5]. Multi-armed bandit is a well-studied topic in the fields of statistics and machine learning (cf. [4, 2]). In traditional non-contextual bandit problem, the learner cannot access the features of arms, and the rewards of different arms are independent. In this setting, the upper confidence bound (UCB) algorithm is proven to be theoretically optimal [13, 3]. However, without using arm features, the performance of UCB algorithm is quite limited in many practical scenarios, especially when there are a large number of arms [14]. On the other hand, contextual bandit problem considers the case where the learner can observe the features of arms. Consequently, the learner can use these observations to infer the rewards of other unseen arms and improve the performance over time. Notably, Auer et al. [3] considered the contextual bandit problem and developed LinRel algorithm that achieved an $\tilde{O}(\sqrt{n})$ regret bound after playing n rounds. Later, Li et al.[14] proposed LinUCB algorithm, which improves the practical performance of LinRel algorithm while enjoys similar regret bound [8]. They applied LinUCB algorithm on a personalized news recommendation task and demonstrated good performance [14].

In the settings of both non-contextual and contextual bandits, the learner is allowed to play one single arm on each round, i.e., recommend one item each time. However, recommending a single item on each round may not satisfy a user's diverse interest. Recently, several work generalized the classical non-contextual bandits to combinatorial bandits [10, 11, 7], where the learner can play a set of arms, which is termed as a super arm, on each round. However, as generalizations of non-contextual bandits, these work did not use arm features. Hence, their performance can be suboptimal in many recommendation tasks, particularly when the number of arms is large. Though our method inherits some concepts (e.g. super arm) from non-contextual combinatorial bandit, both problem formulation and regret analysis are quite different, which are actually our main contributions.

Yue and Guestrin [26] proposed a linear submodular bandit approach for diversified retrieval. Their approach placed a strong restriction on user behavior. In particular, they assumed that user can only scan the items one by one in top-down fashion. In contrast, our framework has no limitation on user behavior. In addition, their framework is specifically designed for a certain type of submodular reward functions, while our approach allows a much larger class of reward functions.

[1] $\tilde{O}(\cdot)$ is variant of big O notation that ignores logarithmic factors.

3 Contextual Combinatorial Bandit

In this section, we formulate the contextual combinatorial bandit problem. Let n be the number of rounds and m be the number of arms. Let $\mathcal{S}_t \subseteq 2^{[m]}$ be the set of all possible subsets of arms on round t. We call each set of arms $S_t \in \mathcal{S}_t$ a *super arm*. On each round $t \in [n]$, a learner observes m feature vectors $\{\mathbf{x}_t(1), \ldots, \mathbf{x}_t(m)\} \subseteq \mathbb{R}^d$ corresponding to m arms. Then, the learner is asked to choose one super arm $S_t \in \mathcal{S}_t$ to play. Once a super arm $S_t \in \mathcal{S}_t$ is played, the learner observes the scores of arms in $\{r_t(i)\}_{i\in S_t}$ and receives a reward $R_t(S_t)$. For each arm $i \in [m]$, its score $r_t(i)$ is assumed to be

$$r_t(i) = \theta_*^T \mathbf{x}_t(i) + \epsilon_t(i), \tag{3.1}$$

where θ_* is a parameter unknown to the learner and the noise $\epsilon_t(i)$ is a zero-mean random variable. On the other hand, the reward $R_t(S_t)$ measures the quality of the super arm S_t and its definition will be specified later. The goal of the learner is to maximize the expected cumulative reward $\mathbb{E}\left[\sum_{t\in[n]} R_t(S_t)\right]$ over n rounds.

The reward $R_t(S_t)$ on round t is an application dependent function which measures the quality of recommended set of arms $S_t \subseteq [m]$. The reward can simply be the sum of the scores of arms in S_t, i.e. $R_t(S_t) = \sum_{i\in S_t} r_t(i)$. However, our framework also allows other more complicated non-linear rewards. For example, in addition to the sum of scores of arms, the reward $R_t(S_t)$ may also consider the "diversity" of arms in S_t, which can be defined as a non-linear function of features of arms.

Specifically, we consider the case where the expected reward $\mathbb{E}[R_t(S_t)]$ is a function of three variables: super arm S_t, feature vectors of arms $\mathbf{X}_t \triangleq \{\mathbf{x}_t(i)\}_{i\in[m]}$ and expected scores $\mathbf{r}_t^* \triangleq \{\theta_*^T \mathbf{x}_t(i)\}_{i\in[m]}$ associated with the arms. Formally, we denote the expected reward of playing S_t as $\mathbb{E}[R_t(S_t)] = f_{\mathbf{r}_t^*,\mathbf{X}_t}(S_t)$. By choosing different types of expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$, our framework covers both linear and non-linear rewards. Finally, in order to carry out our analysis, the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$ is required to satisfy the following assumptions.

Monotonicity The expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ is monotone non-decreasing with respect to the score vector $\mathbf{r}$. Formally, for any set of feature vectors of arms $\mathbf{X}$ and super arm S, if $r(i) \le r'(i)$ for all $i \in [m]$, we have $f_{\mathbf{r},\mathbf{X}}(S) \le f_{\mathbf{r}',\mathbf{X}}(S)$.

Lipschitz continuity The expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ is Lipschitz continuous with respect to the score vector $\mathbf{r}$ restricted on the arms in S. In particular, there exists a universal constant $C > 0$ such that, for any two score vectors $\mathbf{r}$ and $\mathbf{r}'$, we have $|f_{\mathbf{r},\mathbf{X}}(S) - f_{\mathbf{r}',\mathbf{X}}(S)| \le C\sqrt{\sum_{i\in S}[r(i) - r'(i)]^2}$.

Our framework does not require the player to have direct knowledge on how the reward function $f_{\mathbf{r},\mathbf{X}}(S)$ is defined. Alternatively, we assume that the player has access to an oracle $\mathcal{O}_{\mathcal{S}}(\mathbf{r},\mathbf{X})$, which takes the expected scores $\mathbf{r}$ and arms $\mathbf{X}$ as input, and returns the solution of the maximization problem $\arg\max_{S\in\mathcal{S}} f_{\mathbf{r},\mathbf{X}}(S)$. Since the maximization problems of many reward functions $f_{\mathbf{r},\mathbf{X}}(\cdot)$ of practical interest are NP-hard, our framework allows the oracle to produce an approximate solution to the problem. More precisely, an oracle $\mathcal{O}_{\mathcal{S}}(\mathbf{r},\mathbf{X})$ is called *α-approximation oracle* for some $\alpha \le 1$, if given input $\mathbf{r}$ and $\mathbf{X}$, the oracle always returns a super arm $S = \mathcal{O}_{\mathcal{S}}(\mathbf{r},\mathbf{X}) \in \mathcal{S}$ satisfying $f_{\mathbf{r},\mathbf{X}}(S) \ge \alpha\text{opt}_{\mathbf{r},\mathbf{X}}$, where $\text{opt}_{\mathbf{r},\mathbf{X}} = \max_{S\in\mathcal{S}} f_{\mathbf{r},\mathbf{X}}(S)$ is the optimal value of the reward function. Under this setting, when $\alpha = 1$, the α-approximation oracle is exact and always produces the optimal solution.

Recall that the goal of the learner is to maximize its cumulative reward without knowing θ_*. Clearly, with the knowledge of θ_*, the optimal strategy is to choose $S_t = \arg\max_{S_t\in\mathcal{S}_t} f_{\mathbf{r}_t,\mathbf{X}_t}(S_t)$ on round t. Hence, it is natural to evaluate a learner relative to this optimal strategy and the difference of the learner's total reward and the total reward of the optimal strategy is called *regret*. However, if a learner only has accesses to an α-approximation oracle for some $\alpha < 1$, such evaluation would be unfair. Hence, in this paper, we use the notion of α-regret which compares the learner's strategy with α-fraction of the optimal rewards on round t. Formally, the α-regret on round t can be written as

$$\text{Reg}_t^\alpha = \alpha\text{opt}_{\mathbf{r}_t,\mathbf{X}_t} - f_{\mathbf{r}_t,\mathbf{X}_t}(S_t), \tag{3.2}$$

and we are interested in designing an algorithm whose total α-regret $\sum_{t=1}^T \text{Reg}_t^\alpha$ is as small as possible.

4 Algorithm and α-Regret Analysis

In this section, we present contextual combinatorial upper confidence bound algorithm (C^{2}UCB). C^{2}UCB is a general and efficient algorithm for the contextual combinatorial bandit problem. The basic idea of C^{2}UCB is to maintain a confidence set for the true parameter θ_*. For each round t, the confidence set is constructed from feature vectors $\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}$ and observed scores of selected arms $\{r_1(i)\}_{i\in S_1}, \ldots, \{r_{t-1}(i)\}_{i\in S_{t-1}}$ from previous rounds. As we will see later (Theorem 4.2), our construction of the confidence sets ensures that the true parameter θ_* lies in the confidence set with high probability. Using this confidence set of parameter θ_*

and feature vectors of arms $\mathbf{X}_t$, the algorithm can efficiently compute an upper confidence bound for each score $\hat{\mathbf{r}}_t = \{\hat{r}_t(1), \dots, \hat{r}_t(m)\}$. The upper confidence bounds $\hat{\mathbf{r}}_t$ and feature vectors of arms $\mathbf{X}_t$ are given to the oracle as input. Then, the algorithm plays the super arm returned by the oracle and uses the observed scores to adjust the confidence sets. The pseudocode of the algorithm is listed in Algorithm 1. The algorithm has time complexity $O(n(d^3 + md + h))$, where h denotes the time complexity of the oracle.

Algorithm 1 C^{2}UCB

```
input: $\lambda, \alpha_1, \dots, \alpha_n$
Initialize $\mathbf{V}_0 \leftarrow \lambda \mathbf{I}_{d \times d}, \mathbf{b}_0 \leftarrow \mathbf{0}_d$
for $t \leftarrow 1, \dots, n$ do
    $\hat{\theta}_t \leftarrow \mathbf{V}_{t-1}^{-1} \mathbf{b}_{t-1}$
    for $i \in 1, \dots, m$ do
        $\bar{r}_t(i) \leftarrow \hat{\theta}_t^T \mathbf{x}_t(i)$
        $\hat{r}_t(i) \leftarrow \bar{r}_t(i) + \alpha_t \sqrt{\mathbf{x}_t(i)^T \mathbf{V}_t^{-1} \mathbf{x}_t(i)}$
    end for
    $S_t \leftarrow \mathcal{O}_{\mathcal{S}_t}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$
    Play super arm $S_t$ and observe $\{r_t(i)\}_{i \in S_t}$
    $\mathbf{V}_t \leftarrow \mathbf{V}_{t-1} + \sum_{i \in S_t} \mathbf{x}_t(i) \mathbf{x}_t(i)^T$
    $\mathbf{b}_t \leftarrow \mathbf{b}_{t-1} + \sum_{i \in S_t} r_t(i) \mathbf{x}_t(i)$
end for
```

We now state our main theoretical result, a bound on the α-regret of Algorithm 1 when run with an α-approximation oracle. To carry out our analysis, we will need to assume that the l_2-norms of parameter θ_* and feature vectors of arms $\mathbf{X}_t$ are bounded. Using this assumption together with the monotonicity and Lipschitz continuity properties of the expected reward function $f_{\mathbf{r},\mathbf{X}}(\cdot)$, the following theorem states that the α-regret of Algorithm 1 is at most $O(d \log(n)\sqrt{n} + \sqrt{nd \log(n/\delta)})$, or $\tilde{O}(\sqrt{n})$ if one ignores logarithmic factors and regards the dimensionality of the parameter d as a constant.

Theorem 4.1. (α-regret of the Algorithm 1). *Without loss of generality, assume that $\|\theta_*\|_2 \le S$, $\|\mathbf{x}_t(i)\|_2 \le 1$ and $r_t(i) \in [0,1]$ for all $t \ge 0$ and $i \in [m]$. Given $0 < \delta < 1$, set $\alpha_t = \sqrt{d \log\left(\frac{1+tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$. Then, with probability at least $1 - \delta$, the total α-regret of C^{2}UCB algorithm satisfies*

$$\sum_{t=1}^{n} Reg_t^{\alpha} \le C\sqrt{64nd \log(1 + nm/d\lambda)} \cdot \left(\sqrt{\lambda} S + \sqrt{2\log(1/\delta) + d\log(1 + nm/(\lambda d))}\right),$$

for any $n \ge 0$.

Note that the requirements $\|\mathbf{x}_t(i)\|_2 \le 1$ and $r_t(i) \in [0,1]$ can be satisfied through proper rescaling on $\mathbf{x}_t(i)$ and θ_*.

4.1 Proof of Theorem 4.1

We begin with restating a concentration result from Abbasi-Yadkori et al., [1]. This result states that the true parameter θ_* lies within an ellipsoid centered at $\hat{\theta}_t$ simultaneously for all $t \in [n]$ with high probability.

Theorem 4.2. ([1, Theorem 2]) *Suppose the observed scores $r_t(i)$ are bounded in $[0,1]$. Assume that $\|\theta_*\|_2 \le S$ and $\|\mathbf{x}_t(i)\|_2 \le 1$ for all $t \ge 0$ and $i \in [m]$. Define $\mathbf{V}_t = \mathbf{V} + \sum_{t=1}^{n} \sum_{i \in S_t} \mathbf{x}_t(i)\mathbf{x}_t(i)^T$ and set $\mathbf{V} = \lambda \mathbf{I}$. Then, with probability at least $1 - \delta$, for all round $t \ge 0$, the estimate $\hat{\theta}_t$ satisfies* [2]

$$\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \le \sqrt{d \log\left(\frac{1 + tm/\lambda}{\delta}\right)} + \lambda^{1/2} S.$$

The proof of Theorem 4.2 is based on the theory of self-normalized processes. For an introduction to this theory, we refer interested readers to [21, 9].

Next, using Theorem 4.2, we show that with high probability, the upper confidence bounds of scores $\hat{\mathbf{r}}_t$ also do not deviate far from the true value of scores $\mathbf{r}_t^*$ for each round $t \in [n]$.

Lemma 4.1. *If we set $\alpha_t = \sqrt{d \log\left(\frac{1+tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$, with probability at least $1 - \delta$, we have*

$$0 \le \hat{r}_t(i) - r_t^*(i) \le 2\alpha_t \|\mathbf{x}_t(i)\|_{\mathbf{V}_{t-1}^{-1}},$$

holds simultaneously for any round $t \ge 0$ and any arm $i \in [m]$.

Proof. By Theorem 4.2, the random event

$$\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \le \sqrt{d \log\left(\frac{1 + tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$$

holds for all $t \in [n]$ simultaneously with probability at least $1 - \delta$.

Now assume the above random event happens, by the

[2] We denote $\|\mathbf{a}\|_{\mathbf{M}} \triangleq \sqrt{\mathbf{a}^T \mathbf{M} \mathbf{a}}$, where $\mathbf{a}$ is a vector and $\mathbf{M}$ is a positive definite matrix.

definition of $\hat{r}_t(i)$, we have

$$\begin{aligned}
&|\hat{r}_t(i) - r_t^*(i)| \\
=& \left|\hat{\theta}_t^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} - \theta_*^T \mathbf{x}_t(i)\right| \\
\le& \left|(\hat{\theta}_t - \theta_*)^T \mathbf{x}_t(i)\right| + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\le& \left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\le& 2\alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} .
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&\hat{r}_t(i) - r_*(i) \\
=& \hat{\theta}_t^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} - \theta_*^T \mathbf{x}_t(i) \\
=& (\hat{\theta}_t - \theta_*)^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\ge& -\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\ge& -\alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} = 0.
\end{aligned}$$

To prove our main result Theorem 4.1, we need the following technical lemma.

LEMMA 4.2. *Let $\mathbf{V} \in \mathbb{R}^{d\times d}$ be a positive definite matrix. For all $t = 1, 2, \ldots,$ let S_t be a subset of $[m]$ of size less than or equal to k and define $\mathbf{V}_n = \mathbf{V} + \sum_{t=1}^n \sum_{i\in S_t} \mathbf{x}_t(i)\mathbf{x}_t(i)^T$.*

Then, if $\lambda \ge k$ and $\|\mathbf{x}_t(i)\|_2 \le 1$ for all t and i, we have

$$\begin{aligned}
\sum_{t=1}^n \sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}} &\le 2\log\det \mathbf{V}_n - \log\det \mathbf{V} \\
&\le 2d\log((\mathrm{trace}(\mathbf{V}) + nk)/d) \\
&\quad - 2\log\det \mathbf{V}.
\end{aligned}$$

Proof. We have

$$\begin{aligned}
&\det(\mathbf{V}_n) \\
=& \det\left(\mathbf{V}_{n-1} + \sum_{i\in S_n} \mathbf{x}_n(i)\mathbf{x}_n(i)^T\right) \\
=& \det(\mathbf{V}_{n-1})\det\left(\mathbf{I} + \sum_{i\in S_n} (\mathbf{V}_{n-1}^{-1/2}\mathbf{x}_n(i))(\mathbf{V}_{n-1}^{-1/2}\mathbf{x}_n(i))^T\right) \\
=& \det(\mathbf{V}_{n-1})\det\left(\mathbf{I} + \sum_{i\in S_n} \left\|\mathbf{x}_n(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}\right) \\
=& \det(\mathbf{V})\prod_{t=1}^n \left(1 + \sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}\right).
\end{aligned}$$

Now, using the fact that $u \le 2\log(1+u)$ for any $u \in [0,1]$ and that $\left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}} \le \left\|\mathbf{x}_t(i)\right\|^2/\lambda_{\min}(\mathbf{V}_{t-1}) \le 1/\lambda \le 1/k$, we obtain

$$\begin{aligned}
\sum_{t=1}^n \sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}} &\le 2\sum_{t=1}^n \log\left(1 + \sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}\right) \\
&= 2\log\det \mathbf{V}_n - 2\log\det \mathbf{V}.
\end{aligned}$$

We remain to bound $\log\det \mathbf{V}_n$. Since $\|\mathbf{x}_n(i)\|_2 \le 1$ and $|S_i| \le k$ for all $i \in [n]$, the trace of $\mathbf{V}_n$ can be bounded by $\mathrm{trace}(\mathbf{V}_n) \le \mathrm{trace}(\mathbf{V}) + nk$. Apply the Determinant-Trace Inequality [1, Lemma 10], we have

$$\log\det(\mathbf{V}_n) \le d\log((\mathrm{trace}(\mathbf{V}) + nk)/d).$$

Based on Lemma 4.1, Lemma 4.2 and the two assumptions on the expected reward, we are now ready to prove our main theorem.

Proof. (Theorem 4.1) By Lemma 4.1, we have $\hat{r}_t(i) \ge r_t^*(i)$ holds simultaneously for all $t \in [n]$ and $i \in [m]$ with probability at least $1 - \delta$. Now, assume that this random event holds and apply the monotonicity property of the expected reward, for any super arm $S \in \mathcal{S}_t$, we have $f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S) \ge f_{\mathbf{r}_t^*, \mathbf{X}_t}(S)$.

Let $S_t \in \mathcal{S}_t$ be the super arm returned by the oracle $S_t = \mathcal{O}_{\mathcal{S}_t}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$ on round t. We now show that $f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S_t) \ge \alpha\mathrm{opt}_{\mathbf{r}_t^*, \mathbf{X}_t}$. To see this, we denote $S_t^* = \arg\max_{S\in\mathcal{S}_t} f_{\mathbf{r}_t^*, \mathbf{X}_t}(S)$ as the maximizer of $f_{\mathbf{r}_t^*, \mathbf{X}_t}(\cdot)$ and $\hat{S}_t$ as the optimal solution of $\arg\max_{S\in\mathcal{S}_t} f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S)$. Then, we have

$$\begin{aligned}
f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S_t) &\ge \alpha\mathrm{opt}_{\hat{\mathbf{r}}_t, \mathbf{X}_t} \\
&= \alpha f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(\hat{S}_t) \ge \alpha f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S_t^*) \\
&\ge \alpha f_{\mathbf{r}_t^*, \mathbf{X}_t}(S_t^*) = \alpha\mathrm{opt}_{\mathbf{r}_t^*, \mathbf{X}_t},
\end{aligned}$$

where we have used the definition of α-approximation oracle and the optimality of $\hat{S}_t$.

Now, we can bound α-regret at round t as follows,

$$\begin{aligned}
\mathrm{Reg}_t^\alpha &= \alpha\mathrm{opt}_{\mathbf{r}_t^*, \mathbf{X}_t} - f_{\mathbf{r}_t^*, \mathbf{X}_t}(S_t) \\
&\le f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S_t) - f_{\mathbf{r}_t^*, \mathbf{X}_t}(S_t) \\
&\le C\sqrt{\sum_{i\in S_t} (\hat{r}_t(i) - r_t^*(i))^2} \\
&\le C\sqrt{\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}},
\end{aligned}$$

where the second inequality follows from the Lipschitz continuity property of the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$.

Therefore, with probability at least $1-\delta$, for all $n \geq 0$,

$$\begin{aligned}
\sum_{t=1}^{n} \mathrm{Reg}_t^{\alpha} &\leq \sqrt{n \sum_{t=1}^{n} (\mathrm{Reg}_t^{a})^2} \\
&\leq C \sqrt{8n \sum_{t=1}^{n} \sum_{i \in S_t} 4\alpha_t^2 \left\| \mathbf{x}_t(i) \right\|_{\mathbf{V}_{t-1}^{-1}}^2} \\
&\leq C\alpha_n \sqrt{32n} \sqrt{\sum_{t=1}^{n} \sum_{i \in S_t} \left\| \mathbf{x}_t(i) \right\|_{\mathbf{V}_{t-1}^{-1}}^2} \\
&\leq C\alpha_n \sqrt{32n} \sqrt{2d \log(\lambda + nm/d) - 2d \log \lambda} \\
&= C\sqrt{64nd \log(1 + nm/d\lambda)} \cdot \\
&\qquad \left(\sqrt{d \log\left((1 + nm/\lambda)/\delta\right)} + \sqrt{\lambda} S \right),
\end{aligned}$$

where the last inequality follows from Lemma 4.2, the fact that $|S_t| \leq m$ for all t and that $\mathbf{V} = \lambda \mathbf{I}$.

5 Application: Online Diversified Movie Set Recommendation

In this section, we consider an application with substantial practical interest: diversified movie recommendation. In this application, the recommender system recommends sets of movies, rather than individual ones. In addition, the recommended movies should be diversified such that the coverage of information that interests users is maximized. Furthermore, the recommender system need to use the user's feedback to improve its performance for future recommendations.

This application can be naturally formulated as a contextual combinatorial problem as follows. Suppose, on each round t, there are m available movies and each movie is represented as a feature vector $\mathbf{x}_t(i) \in \mathbb{R}^d$. We can view the m movies as m arms and regard the feature vectors of movies as the feature vectors associated with arms. Then, the parameter $\theta_* \in \mathbb{R}^d$ corresponds to the user's (unknown) preference and the scores $r_t(i)$ are the ratings given by the user. On each round t, the system need to recommend a set of exactly k movies. This cardinality constraint is equivalent to assign the set of allowed super arms $\mathcal{S}_t = \{S | S \in 2^{[m]} \text{ and } |S| = k\}$ to be the set of all subsets of size k for all $t \geq 0$.

Next, we define the expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ of a super arm S and construct an α-approximation oracle that associates to the expected reward. The definition of reward of super arm S should reflect both relevance and diversity of the set of movies in the super arm. In this paper, we consider the following definition of reward which is proposed recently by Qin and Zhu [22],

$$(5.3) \qquad f_{\mathbf{r},\mathbf{X}}(S) = \sum_{i \in S} r(i) + \lambda h(S, \mathbf{X}),$$

where $h(S, \mathbf{X}) = \frac{1}{2}|S| \log(2\pi e) + \frac{1}{2} \log \det(\mathbf{X}(S)^T \mathbf{X}(S) + \sigma^2 \mathbf{I})$ is called entropy regularizer since it quantifies the posterior uncertainty of ratings of movies in the set S. Here, the matrix $\mathbf{X}(S) \in \mathbb{R}^{d \times |S|}$ denotes a submatrix of $\mathbf{X}$ that consists of columns indexed by S and σ^2 is a smoothing parameter. This definition of entropy regularizer is derived as the differential entropy of ratings based on the Probabilistic Matrix Factorization (PMF) model. The derivation is omitted here due to space constraint and we refer interested readers to [22] for details. Finally, the parameter λ of Eq. (5.3) is a regularization constant which trades-off between relevance and diversity.

As shown in [22], this definition of reward has several desirable properties. First, the value of entropy regularizer $h(S, \mathbf{X})$ is maximized if the feature vectors of movies in S are orthogonal (most dissimlar) and is minimized when the feature vectors are linearly dependent (most similar). This property captures the intuition of the diversity of a set of feature vectors. Second, the function $f_{\mathbf{r},\mathbf{X}}(\cdot)$ is submodular and monotone for $\sigma^2 \geq 0.0586$. Consequently, there exists efficient approximation algorithms with rigorous guarantees to solve the combinatorial maximization problem of finding the super arm with the highest expected reward, which can be formulated as $\arg\max_{|S|=k} f_{\mathbf{r},\mathbf{X}}(S)$.

In particular, a simple greedy algorithm is guaranteed to find the super arm with reward larger than $(1 - 1/e)OPT$, where OPT is the reward of the best super arm [20, 22]. By definition, this algorithm can be employed as a valid $(1 - 1/e)$-approximation oracle in the contextual combinatorial bandit framework. The detailed implementation of the greedy algorithm as an $(1 - 1/e)$-approximation oracle $\mathcal{O}_k^{\mathrm{div}}(\mathbf{r}, \mathbf{X})$ is shown in Algorithm 2. The time complexity of Algorithm 2 is $O(k^4)$, which is acceptable in most applications where k is a small constant ranging from 5 to 20.

Now, we can plug this oracle $\mathcal{O}_k^{\mathrm{div}}(\mathbf{r}, \mathbf{X})$ in C^{2}UCB to construct an algorithm for the online diversified movie recommendation application. This can be done by simply changing Line 9 of Algorithm 1 to $S_t \leftarrow \mathcal{O}_k^{\mathrm{div}}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$. We denote the resulting algorithm as C^{2}UCB$^{\mathrm{div}}$, whose total time complexity is $O(n(d^3 + md + k^4))$.

To rigorously establish the theoretical guarantees for C^{2}UCB$^{\mathrm{div}}$ algorithm, we remain to verify whether the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$ defined in Eq (5.3) satisfies the monotonicity and Lipschitz continuity properties, which are required by Theorem 4.1. The monotonicity property is straightforward since $f_{\mathbf{r},\mathbf{X}}(\cdot)$ depends on $\mathbf{r}$ only through $\sum_{i\in S} r(i)$, which is clearly monotone with respect to $\mathbf{r}$. On the other hand, for any set S of size k and any collection of feature vectors $\mathbf{X}$, we have

$$\begin{aligned}|f_{\mathbf{r},\mathbf{X}}(S) - f_{\mathbf{r}',\mathbf{X}}(S)| &= \left|\sum_{i\in S} r(i) - r'(i)\right| \\ &\le \sqrt{k}\sqrt{\sum_{i\in S}(r(i) - r'(i))^2}.\end{aligned}$$

Hence, the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$ satisfies the Lipschitz continuity property with Lipschitz constant $C = \sqrt{k}$. Therefore, by Theorem 4.1, we can immediate obtain the $(1-1/e)$-regret bound of recommending diversified movie sets using C^{2}UCB$^{\text{div}}$as

$$\begin{aligned}&k\sqrt{64nd\log(1+nm/d\lambda)} \\ &\quad\left(\sqrt{\lambda}S + \sqrt{2\log(1/\delta) + d\log(1+nm/(\lambda d))}\right).\end{aligned}$$

Algorithm 2 $\mathcal{O}_k^{\text{div}}(\mathbf{r},\mathbf{X})$: a $(1-1/e)$-approximation oracle for diversified movie set recommendation

```
input: scores $\mathbf{r} \in \mathbb{R}^m$, feature vectors $\mathbf{X} \in \mathbb{R}^{d\times m}$
$S \leftarrow \emptyset$, $\mathbf{C} \leftarrow \sigma^{-2}$
for $j = 1$ to $k$ do
    for $i \in [m]\backslash S$ do
        $\Sigma_{iS} \leftarrow \mathbf{x}(i)^T\mathbf{X}(S)$
        $\delta_i^{(R)} \leftarrow r_i$
        $\delta_i^{(g)} \leftarrow \frac{1}{2}\log(2\pi e(\sigma^2 + \Sigma_{iS}\mathbf{C}\Sigma_{iS}^T))$
    end for
    $i^* \leftarrow \arg\max_{i\in[m]\backslash S} \delta_i^{(R)} + \lambda\delta_i^{(g)}$
    $S \leftarrow S \cup \{i^*\}$
    $\mathbf{C} \leftarrow (\mathbf{X}(S)^T\mathbf{X}(S) + \sigma^2\mathbf{I})^{-1}$
end for
return $S$
```

6 Experiments

6.1 Experiment Setup

We conduct experiments on the MovieLens dataset, which is a public dataset consisting 1,000,029 ratings for 3900 movies by 6040 users of online movie recommendation service [19]. Each element of the dataset is represented by a tuple $t_{i,j} = (u_i, v_j, r_{i,j})$, where u_i denotes userID, v_j denotes movieID, and $r_{i,j}$ which is an integer score between 1 and 5 denotes the rating of user i for movie j (higher score indicates higher preference).

We split the dataset into training and test set as follows. We construct the test set by randomly selecting 300 users such that each selected user has at least 100 ratings. The remaining 5740 users and their ratings belong to the training set. Then, we apply a rank-d probabilistic matrix factorization (PMF) algorithm on the training data to learn the feature vectors of movies (each feature vector is d-dimensional). These feature vectors will be used later by the bandit algorithms as the feature vectors of arms.

Baselines. We compare our combinatorial contextual bandit algorithm with the following baselines.

k-LinUCB algorithm. LinUCB [14] is a contextual bandit algorithm which recommends exactly one arm at each time. To recommend a set of k movies, we repeat LinUCB algorithm k times on each round. By sequentially removing recommended arms, we ensure the k arms returned by LinUCB are distinct on each round. Finally, we highlight that the resulting bandit algorithm can be regarded as a combinatorial contextual bandit with linear expected reward function

$$f_{\mathbf{r},\mathbf{X}}(S) = \sum_{i\in S} r(i).$$

Therefore, the major difference between k-LinUCB algorithm and our C^{2}UCB$^{\text{div}}$algorithm, which uses a reward function defined in Eq. (5.3), lies in that our algorithm optimizes the diversity of arms in set S_t.

Warm-start diversified movie recommendation. We denote this baseline as "warm-start" for short. For each user u in test set, we randomly select η ratings to train an user preference vector using PMF model. We call the parameter η as warm-start offset. With the estimated preference vector, one can repeatedly recommend sets of diverse recommendation results by maximizing the reward function 5.3. Note that this method cannot dynamically adapts to user's feedback, and thus each round is independent with the others.

Metric. We use precision to measure the quality of recommended movie sets over n rounds. Specifically, for each user u in the test test, we define the set of "preferred movies" L_u as the set of movies which user u assigned a rating of 4 or 5. Intuitively, a good movie set recommendation algorithm should recommend movie sets which cover a large fraction of preferred movies. Formally, on round t, suppose the recommendation algorithm recommends a set of movies S_t. The precision $p_{t,u}$ of user u on round t is defined as

$$p_{t,u} = \frac{|S_t \cap L_u|}{|S_t|}.$$

Then, the average precision of P_t of all test users up to round t is given by

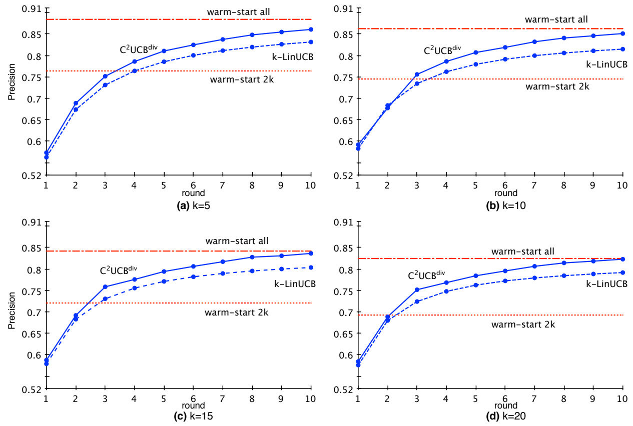
$$P_t = \frac{1}{t|U|} \sum_{u \in U} \sum_{i=1}^{t} p_{i,u}.$$

Note that we do not aim at predicting the ratings of movies, but to provide more satisfying recommendation lists. Hence, precision is a more appropriate metric rather than the root mean square error (RMSE). Actually, our algorithm (as well as baselines) essentially used an l_2-regularized linear regression method to predict movie ratings based on existing observations. This is equivalent to the rating prediction methods used by many matrix factorization algorithms, which are shown to have low RMSEs [18]. Moreover, we cannot use regret as a metric either, because the definitions of regrets vary greatly for different bandit algorithms.

6.2 Experiment Results We consider recommending different number of movies to each user on each round, i.e., the size of super arm k takes values in $\{5, 10, 15, 20\}$. For each k, we set the exploration parameter $\alpha_t = 1.0$. The parameters of entropy regularizer are set to be $\lambda = 0.5$ and $\sigma = 1.0$.

For the warm-start baseline that allows an offline-estimated user preference, we consider two cases where η takes different values. In one case that we denote as "warm-start 2k", $\eta = k \times 2$ which indicates the method can access ratings of two rounds. In the other case that we denote as "warm-start all", η equals the total amount of observations, which indeed corresponds to the best solution, i.e., all observations are used to train user preference.

The results are shown in Figure 1 and Table 1, where our approach is denoted as $\text{C}^2\text{UCB}^{\text{div}}$. We can see that, in all cases, the "warm-start 2k" baseline outperforms bandit algorithms on earlier rounds, which is reasonable since the "warm-start 2k" baseline is provided warm-start observations to learn the user preference. But when more user feedbacks are available, bandit algorithms improve performance by dynamically adapting to user feedbacks. Near the end of 10 rounds, $\text{C}^2\text{UCB}^{\text{div}}$ can achieve a result that is comparable to "warm-start all". Compared to k-LinUCB, our method $\text{C}^2\text{UCB}^{\text{div}}$ finds a better match between recommended movies and user interest (i.e., the movies liked by each given user), and thus improves the overall performance. Furthermore, when k is larger, $\text{C}^2\text{UCB}^{\text{div}}$ algorithm obtains larger performance gain over k-LinUCB algorithm. The experiment results indicate that our method helps uncover users' diverse interest by using a non-linear diversity promoting reward function.



(a) k=5

(b) k=10

(c) k=15

(d) k=20

Figure 1: Experiment results comparing $\text{C}^2\text{UCB}^{\text{div}}$ with k-LinUCB, warm-start $2t$ and warm-start all on different choices of k.

	$t = 5$		$t = 10$		warm-start	
	KL	CC	KL	CC	$\eta = 2k$	$\eta = \text{all}$
$k = 5$	0.785	0.810	0.831	0.861	0.763	0.884
$k = 10$	0.779	0.806	0.814	0.851	0.745	0.862
$k = 15$	0.770	0.793	0.803	0.836	0.720	0.841
$k = 20$	0.762	0.784	0.791	0.822	0.692	0.824

Table 1: Precision values of competing algorithms. CC: $\text{C}^2\text{UCB}^{\text{div}}$algorithm. KL: k-LinUCB algorithm.

7 Conclusion

We presented a general framework called contextual combinatorial bandit that accommodates combinatorial nature of contextual arms. We developed an efficient algorithm C^2UCB for contextual combinatorial bandit and provide a rigorously regret analysis. We further applied this framework on online diversified movie recommendation task, and developed a specific algorithm $\text{C}^2\text{UCB}^{\text{div}}$for this application. Experiments on public MovieLens dataset demonstrate that our approach helps explore and exploit users' diverse preference, and hence improves the performance of recommendation task.

Acknowledgments

This work is supported by China National Basic Research Program No. 2012CB316301, China National Basic Research Program No. 2013CB329403, and the Chinese Natural Science Foundation important project under grant No.61332007.

References

[1] Y. Abbasi-Yadkori, C. Szepesvári, and D. Tax, *Improved algorithms for linear stochastic bandits*, in Advances in Neural Information Processing Systems, 2011, pp. 2312–2320.

[2] V. Anantharam, P. Varaiya, and J. Walrand, *Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards; part ii: Markoveian rewards*, Automatic Control, IEEE Transactions on, 32 (1987), pp. 968–982.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, Machine learning, 47 (2002), pp. 235–256.

[4] D. M. Berry, *Bandit problems*, (1985).

[5] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski, *A contextual-bandit algorithm for mobile context-aware recommender system*, in Neural Information Processing, Springer, 2012, pp. 324–331.

[6] R. Burke, *Hybrid recommender systems: Survey and experiments*, User modeling and user-adapted interaction, 12 (2002), pp. 331–370.

[7] W. Chen, Y. Wang, and Y. Yuan, *Combinatorial multi-armed bandit: General framework and applications*, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 151–159.

[8] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, *Contextual bandits with linear payoff functions*, in International Conference on Artificial Intelligence and Statistics, 2011, pp. 208–214.

[9] V. H. de la Peña, M. J. Klass, and T. L. Lai, *Self-normalized processes: exponential inequalities, moment bounds and iterated logarithm laws*, Annals of probability, (2004), pp. 1902–1933.

[10] Y. Gai, B. Krishnamachari, and R. Jain, *Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation*, in New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on, IEEE, 2010, pp. 1–9.

[11] Y. Gai, B. Krishnamachari, and R. Jain, *Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations*, IEEE/ACM Transactions on Networking (TON), 20 (2012), pp. 1466–1478.

[12] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.

[13] T. L. Lai and H. Robbins, *Asymptotically efficient adaptive allocation rules*, Advances in applied mathematics, 6 (1985), pp. 4–22.

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

[15] L. Li, W. Chu, J. Langford, and X. Wang, *Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms*, in Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 297–306.

[16] P. Massa and P. Avesani, *Trust-aware recommender systems*, in Proceedings of the 2007 ACM conference on Recommender systems, ACM, 2007, pp. 17–24.

[17] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, *Ontological user profiling in recommender systems*, ACM Transactions on Information Systems (TOIS), 22 (2004), pp. 54–88.

[18] A. Mnih and R. Salakhutdinov, *Probabilistic matrix factorization*, in Advances in neural information processing systems, 2007, pp. 1257–1264.

[19] *MovieLens dataset*, in http://movielens.org.

[20] G. Nemhauser, L. Wolsey, and M. Fisher, *An analysis of approximations for maximizing submodular set functionsi*, Mathematical Programming, 14 (1978), pp. 265–294.

[21] V. H. Peña, H. Víctor, T. L. Lai, and Q.-M. Shao, *Self-Normalized Processes: Limit Theory and Statistical Applications*, Springer, 2009.

[22] L. Qin and X. Zhu, *Promoting diversity in recommendation by entropy regularizer*, in Proceedings of the Twenty-third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 2698–2704.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *Item-based collaborative filtering recommendation algorithms*, in Proceedings of the 10th international conference on World Wide Web, ACM, 2001, pp. 285–295.

[24] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, *Methods and metrics for cold-start recommendations*, in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2002, pp. 253–260.

[25] C. Yu, L. Lakshmanan, and S. Amer-Yahia, *It takes variety to make a world: diversification in recommender systems*, in Proceedings of the 12th international conference on extending database technology: Advances in database technology, ACM, 2009, pp. 368–378.

[26] Y. Yue and C. Guestrin, *Linear submodular bandits and their application to diversified retrieval*, in Advances in Neural Information Processing Systems, 2011, pp. 2483–2491.