

A Hybrid Bandit Framework for Diversified Recommendation

Qinxu Ding,¹ Yong Liu,^{1,2*} Chunyan Miao,³ Fei Cheng,⁴ Haihong Tang⁴

¹ Alibaba-NTU Singapore Joint Research Institute

² Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY)

³ School of Computer Science and Engineering, Nanyang Technological University

⁴ Alibaba Group

{qding001@e., stephenliu@, asycymiao@}ntu.edu.sg, hongyuan.cf@alibaba-inc.com, piaoxue@taobao.com

Abstract

The interactive recommender systems involve users in the recommendation procedure by receiving timely user feedback to update the recommendation policy. Therefore, they are widely used in real application scenarios. Previous interactive recommendation methods primarily focus on learning users' personalized preferences on the relevance properties of an item set. However, the investigation of users' personalized preferences on the diversity properties of an item set is usually ignored. To overcome this problem, we propose the Linear Modular Dispersion Bandit (LMDB) framework, which is an online learning setting for optimizing a combination of modular functions and dispersion functions. Specifically, LMDB employs modular functions to model the relevance properties of each item, and dispersion functions to describe the diversity properties of an item set. Moreover, we also develop a learning algorithm, called Linear Modular Dispersion Hybrid (LMDH) to solve the LMDB problem and derive a gap-free bound on its n -step regret. Extensive experiments on real datasets are performed to demonstrate the effectiveness of the proposed LMDB framework in balancing the recommendation accuracy and diversity.

Introduction

The development of interactive recommender systems (IRS) is necessary to capture user's preferences in time. One of the powerful tools is the multi-armed bandit (MAB) which has been extensively studied and applied in different machine learning tasks (Bubeck and Cesa-Bianchi 2012). MAB models an agent that simultaneously attempts to acquire knowledge and optimizes its decisions based on existing knowledge about the environment, during a sequential decision-making process. Assume there are m arms (or actions) presented to a decision-maker on every of n rounds. The objective of the decision-maker is to maximize the sum of rewards over all the n rounds (Agrawal, Hedge, and Tenenbaum 1988). Contextual bandit is a type of MAB framework, where each arm is represented by a feature vector (Chu et al. 2011; Li et al. 2010). In the literature, contextual bandit has been proven to be powerful for solving large scale interactive recommendation problems (Li et al. 2010; Wen, Kveton, and Ashkan 2015) by adopting the user's timely feedback.

In practice, the user's implicit feedback (e.g., click) is commonly used to build the recommender systems, as it can be easily collected (Shi, Larson, and Hanjalic 2014). Especially, for online applications, this type of feedback becomes an increasingly important source of data to study the user behaviors (Liu et al. 2015, 2018; Yang et al. 2018). However, the implicit feedback does not explicitly describe the user preferences. Thus, it is difficult for non-interactive recommender systems to accurately estimate the user's interests. In IRS, the recommendation agents can interact with the users. They can explore the user's interests by sequentially recommending a set of items to her and collecting her implicit feedback in multiple rounds.

Previous contextual bandit based IRS mainly focus on optimizing the accuracy of recommendation results by learning the user preferences on the relevance features of items (Chu et al. 2011; Li et al. 2010). They usually ignore the diversity of recommendation results, thus are more likely to result in similar item recommendations. A relevant but redundant item set would make the user feel bored with the recommendation result. On the other hand, although there exist some IRS considering diversity in bandit frameworks (Liu et al. 2020; Qin, Chen, and Zhu 2014), they do not learn the user's personalized preferences on the diversity properties of a set of items. In fact, different users may have different preferences with the diversity properties of an item set. For example, a user with specific interest may prefer a relevant item set than a diverse item set, while a user without specific interest may prefer a diverse item set to explore her interests.

In this paper, we propose a novel bandit learning framework for IRS, which considers the user's preferences on both the item relevance features and the diversity features of the item set. The objective is to achieve a good trade-off between the accuracy and diversity of recommendation results. Specifically, we use modular functions and dispersion functions to model the relevance features and diversity features, respectively. The dispersion function has been shown to be effective in describing the item set diversity when building the non-interactive recommender system (Sha, Wu, and Niu 2016). We employ it in our bandit framework to develop the diversified IRS. In summary, the main contributions made in this work are as follows: (1) we propose a novel bandit learning framework, namely Linear Modular Dispersion Bandit (LMDB), for diversified interactive recommendation; (2) we

*Corresponding author.

propose a Linear Modular Dispersion Hybrid (LMDH) algorithm to solve the LMDB problem; (3) we also provide theoretical analysis for the algorithm and derive a gap-free upper bound on its scaled n -step regret; (4) we perform extensive experiments on simulated datasets and real datasets to verify the effectiveness of the proposed LMDB framework in balancing the recommendation accuracy and diversity.

Related Work

Diversified Recommendation

The diversity promoting recommendation methods are usually based on greedy heuristics (Wu et al. 2019b). For example, the maximal marginal relevance (MMR) method (Carbonell and Goldstein 1998) selects a list of items by maximizing a score function defined as a combination of relevance and diversity metrics. In (Qin and Zhu 2013), an entropy regularizer which quantifies the posterior uncertainty of ratings of items in a set is derived for promoting the diversity of recommendation results. This entropy regularizer is a monotone submodular function, and there exists a greedy algorithm which is guaranteed to produce a near optimal solution for monotone submodular function within a factor $(1 - \frac{1}{e})$ of optimal (Nemhauser, Wolsey, and Fisher 1978). Determinantal point process (DPP) (Kulesza, Taskar et al. 2012) has been studied for diversified recommendation problems (Chen, Zhang, and Zhou 2018; Wilhelm et al. 2018; Wu et al. 2019a), and it can also be formulated as a monotone submodular maximization problem (Gillenwater, Kulesza, and Taskar 2012). Besides these submodular functions, dispersion function that needs less computation has also been studied for diversified recommendation (Gollapudi and Sharma 2009; Sha, Wu, and Niu 2016). The above methods are developed for building the non-interactive recommendation systems. For more details of diversity promoting recommendation methods, you can also refer to (Chapelle et al. 2011; Vargas et al. 2014; Puthiya Parambath, Usunier, and Grandvalet 2016; Parambath 2019).

Interactive Recommendation

In the literature, bandit frameworks have been widely studied for building interactive recommender systems. In non-contextual bandit methods, the learner cannot access the arm features. Under this setting, the upper confidence bound (UCB) algorithm is proven to be theoretically optimal (Auer, Cesa-Bianchi, and Fischer 2002). However, without using arm features, the performances of UCB algorithm are quite limited in many applications. In contextual bandit methods, the learner can observe the arm features. LinUCB (Li et al. 2010) is a representative contextual bandit algorithm proposed for a personalized news recommendation. It assumes the agent recommends a single arm in each round and focuses on optimizing the recommendation accuracy. For large scale combinatorial multi-armed bandit problems, the performances of non-contextual methods can be suboptimal, because the agents do not use arm features (Chen, Wang, and Yuan 2013). To improve the performances, (Wen, Kveton, and Ashkan 2015) introduces an efficient learning framework that exploits the arm features and considers the

modular function maximization problem. To support the diversified online recommendation, (Yue and Guestrin 2011) propose a linear submodular bandit to optimize a class of submodular utility functions. (Hiranandani et al. 2020) extend it to a cascading setting and develop linear cascading submodular bandit. (Li, Feng, and Rijke 2020) develop cascading hybrid bandit by combining a modular function and a submodular function. (Qin, Chen, and Zhu 2014) propose a contextual combinatorial bandit framework based on the entropy regularizer (Qin and Zhu 2013). In addition, (Liu et al. 2020) develops a full Bayesian online recommendation framework to balance the recommendation accuracy and diversity, by integrating DPP with Thompson sampling (Chapelle and Li 2011). Overall, these methods are developed based on submodular functions. Due to the inefficiency of computing DPP kernel or entropy regularizer, we find dispersion functions are more suitable for our task. In this work, the proposed LMDB framework considers both relevance features and diversity features of arms, and is formulated as an optimization of the combination of modular functions and dispersion functions.

Problem Formulation

In this section, we firstly describe the properties of our utility function. Throughout this paper, we use personalized item recommendation as our motivating example. Under this setting, utility corresponds to the relevance properties and diversity properties of the recommended item set. Suppose that we recommend a set of items only based on their relevance properties, then the user may be bored with the recommendation results as the items are likely to be very similar. Thus, we should consider both the relevance and diversity properties of the item set to attract users. We formulate this task as a combinatorial optimization problem. Suppose that $E = \{1, \dots, L\}$ is a set of L items, called the ground set. $\mathcal{A} \subseteq \{A \subseteq E : |A| \leq K\}$ is a family of subsets of E with up to K items. Given the user's preferences $\eta = [\theta^T, \beta^T]^T \in \mathbb{R}^{d+m}$, where $\theta \in \mathbb{R}^d$ are the user's preferences on the relevance properties and $\beta \in \mathbb{R}^m$ are the user's preferences on the diversity properties. Our objective is to find a set $A \in \mathcal{A}$, which maximizes the following utility function

$$F(A|\eta) = \sum_{i=1}^d \theta_i R_i(A) + \sum_{i=1}^m \beta_i V_i(A), \quad (1)$$

where $R_i(A)$ is a modular function to capture the item's i -th relevance property (e.g., the category of the item) and $V_i(A)$ is a dispersion function to capture the item set's i -th diversity property (e.g., Cosine distance of the items in the set). In fact, we assume different distance metrics can describe different diversity properties of the item set. Hence, we have different definitions of the dispersion function V in Eq. (1).

Modular Function. We firstly introduce the properties of the modular function in Eq. (1). If R is a modular function mapping the set of items A to real values, then we have

$$R(A) = \sum_{i \in A} R(i). \quad (2)$$

From Eq. (2), we can get the marginal gain of a modular function for adding an item a to the set A ,

$$R(A \cup \{a\}) - R(A) = R(a), \quad (3)$$

which is independent with other items. It is usually consistent with the recommendation scenario, as the user considers the relevance property of an item is independent with those of other items in a set. Therefore, we define the item a 's relevance features as follows,

$$\Delta_R(a|A) = \langle R_1(a), \dots, R_d(a) \rangle. \quad (4)$$

Furthermore, a modular function is monotone, if for any subset $A \subseteq B \subseteq E$, $R(A) \leq R(B)$. Actually if $R(a)$ is non-negative for any item $a \in E$, then R is a monotone modular function.

Dispersion Function. We secondly introduce the properties of the dispersion functions in Eq. (1). If V is a dispersion function mapping an item set A to real values, then we have,

$$V(A) = \sum_{\{i,j\}:i,j \in A} h(i,j), \quad (5)$$

where $h(i,j)$ can be any distance metric (e.g., Cosine distance) for items $i, j \in A$. From Eq. (5), we can get the marginal gain of a dispersion function for adding an item a to the set A ,

$$V(A \cup \{a\}) - V(A) = \sum_{j \in A} h(a,j), \quad (6)$$

which is dependent with the items in A . This is also consistent with the recommendation scenario, as the user considers the diversity property of an item a by comparing it with other items in A . Therefore, we define the item a 's diversity features as,

$$\Delta_V(a|A) = \langle \sum_{j \in A} h_1(a,j), \dots, \sum_{j \in A} h_m(a,j) \rangle. \quad (7)$$

Local Linearity. One attractive property of $F(A|\eta)$ is that the incremental gains are locally linear. In particular, the incremental gain of adding a to A can be written as $\eta^T \Delta(a|A)$, where

$$\Delta(a|A) = \langle \Delta_R(a|A), \Delta_V(a|A) \rangle. \quad (8)$$

In other words, $\Delta_R(a|A)$ corresponds to the relevance gain by item a , and $\Delta_V(a|A)$ corresponds to the diversity gain by item a , conditioned on the items having already been selected into A .

Optimization. Another attractive property of our utility function is that if $F(A|\eta)$ can be formulated as the combination of a monotone modular function and a dispersion function, there is a greedy algorithm which is guaranteed to produce a near-optimal solution (Borodin, Lee, and Ye 2012) within a factor $\frac{1}{2}$. However, their greedy algorithm is "non-oblivious" as it is not selecting the next element with respect to the objective function $F(A|\eta)$. Therefore, we propose Algorithm 1, which is a greedy algorithm with the objective function $F(A|\eta)$ for bandit learning setting. The following theorem describes the approximation property of our algorithm.

Algorithm 1 Modular Dispersion Greedy Search

- 1: **Input:** User preferences η , modular functions R_1, \dots, R_d , distance metrics h_1, \dots, h_m , candidate set E , integer K
 - 2: **Output:** Item subset $A \subseteq E$ with $|A| = K$
 - 3: $A \leftarrow \emptyset$
 - 4: **while** $|A| < K$ **do**
 - 5: $a \leftarrow \arg \max_{a \in E-A} \eta^T \Delta(a|A)$
 - 6: $A \leftarrow A \cup \{a\}$
 - 7: **end while**
 - 8: **return** A
-

Theorem 1. *Let E be the underlying ground set. For any $A \subseteq E$, let $R_1(A), \dots, R_d(A)$ be any modular functions, and let h_1, \dots, h_m be any distance metrics for dispersion function. Let $\theta \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^m$ be the user's preferences. We write $\eta = [\theta^T, \beta^T]^T$. The object function $F(A|\eta)$ is defined as Eq. (1). Let A^{greedy} be the solution computed by the Algorithm 1, and A^* be the optimal solution of the constrained optimization problem $\arg \max_{A:|A| \leq K} F(A|\eta)$. If $\theta^T \langle R_1(a), \dots, R_d(a) \rangle \geq 0$ for any element $a \in E$, and β is a non-negative vector, then we have*

$$F(A^{greedy}|\eta) \geq \gamma F(A^*|\eta), \quad (9)$$

where $\gamma = \frac{1}{4}$. In other words, the approximation ratio of the greedy algorithm is γ .

Note that a similar condition of the user preferences can be found in Yue and Guestrin (2011). The condition is reasonable in practice, as the weighted relevance score of each item should be non-negative for a user. In Section , we observe that our approximation ratio is close to 1 in practice, which is significantly better than that suggested in Theorem 1. The proof of Theorem 1 can be found in the Appendix.

Linear Modular Dispersion Bandit

In this section, we present our LMDB framework. Let $E = \{1, 2, \dots, L\}$ be a ground set of L items and $K \leq L$ be the number of recommended items. $\theta^* \in \mathbb{R}^d$ and $\beta^* \in \mathbb{R}^m$ are the user's preferences on relevance features and diversity features respectively, which are unknown to the learning agent. To simplify the notation, we write $\eta^* = [\theta^{*T}, \beta^{*T}]^T$. The relevance properties for each item are given by R_1, \dots, R_d and the distance metrics for diversity properties are given by h_1, \dots, h_m , which are known to the learning agent.

Our learning agent interacts with the user as follows. At time t , a random subset $E_t \subseteq E$ is presented and the agent recommends a list of K items $A_t = (a_1^t, \dots, a_K^t) \in \Pi_K(E_t)$, where a_k^t is the k -th recommended item and $\Pi_K(E_t)$ is the set of all K -permutations of the set E_t . The reward of item a_k at time t , $w_t(a_k^t)$, is a realization of an independent Bernoulli random variable with mean $\eta^{*T} \Delta(a_k^t | \{a_1^t, \dots, a_{k-1}^t\})$. The user examines the list from the first item a_1^t to the last item a_K^t and then provides the feedback (e.g., clicks on or ignores each item). Formally, for

any list of items A_t , the total rewards $f(A_t, w_t)$ can be written as the sum of rewards at each position,

$$f(A_t, w_t) = \sum_{k=1}^{|A_t|} w_t(a_k^t). \quad (10)$$

From the definitions of Eq. (10), we notice that

$$\mathbb{E}[f(A_t, w_t)] = \sum_{k=1}^{|A_t|} \boldsymbol{\eta}^{*T} \Delta(a_k^t | \{a_1^t, \dots, a_{k-1}^t\}). \quad (11)$$

Eq. (11) implies that $\mathbb{E}[f(A_t, w_t)] = F(A_t | \boldsymbol{\eta}^*)$ for F defined as in Eq. (1). Thus, Algorithm 1 can greedily select items to achieve reward within a factor γ of the optimal with perfect knowledge of $\boldsymbol{\eta}^*$. The goal of the learning agent is to maximize the expected cumulative return in n -episodes. The regret is formally defined in Section .

Learning Algorithm

The expected reward of an item is a combination of a modular function and a dispersion function. Thus, we propose a hybrid optimization algorithm to solve the LMDH problem.

The algorithm can access the item relevance feature \mathbf{z}_a and diversity feature \mathbf{x}_a . It does not know the user's preference $\boldsymbol{\theta}^*$ and $\boldsymbol{\beta}^*$, and estimates them through repeated interactions with the user. It also has two tunable parameters $\lambda, \alpha > 0$, where λ is the regularization parameter, and α controls the trade-off between exploitation and exploration. The details of the proposed LMDH algorithm are summarized in Algorithm 2. At the beginning of each step (line 4), LMDH estimates the user's preference as $\hat{\boldsymbol{\theta}}_t$ and $\hat{\boldsymbol{\beta}}_t$ based on the observations of previous $t - 1$ steps. The estimation can be viewed as a least-squares problem and the derivation of line 4 is based on the matrix block-wise inversion. Due to the space limitation, we omit the derivation process. $\mathbf{H}_t, \mathbf{B}_t, \mathbf{M}_t$ and $\mathbf{u}_t, \mathbf{y}_t$ summarize the features and click feedback of all observed items in the previous $t - 1$ steps, respectively. In the second stage (lines 5-14), LMDH recommends a list of items sequentially from the subset $E_t \subseteq E$. The list is generated by the greedy algorithm (i.e., Algorithm 1), where the reward of an item a is overestimated as lines 8-10 by the upper confidence bound (UCB) (Auer, Cesa-Bianchi, and Fischer 2002). In the last stage (lines 15-22), LMDH displays the item list A_t to the user and collects her feedback, which is used to update the statistics. The per-step computational complexity of LMDH is $O(LK(d^2 + m^2 + md))$. Computing matrix inverse is the main computational cost of Algorithm 2. In practice, we update \mathbf{H}_t^{-1} and \mathbf{M}_t^{-1} instead of \mathbf{H}_t and \mathbf{M}_t (Golub and Van Loan 2012), which is $O(m^2 + d^2)$. LMDH selects K items out of L in $O(LK(d^2 + m^2 + md))$, because computing UCB for each item requires $O(d^2 + m^2 + md)$.

Regret Analysis

This section provides the regret analysis for our proposed algorithm LMDH. Let $\gamma = \frac{1}{4}$, A_t^* be the optimal solution for

Algorithm 2 Linear Modular Dispersion Hybrid (LMDH)

```

1: Inputs: Parameters  $\lambda > 0$  and  $\alpha > 0$ 
2:  $\mathbf{H}_1 \leftarrow \lambda \mathbf{I}_d, \mathbf{u}_1 \leftarrow \mathbf{0}_d, \mathbf{M}_1 \leftarrow \lambda \mathbf{I}_m, \mathbf{y}_1 \leftarrow \mathbf{0}_m, \mathbf{B}_1 \leftarrow \mathbf{0}_{d \times m}$ 
3: for  $t = 1, 2, \dots, n$  do
4:    $\hat{\boldsymbol{\theta}}_t \leftarrow \mathbf{H}_t^{-1} \mathbf{u}_t, \hat{\boldsymbol{\beta}}_t \leftarrow \mathbf{M}_t^{-1} (\mathbf{y}_t - \mathbf{B}_t^T \hat{\boldsymbol{\theta}}_t)$ 
5:    $A_t \leftarrow \emptyset$ 
6:   for  $k = 1, 2, \dots, K$  do
7:     for all  $a \in E_t \setminus A_t$  do
8:        $\mathbf{z}_a \leftarrow \Delta_R(a | A_t), \mathbf{x}_a \leftarrow \Delta_V(a | A_t)$ 
9:        $v_a \leftarrow \mathbf{z}_a^T \mathbf{H}_t^{-1} \mathbf{z}_a - 2 \mathbf{z}_a^T \mathbf{H}_t^{-1} \mathbf{B}_t \mathbf{M}_t^{-1} \mathbf{x}_a + \mathbf{x}_a^T \mathbf{M}_t^{-1} \mathbf{x}_a + \mathbf{x}_a^T \mathbf{M}_t^{-1} \mathbf{B}_t^T \mathbf{H}_t^{-1} \mathbf{B}_t \mathbf{M}_t^{-1} \mathbf{x}_a$ 
10:       $\mu_a \leftarrow \hat{\boldsymbol{\theta}}_t^T \mathbf{z}_a + \hat{\boldsymbol{\beta}}_t^T \mathbf{x}_a + \alpha \sqrt{v_a}$ 
11:    end for
12:     $a_k^t \leftarrow \arg \max_{a \in E_t \setminus A_t} \mu_a$ 
13:     $A_t \leftarrow A_t \cup \{a_k^t\}$ 
14:  end for
15:  Recommend  $A_t$  with the order  $(a_1^t, \dots, a_K^t)$ , and observe the reward  $\{w_a\}_{a \in A_t}$ 
16:   $\mathbf{H}_t \leftarrow \mathbf{H}_t + \mathbf{B}_t \mathbf{M}_t^{-1} \mathbf{B}_t^T$ 
17:   $\mathbf{u}_t \leftarrow \mathbf{u}_t + \mathbf{B}_t \mathbf{M}_t^{-1} \mathbf{y}_t$ 
18:   $\mathbf{M}_{t+1} \leftarrow \mathbf{M}_t + \sum_{a \in A_t} \mathbf{x}_a \mathbf{x}_a^T$ 
19:   $\mathbf{B}_{t+1} \leftarrow \mathbf{B}_t + \sum_{a \in A_t} \mathbf{z}_a \mathbf{x}_a^T$ 
20:   $\mathbf{y}_{t+1} \leftarrow \mathbf{y}_t + \sum_{a \in A_t} w_a \mathbf{x}_a^T$ 
21:   $\mathbf{H}_{t+1} \leftarrow \mathbf{H}_t + \mathbf{z}_a \mathbf{z}_a^T - \mathbf{B}_{t+1} \mathbf{M}_{t+1}^{-1} \mathbf{B}_{t+1}^T$ 
22:   $\mathbf{u}_{t+1} \leftarrow \mathbf{u}_t - \mathbf{B}_{t+1} \mathbf{M}_{t+1}^{-1} \mathbf{y}_{t+1}$ 
23: end for

```

Eq. (11) at time t , and A_t be the solution selected from Algorithm 2. The γ -scaled n -step regret is defined as follows,

$$R^\gamma(n) = \sum_{t=1}^n \mathbb{E}[F(A_t^* | \boldsymbol{\eta}^*) - F(A_t | \boldsymbol{\eta}^*) / \gamma], \quad (12)$$

where $1/\gamma$ accounts for the fact that A_t is a γ -approximation at any time t . Similar definitions of this scaled regret can be found in (Chen et al. 2016; Wen et al. 2017). This definition is reasonable, because it is computationally inefficient to get the optimal solution A_t^* of this problem, even for the offline variant setting. We now state our main result, which essentially bounds the greedy regret.

Theorem 2. *Under LMDH, for any $\lambda \geq 0$, any $\delta \in (0, 1)$, any $\|\boldsymbol{\eta}^*\|_2 \leq 1$, and any*

$$\alpha \geq \sqrt{(d+m) \log \left[1 + \frac{nK}{(d+m)\lambda} \right] + 2 \log \left(\frac{1}{\delta} \right)} \|\boldsymbol{\eta}^*\|_2 + \lambda^{\frac{1}{2}} \quad (13)$$

in Algorithm 2, we have

$$R^\gamma(n) \leq \frac{2\alpha K}{\gamma} \sqrt{\frac{n(d+m) \log \left[1 + \frac{nK}{(d+m)\lambda} \right]}{\lambda \log(1 + \frac{1}{\lambda})}} + nK\delta. \quad (14)$$

Note that if we choose $\lambda = 1$, $\delta = \frac{1}{nK}$, and α as the lower bound specified in Eq. (13), then the regret bound in Theorem 2 is $\tilde{O}(K(d+m)\sqrt{n}/\gamma)$. The factor \sqrt{n} is considered near-optimal in gap-free regret bounds. The factor $d+m$ is the number of features, which is standard in linear bandits (Abbasi-Yadkori, Pál, and Szepesvári 2011). The factor $\frac{1}{\gamma}$ is due to the fact that A_t is a γ -approximation. The factor K is because of the number of recommended items for each trial. The proof of Theorem 2 can be found in the Appendix.

Experiments

In this section, we perform two kinds of numerical experiments to verify the performance of our learning algorithm. In the first experiment, we show our algorithm outperforms other baseline methods for the task of balancing the accuracy and diversity of the recommendation result on real-world datasets. In the second experiment, we compare the regret of our algorithm with other baseline methods to verify our Theorem 2 on simulated datasets and validate the approximation ratio of Algorithm 1.

Experiments on Real-world Datasets

The experiments are performed on the following datasets: Movielens-100K, Movielens-1M¹, and Anime². Movielens-100K contains 100,000 ratings given by 943 users to 1682 movies, and Movielens-1M contains 1,000,209 ratings given by 6,040 users to 3,706 movies. Anime contains 7,813,737 ratings given by 73,515 users to 11,200 animes. Following (Liu et al. 2020), we keep the ratings larger than 3 as positive feedback on Movielens-100K and Movielens-1M, and keep the ratings larger than 6 as positive feedback on Anime. We use the unbiased offline evaluation strategy (Li et al. 2011) to evaluate different recommendation methods. Firstly, we randomly split each dataset into two non-overlapping sets. Specifically, 80% of the users are used for training, and the remaining 20% users are used for testing. After that, BPRMF (Rendle et al. 2009) is employed to learn the embeddings of all items from the training data. Empirically, we set the dimensionality of the item embeddings to 10. The item embeddings are then normalized into a bounded space $[-1, 1]^{10}$, which is considered as the relevance features \mathbf{z}_a of an item a . For fair comparison, we only use the average Cosine distance as the distance metric of the dispersion function, which is defined by $\frac{2}{|A_t|(|A_t|-1)}(1 - \text{sim}(\mathbf{z}_i, \mathbf{z}_j))$, where $\text{sim}(\mathbf{z}_i, \mathbf{z}_j)$ denotes the Cosine similarity between the relevance features of two items i and j in A_t , and $|A_t|$ is the size of the item set A_t . For each trial t , the candidate item set E_t is generated by removing the recommended items of the last $t-1$ steps from the total item set. The statistics of the experimental datasets are summarized in Table 1 in Appendix.

We use Recall(t) to measure the accumulative recommendation accuracy over all the t recommendation rounds, and measure the diversity of recommendation results by Diversity(t) which is defined by

computing the average recommendation diversity over all the t recommendation rounds (Zhang and Hurley 2008). We define $\text{Recall}(t) = \sum_{\ell=1}^t \frac{|A_\ell \cap \mathcal{I}|}{|\mathcal{I}|}$ and $\text{Div.}(t) = \frac{1}{t} \sum_{\ell=1}^t \left[\frac{2}{|A_\ell|(|A_\ell|-1)} \sum_{\{i,j\}:i,j \in A_\ell} (1 - \text{sim}(\mathbf{z}_i, \mathbf{z}_j)) \right]$, where A_ℓ is the recommended item set at the ℓ -th round, and \mathcal{I} denotes the set of items interacted with the user in testing data. We compute the accuracy and diversity for each user and report the averaged values over all users to obtain the accuracy and diversity metrics. F_β -score is used to measure the effectiveness of each method in balancing recommendation accuracy and diversity, where $F_\beta(t) = (1 + \beta^2) * \text{Recall}(t) * \text{Diversity}(t) / [\beta^2 * \text{Diversity}(t) + \text{Recall}(t)]$. β is a constant indicating Recall is considered as β times as important as Diversity. Empirically, we set β to 1 and 2.

We compare LMDB with the following baseline methods: (1) **LogRank**: In this method, the relevance score of each item a is computed as $r_a = 1/(1 + \exp(-\bar{\mathbf{u}}\mathbf{z}_a^T))$, where $\bar{\mathbf{u}}$ is the mean of the user embeddings learnt from the training data. Then, this method selects K items with the highest quality scores as A_t in the t -th recommendation round; (2) **MMR** (Carbonell and Goldstein 1998): For each round t , this method sequentially selects an available item with the largest maximal marginal score, which is defined as $\bar{r}_a = \alpha r_a - \frac{1-\alpha}{|A_t|} \sum_{j \in A_t} \text{sim}(\mathbf{z}_a, \mathbf{z}_j)$, where r_a is the item quality defined in the LogRank method; (3) **ϵ -Greedy**: This method randomly selects an item into the set A_t with probability ϵ , and selects the item with the highest relevance score into the set A_t with probability $1 - \epsilon$, which is defined the same as in LogRank method; (4) **C2UCB** (Qin, Chen, and Zhu 2014): This is a contextual combinatorial bandit with an entropy regularizer for diversity recommendation; (5) **DC2B** (Liu et al. 2020): This is a full Bayesian online recommendation framework developed based on determinantal point process. We set the size of A_t to 10 for all methods. The hyper-parameter settings for each method are summarized in Appendix.

Experiment Results. The recommendation accuracy and diversity of different methods are shown in Figure 1 and Figure 2, respectively. As shown in Figure 1, the proposed LMDB framework consistently achieves the best recommendation accuracy over all 30 recommendation rounds on all datasets. When performing more interactions between the user and the recommender agent, LMDB and C2UCB can gradually achieve more improvements over other methods. For DC2B, it is effective in the first few recommendation rounds. However, its accuracy then becomes poorer than other methods with the increase of the number of recommendation rounds. Moreover, from Figure 2, we can note that LMDB usually achieves the second or third best recommendation diversity on all datasets. Over all recommendation rounds, LMDB consistently achieves better recommendation diversity than C2UCB, and it can usually generate more diverse recommendation results than DC2B in the first 15 rounds.

For better understanding of the results, F_β -score is used to study the effectiveness of each method in balancing the recommendation accuracy and diversity. Figure 3 summarizes

¹<https://grouplens.org/datasets/movielens/>

²<https://www.kaggle.com/CooperUnion/anime-recommendations-database>

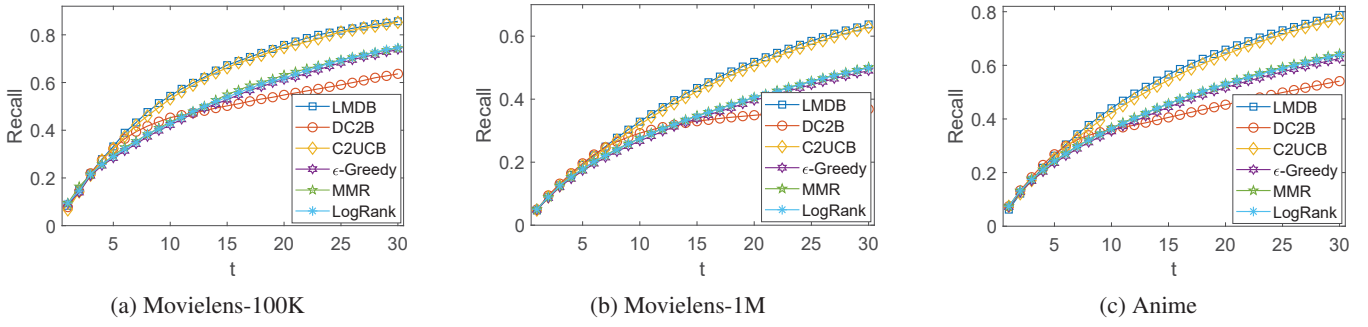


Figure 1: The recommendation performances of different methods measured by Recall.

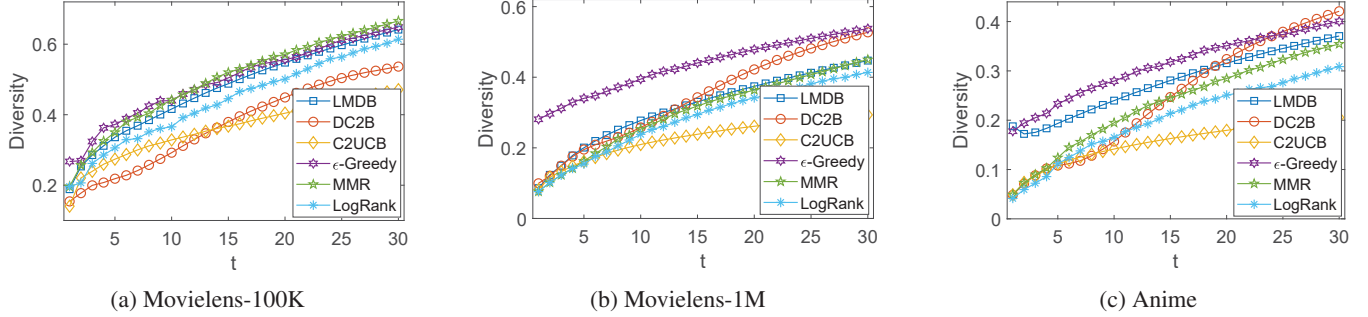


Figure 2: The recommendation performances of different methods measured by Diversity.

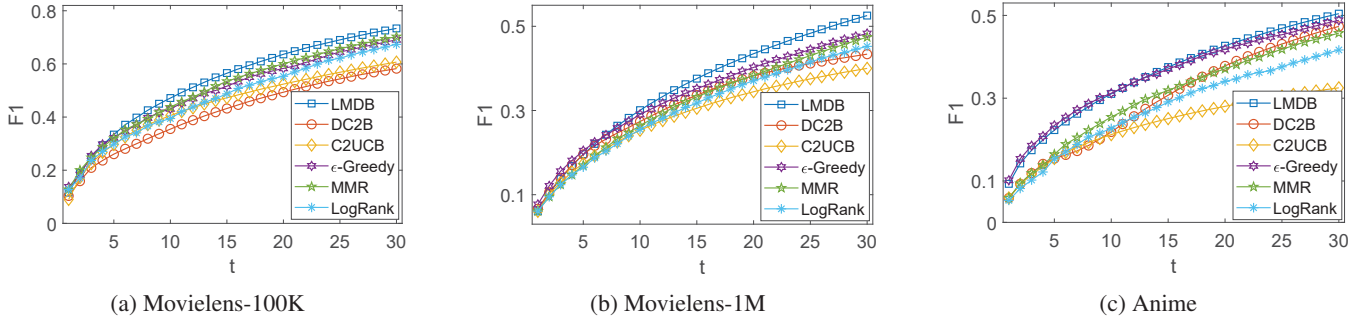


Figure 3: The recommendation performances of different methods measured by F_1 -score.

the F_1 -score achieved by different methods. As shown in Figure 3(a) and 3(b), LMDB usually outperforms other baselines over all recommendation rounds, in terms of F_1 -score. From Figure 3(c), we can also notice that LMDB achieves comparable F_1 -score with ϵ -Greedy and better F_1 -score than other baseline methods. ϵ -Greedy is a simple exploration method that can usually achieve high recommendation diversity. Thus, it may also have a high F_1 -score. In practice, the recommendation accuracy is usually more important than recommendation diversity. Considering this factor, we also report the recommendation performances in terms of F_2 -score, which treats Recall as 2 times as important as Diversity. The results shown in Figure 1 in Appendix indicate that LMDB achieves the best F_2 -score on all datasets over all recommendation rounds. These results demonstrate that the proposed LMDB framework is more effective in balancing

the recommendation accuracy and diversity than baseline methods, especially when we place more emphasis on recommendation accuracy than recommendation diversity. We believe this is because only LMDB learns the user’s personalized preferences across the relevance properties and diversity properties of the recommended item set.

Moreover, we also evaluate the impacts of λ , α , and the size of recommended item set $|A_t|$ on the performances of LMDB. As shown in Figure 4(a) and 4(b), we can note that the values of λ and α do not have significant impacts on the performances of LMDB, when λ and α are equal to or larger than 1. In addition, we also vary $|A_t|$ in $\{5, 10, 15\}$. Considering both Figure 3(a) and 4(c), we can note that the proposed LMDB framework consistently outperforms other combinatorial bandit learning methods C2UCB and DC2B in terms of F_1 -score, with different settings of $|A_t|$. These

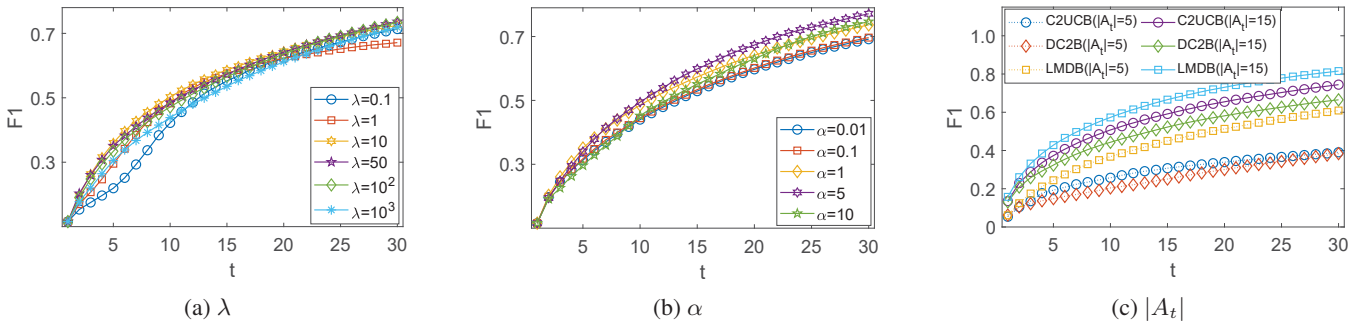


Figure 4: The F_1 -score of LMDB w.r.t. different settings of λ , α , and $|A_t|$ on Movielens-100K.

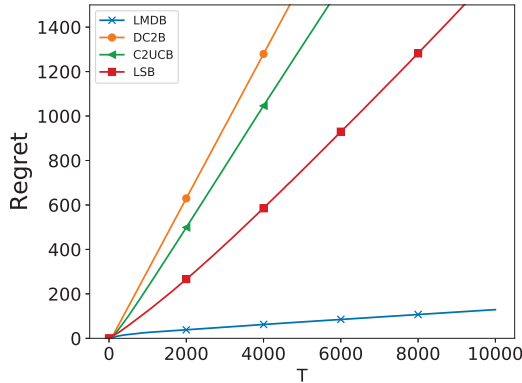


Figure 5: Regret of different bandit methods on simulated dataset.

results demonstrate that LMDB are stable and insensitive to the settings of hyper-parameters.

Experiment on Simulated Dataset

This experiment is to verify the regret in Theorem 2 and compare the result of our Algorithm 2 with other algorithms. Here, we only consider the bandit based methods including C2UCB, DC2B and LSB (Yue and Guestrin 2011). We already describe C2UCB and DC2B in Section . LSB is a linear contextual bandit, which optimizes a class of submodular utility functions for diversified retrieval. The item’s feature of LSB is extracted from the category information of the item, which is not consistent with the setting of Section . Hence, we only evaluate it for this experiment setting. We consider a setting, where there are 20 candidate items. The objective is to recommend a relevant and diverse item list to a user. The list size K is empirically set to 5. The relevance features of each item are uniformly generated from $[0, 0.5]^{10}$. Here, we only use one-dimensional diversity feature. The distance metric $h(i, j)$ of the dispersion function is defined the same as in Section . The user’s preferences on the relevance properties and diversity property are generated from $[0, 0.2]^{10}$ and $[0, 0.2]$, respectively. We run 20 independent simulations and compute the average regret over them.

Figure 5 compares the performances of our algorithm

with the baselines. The average regret of LMDB flattens with the number of trials T and is much smaller than other baselines. Although all of the three baselines consider the diversity of the recommendation, they never learn the user’s preferences on the diversity properties of the recommended item set. Hence, they recommend a diverse item set but possibly not suitable for the user.

In Section , we propose the greedy Algorithm 1 to compute a near-optimal set for our optimization problem (Eq. (1)), which can reach the approximation ratio at $\frac{1}{4}$ of the optimal solution. Here, we empirically demonstrate that the approximation ratio is close to 1 in a domain of our interest. We choose 100 random users and 20 random items and vary the number of recommended items K from 2 to 5. The users’ preferences and items’ features are generated from the uniform distribution stated above. For each user and K , we compute the optimal list by exhaustive search and the sub-optimal list by Algorithm 1. After that, we compute the approximation ratio $F(A^{greedy}|\eta)/F(A^*|\eta)$, where $F(\cdot)$ is defined in Eq. (1). We find that the average approximation ratios over all users are 0.9995, 0.9992, 0.9989, and 0.9971 when K is set to 2, 3, 4, and 5 respectively. This indicates that the approximation ratio in practice may be much better than theoretical result.

Conclusion and Future Work

This paper introduces a novel bandit model named LMDB (i.e., Linear Modular Dispersion Bandit) for optimizing a combination of modular functions and dispersion functions. This setting is useful for building the diversified interactive recommender systems that interactively learn the user’s preferences from her timely implicit feedback. Specifically, LMDB employs modular functions and dispersion functions to describe the relevance properties and diversity properties of the recommended item list, respectively. Moreover, we propose the linear modular dispersion hybrid algorithm to solve the LMDB problem, and derive a gap-free upper bound on its scaled n -step regret. We conduct empirical experiments on three datasets and find that the proposed LMDB model outperforms existing bandit learning methods in balancing the recommendation accuracy and diversity. For future work, we would like to improve our analysis method to get a better approximation ratio of the optimal solution in the bandit setting.

Acknowledgments

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (Alibaba-NTU-AIR2019B1), Nanyang Technological University, Singapore. This research is also supported, in part, by the National Research Foundation, Prime Minister's Office, Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-003) and under its NRF Investigatorship Programme (NRFI Award No. NRF-NRFI05-2019-0002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. This research is also supported, in part, by the Singapore Ministry of Health under its National Innovation Challenge on Active and Confident Ageing (NIC Project No. MOH/NIC/COG04/2017 and MOH/NIC/HAIG03/2017).

Ethical Impact

This work has the following potential positive impacts for the companies which need recommender systems. We propose a diversified recommendation framework, which achieves good performance on the trade-off between the accuracy and diversity of recommendation results. It can be beneficial to show more different products to the users while keeping them in the system. At the same time, this work may have some negative consequences. The accuracy of the recommendation results would be reduced by improving the diversity of recommendation results. Thus, this will increase the risk that users may leave the system. Furthermore, our testing is based on an offline evaluation strategy (Li et al. 2011). A direct application to online recommendation scenarios has the probability that users are not satisfied with the recommendation results. Therefore, more work is needed to test the performance of our recommendation framework on real recommendation scenarios.

References

- Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.
- Agrawal, R.; Hedge, M.; and Teneketzis, D. 1988. Asymptotically efficient adaptive allocation rules for the multi-armed bandit problem with switching cost. *IEEE Transactions on Automatic Control* 33(10): 899–906.
- Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3): 235–256.
- Borodin, A.; Lee, H. C.; and Ye, Y. 2012. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, 155–166. ACM.
- Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Machine Learning* 5(1): 1–122.
- Carbonell, J.; and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336. ACM.
- Chapelle, O.; Ji, S.; Liao, C.; Velipasaoglu, E.; Lai, L.; and Wu, S.-L. 2011. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval* 14(6): 572–592.
- Chapelle, O.; and Li, L. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, 2249–2257.
- Chen, L.; Zhang, G.; and Zhou, E. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, 5622–5633.
- Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, 151–159.
- Chen, W.; Wang, Y.; Yuan, Y.; and Wang, Q. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research* 17(1): 1746–1778.
- 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.
- Gillenwater, J.; Kulesza, A.; and Taskar, B. 2012. Near-optimal map inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, 2735–2743.
- Gollapudi, S.; and Sharma, A. 2009. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, 381–390. ACM.
- Golub, G. H.; and Van Loan, C. F. 2012. *Matrix computations*, volume 3. JHU press.
- Hiranandani, G.; Singh, H.; Gupta, P.; Burhanuddin, I. A.; Wen, Z.; and Kveton, B. 2020. Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In *Uncertainty in Artificial Intelligence*, 722–732. PMLR.
- Kulesza, A.; Taskar, B.; et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5(2–3).
- Li, C.; Feng, H.; and Rijke, M. d. 2020. Cascading hybrid bandits: Online learning to rank for relevance and diversity. In *Fourteenth ACM Conference on Recommender Systems*, 33–42. ACM.
- 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. ACM.
- Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. 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*, 297–306. ACM.
- Liu, Y.; Xiao, Y.; Wu, Q.; Miao, C.; Zhang, J.; Zhao, B.; and Tang, H. 2020. Diversified Interactive Recommendation with Implicit Feedback. In *Proceedings of 34th AAAI Conference on Artificial Intelligence*, 4932–4939. AAAI Press.
- Liu, Y.; Zhao, L.; Liu, G.; Lu, X.; Gao, P.; Li, X.-L.; and Jin, Z. 2018. Dynamic Bayesian logistic matrix factorization for recommendation with implicit feedback. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3463–3469. AAAI Press.
- Liu, Y.; Zhao, P.; Sun, A.; and Miao, C. 2015. A boosting algorithm for item recommendation with implicit feedback. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1792–1798. AAAI Press.
- Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14(1): 265–294.
- Parambath, S. A. P. 2019. Re-ranking Based Diversification: A Unifying View. *arXiv preprint arXiv:1906.11285*.
- Puthiya Parambath, S. A.; Usunier, N.; and Grandvalet, Y. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 15–22. ACM.
- Qin, L.; Chen, S.; and Zhu, X. 2014. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, 461–469. SIAM.
- Qin, L.; and Zhu, X. 2013. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2698–2704. AAAI Press.
- Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 452–461. ACM.
- Sha, C.; Wu, X.; and Niu, J. 2016. A Framework for Recommending Relevant and Diverse Items. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3868–3874. AAAI Press.
- Shi, Y.; Larson, M.; and Hanjalic, A. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47(1): 1–45.
- Vargas, S.; Baltrunas, L.; Karatzoglou, A.; and Castells, P. 2014. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proceedings of the 8th ACM Conference on Recommender systems*, 209–216. ACM.
- Wen, Z.; Kveton, B.; and Ashkan, A. 2015. Efficient learning in large-scale combinatorial semi-bandits. In *International Conference on Machine Learning*, 1113–1122.
- Wen, Z.; Kveton, B.; Valko, M.; and Vaswani, S. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. In *Advances in neural information processing systems*, 3022–3032.
- Wilhelm, M.; Ramanathan, A.; Bonomo, A.; Jain, S.; Chi, E. H.; and Gillenwater, J. 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2165–2173. ACM.
- Wu, Q.; Liu, Y.; Miao, C.; Zhao, B.; Zhao, Y.; and Guan, L. 2019a. PD-GAN: adversarial learning for personalized diversity-promoting recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3870–3876. AAAI Press.
- Wu, Q.; Liu, Y.; Miao, C.; Zhao, Y.; Guan, L.; and Tang, H. 2019b. Recent advances in diversified recommendation. *arXiv preprint arXiv:1905.06589*.
- Yang, P.; Zhao, P.; Liu, Y.; and Gao, X. 2018. Robust cost-sensitive learning for recommendation with implicit feedback. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 621–629. SIAM.
- Yue, Y.; and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, 2483–2491.
- Zhang, M.; and Hurley, N. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, 123–130. ACM.