SWCB: An Efficient Switch-Clustering of Bandit Model

Xiangyu Xu[a, b], Qifeng Zhou[a, b*], Qing Wang[c], Langcai Cao[a, b]
[a] Department of Automation, Xiamen University, Xiamen 361005, China
[b] Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-making, Xiamen 361005, China
[c] Department of Cognitive Service Foundations, IBM T.J. Watson Research Center, NY, USA
{Xiangyu Xu}@xuxiangyu0209@stu.xmu.edu.cn, {Qifeng Zhou}@zhouqf@xmu.edu.cn

2024 36th Chinese Control and Decision Conference (CCDC) | 979-8-3503-8778-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCDC62350.2024.10587841

***Abstract*—Bandit model is a general framework for solving the cold-start problem in recommender systems. In recent works, clustering strategies have been adopted in the bandit settings to further mitigate the lack of information for new users and items. Although these approaches prove to be effective, their performance may suffer from the clustering efficiency. On one hand, quickly identifying the underlying cluster to share the information has a significantly impact on retaining new users with low levels of interaction in the early stages of recommendation. On the other hand, problems like overfitting might result from repeatedly using the same clustering algorithm on the same set of users, which may decrease the recommended performance over an extended period in the later stages. To address these issues, we propose a Switch-Clustering of Bandit (SWCB), which utilizes the Bandit framework to enable online recommendation as well as new user clustering in the cold-start phase. In particular, the proposed model improves the efficiency of recommendations by switching the clustering algorithms automatically. SWCB can quickly identify the underlying clusters by splitting and merging operation. When clustering metric updated by Bandit reaches a certain threshold, SWCB alters the way of user clustering, thus speeding the information retrieval in the cold start stage and helping to retain new users of online recommendations. We evaluate the efficiency and effectiveness of the proposed model on one synthetic dataset and two real-world datasets. Compared with the baseline bandit models, SWCB gets higher rewards within the rounds of the cold-start phase.**

***Index Terms*—Recommended System, Bandit, Cold Start, Clustering**

I. Introduction

Nowadays, recommender systems have become one of the most popular applications for users filtering information and obtaining personalized suggestions [1]. With the rise of application complexity, traditional recommender systems that suppose user preferences and item lists are static no longer meet real-world dynamic scenarios such as providing decision-making strategies for uses in E-Commerce. Therefore, online recommender systems that are able to analyze users' real-time feedback start to draw more attentions. Meanwhile, with the new users arrival, due to the lack of the users' past interaction information with items, online recommender systems meet a big challenge, that is, the cold start problem.

Multi-Armed Bandit (MAB) as a powerful framework for studying decision-making over time under uncertainty has been extended for solving the cold start problem [2]. It is an effective solution addressing the Exploration-Exploitation (EE) dilemma in recommender systems [3]. By leveraging the MAB, the system can balance the trade-off between exploring new items or exploiting already known items to maximize rewards. MAB-based algorithms can make appropriate recommendations to new users even when there is little information available about their preferences [4].

In order to fully utilize the information of users and items to mitigate the cold-start problem, researchers have integrated clustering algorithms into the Bandit framework to share the feature information [6]. In practical scenarios, clustering efficiency is one of important factors for retaining new users with little interactive information in the early stages of recommendation and enhancing the recommendation performance over an extended period in the later stages [5].

However, the state-of-the-art clustering-based bandits pay more attention to improving the bandit performance and ignore the efficiency of clustering algorithms, which may also decrease the recommended performance due to several reasons. One is if a clustering algorithm spends too much time in the cold start phase, it will slow down the recommendation efficiency and reduce the feedback of Bandit within a certain user's attention time. Second, existing clustering-based Bandit models usually use one clustering algorithm repeatedly to find similar users or items in the whole cold start process [7], then the clustering results will remain the same or become less informative over time, this may prevent discovering the changes of user behavior and the clustering algorithm may become too closely tailored to the specific set of users and decrease the generalization ability of recommendation systems. Meanwhile, under a dynamically evolving process of online recommendations, using the same clustering algorithm may create a bias towards specific features and can not capture the full diversity and complexity of the user groups.

To solve the aforementioned problems, in this paper we propose a model called SWCB (Switching Upper Confidence Bound), which utilizes the bandit framework to enable online recommendation as well as efficient clustering of new users in the cold-start phase. We first design a splitting-and-emerging clusters module to speed up the process of identifying the underlying user clusters. Then when clustering metrics updated by Bandit reaches a certain threshold, SWCB switches the clustering algorithm to further accelerate the user clustering process and prevent the negative effects of overfitting re-

sulting from repeatedly using the same clustering algorithm on the same set of users. We conduct emperical studys on one synthetic dataset and two real-world applications, and the experimental results demonstrate the effectiveness of our proposed approach.

II. Related Work

UCB is a popular and effective algorithm for multi-armed bandit problems. It addresses the exploration-exploitation dilemma by balancing the exploitation of actions with high expected rewards and the exploration of actions with uncertain rewards. UCB algorithm calculates an upper confidence bound for each action based on its estimated mean reward and a measure of its uncertainty. And then selects the action with the highest upper confidence bound, which encourages exploration of actions with high uncertainty and exploitation of actions with high expected rewards. As more data is collected, the algorithm updates the estimated mean reward and the uncertainty measure for each action, leading to more informed decisions over time.

Researchers have found that dividing users into different clusters and customizing the bandits for each cluster can enhance the bandit model's performance by leveraging user features to identify clusters of similar users [8].

Online Clustering of Bandits (CLUB) is a bandit algorithm for adaptive clustering of users based on graph [9]. CLUB treats each user as a node in the graph and initializes a cluster of all users, which is represented by an undirected complete graph. Users are clustered together through bandit parameters and their similarities are described through a graph where users with similar interests are connected by edges. By removing edges between users who are no longer similar in their interests, the initial undirected graph is partitioned into several clusters.

Dynamic Clustering of Contextual Multi-Armed Bandits (DynUCB) divides the population of users into multiple clusters and customizes the bandits to each cluster [10], [11]. Users can shift between groups when their interest preferences have altered. Compared to CLUB, DynUCB enables users to migrate between clusters instead of splitting clusters.

Dynamic Clustering based Contextual Combinatorial Multi-Armed Bandits (DCMAB) based Contextual Combinatorial Multi-Armed Bandits, which consists of three configurable key components [12]. Specifically, a dynamic user clustering strategy enables different users in the same cluster to cooperate in estimating the expected rewards of arms. A dynamic item partitioning approach based on collaborative filtering significantly reduces the scale of arms and produces a recommendation list instead of one item to provide diversity. In addition, a multi-class reward mechanism based on fine-grained implicit feedback helps better capture user preferences.

In these clustering-based Bandit models, clustering speed has big impact on the phase of cold start. However, these bandit models ignore the efficiency of clustering algorithms. In this paper, we propose SWCB which evaluates the quality of clustering and adaptively switches the clustering algorithm to capture the diversity and complexity of users and speed up the cold start process. The quality of clustering and switching thresholds are computed based on parameters updated by user feedback from bandit.

III. Problem Definition

In this section, we formulate the contextual bandit problem. The specific description is as follows.

Let $N = \{u_1, u_2, ..., u_n\}$ represents a set of n users in the contextual bandit problem. The learning agent chooses a user for each round $t = \{1, 2, ...T\}$ and receives a set of context vectors $V_{u_t} = \{v_1, v_2, ..., v_j\}$ used to pull for user $u_t \in N$. V_{u_t} consists of the vectors of candidate arms connected to the users in the current round. Most applications depict the item as the arm that is suggested for user u_t.

Based on the estimated expectation acquired from the historical feedback $< u_t, v_t, r_t >$, the learning agent selects the best arm $v_t \in V_{u_t}$ for the user. After pulling the arm, the corresponding reward r_t is obtained, which comes from a reward function associated with both user and item. The agent updates the user parameter $\theta_{u,t}$ and the item parameter v_t respectively based on the reward. In standard linear contextual bandit problems, the reward is determined by a linear function consisting of θ_{u_t}, v_t and noise η_t:

$$r_t = \theta_{u_t}^T v_t + \eta_t \tag{1}$$

where η_t is a noise with zero-mean and σ-bounded variance, drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

The total reward obtained after T rounds is $\Sigma_{t=1}^T r$. The goal of standard Bandit is to minimize cumulative regret, and the objective function is defined as follows:

$$R_T = \Sigma_{t=1}^T r^* - \Sigma_{t=1}^T r = \Sigma_{t=1}^T \theta_{u_t}^T v_t^* - \Sigma_{t=1}^T \theta_{u_t}^T v_t \tag{2}$$

where r^* is the reward received at round t for selecting the optimal arm. It is impossible to observe the rewards of all arms, obtaining the optimal arm is difficult. Since minimizing the cumulative regret R_T is equivalent to maximizing the total reward $\Sigma_{t=1}^T r$, the optimization objective function of the bandit problem alters to maximize the following total reward:

$$G = \Sigma_{t=1}^T r = \Sigma_{t=1}^T \theta_{u_t}^{\mathrm{T}} v_t \tag{3}$$

To address the aforementioned problem and predict reward r accurately, contextual bandit algorithms have been proposed to balance the trade-off between exploration and exploitation in arm selection. The LinUCB algorithm is one of the main strategies. It estimates the expected reward for each arm as linear regression on user features $\theta_{u_t}^T$ and a vector of contexts v_t, where $\theta_{u_t} \in R^d$ is the regression coefficient to be learned:

$$\bar{v} = \arg\max_{v_t \in V_{u_t}} \theta_{u_t}^{\mathrm{T}} v_t \tag{4}$$

However, Eq.(4) solely maximizing predicted rewards without exploring other arms to determine which arm performs better will be detrimental to long-term recommendations. To overcome this issue, confidence intervals are taken into account. α controls whether the model prefers exploration or

exploitation. The arm that maximizes the upper confidence bound is chosen.

$$\bar{v} = \arg\max_{v_i \in V_{u_t}} (\theta_{u_t}^{\mathrm{T}} v_t + \alpha \sqrt{v_t^{\mathrm{T}} H_{u_t}^{-1} v_t}) \tag{5}$$

where the update weight, $H_{u_t}^{-1} \in R^{d \times d}$, is the covariance of the coefficient $\theta_{u_t}^{T}$, which contains the feature of user and arm chosen in the previous round. And $\theta_{u_t}^{T}$ is calculated by ridge regression and is estimated as follows:

$$\theta_{u_t}^{T} = H_{u_t}^{-1} b_{u_t} \tag{6}$$

where $b_{u_t} \in R^{d}$ be the corresponding response vector(e.g, the corresponding d click/no-click user feedback).

IV. The SWCB Model

In this section, we introduce the proposed SWCB model which utilizes the bandit framework to enable online recommendation and switches clustering algorithm to speed up the cold start process. The overall architecture of SWCB is shown in figure 2. It consists of the following three modules to implement efficient online recommendations: Initialization module, Recommend and Update module, and Switching and Clustering module.

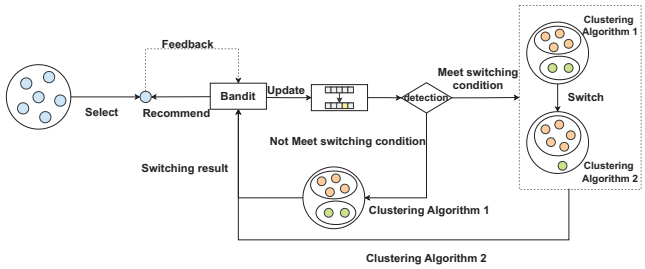


Fig. 1. The overall architecture of SWCB.

Algorithm 1 SWCB

```
Input: Exploration Hyperparameters β, γ.
Initialization
for stage g = 1, 2, ··· , 2^g do
    Set m_u = 0 for user u in each cluster.
    Set θ_k = H_k^{-1} b_k for each cluster k, k is the cluster index in K.
    for t = 1, 2, ··· , T do
        Recommend and Update
        if Signal = 0 then
            Execute Clustering Module 1
        end if
        if Signal = 1 then
            Execute Clustering Module 2
        end if
        Set m_u = 1.
    end for
    Execute Switching module to determine whether it is feasible to switch.
end for
```

Initialization module In the initialization stage, SWCB sets appropriate values for the correlation vectors b, θ, and matrices H of users and clusters respectively. The model also initializes the cluster-specific information (users in the cluster, index, and the total number of clusters) and the number of user interactions. Three parameters α, β, and γ need to be set, $alpha$ is obtained according to Eq.(7), and it is used to control the upper confidence bound when predicting rewards, β and γ are hyperparameters for merging and splitting clusters.

$$\alpha = \sqrt{d \times \ln \frac{1+N}{d} + 4 \times \ln t + \ln 2} + 1 \tag{7}$$

Moreover, we record the number of user interactions (note that the number of interactions in a cluster is set to be the sum of all users in that cluster) which is also one of the factors affecting the performance of clustering. The interact frequency of each user is set to 1 initially for computational feasibility. The initial cluster is set as the whole users with index 1.

Algorithm 2 Initialization

```
Setting parameters for each user:
H_{u_0} ← I ∈ R^{d×d}, b_{u_0} ← 0 ∈ R^d and F_u ← 1
Setting parameters for the cluster:
Index K ← {1}, H_k ← I ∈ R^{d×d}, b_k ← 0 ∈ R^d, C_k ← {N}, K(u) = 1 and F_k ← 1.
Set Signal = 0
Set X = 0
```

Recommend and Update Module SWCB performs the recommendation process on a series of stages, denoted by the index g. Each stage comprises T timestamps, representing a sequence of discrete time intervals. Within each stage, SWCB employs a selective exploration strategy, prioritizing the exploration of poor user clusters while keeping good clusters unaffected. At the start of each stage, every user u is initialized as unserved, with a binary variable m_{u_t} set to 0. If a user u is served at timestamp t, the corresponding variable m_{u_t} is updated to 1. Here, t denotes the current timestamp and subscript u_t represents the user selected for recommendation at that time. The context vectors $V_{u_t} = \{v_1, v_2, ..., v_j\}$ correspond to the set of available items for recommendation to user u_t at timestamp t. The cluster $c_k (u_t \in c_k)$ can be found and the algorithm computes the bandit parameters of user cluster c_k by $\theta_k = {H_k}^{-1} b_k$. Then SWCB recommends $\bar{v}$ for user u_t according to Eq.(5)

The information about the user and the cluster is updated once the recommendation for user u_t is completed and the user's feedback r_t is obtained. The correlation vectors b, θ, and matrices H of users and clusters are updated according to payoff r. The equation is as follows:

$$H_{u_t} = H_{u_t} + v_j v_j^{\mathrm{T}}, b_{u_t} = b_{u_t} + r_t v_j, \theta u_t = {H_{u_t}}^{-1} b_{u_t} \tag{8}$$

Switching and Clustering SWCB splits and merges clusters during the beginning of cold-start to construct the profile of user clusters efficiently (as shown in Algorithm 3). As the recommendation proceeding, most users within the same cluster will have similar preferences and only a few users have different interests. Then SWCB evaluates the quality of clusters according to the frequency of user interactions and change

Algorithm 3 Recommend and Update Module

Receive user $u_t \in N$
Get context vectors $V_{u_t} = \{v_1, \cdots, v_j\}$
Compute the coefficient of cluster k: $\theta_k = {H_k}^{-1} b_k$.
Recommend item $\bar{v} = \arg\max_{v_j \in V_{u_t}} ({\theta_k}^{\mathrm{T}} v_j + \alpha \sqrt{v_j^{\mathrm{T}} {H_{c_j}}^{-1} v_j})$.
Receive the feedback r_t given by user u_t.
Update:
$H_{u_t} = H_{u_t} + v_j v_j^{\mathrm{T}}, b_{u_t} = b_{u_t} + r_t v_j, F_{u_t} = F_{u_t} + 1$
$\theta u_t = {H_{u_t}}^{-1} b_{u_t}, P_{u_t} = \frac{F_{u_t}}{t}$
$H_k = H_k + v_j v_j^{\mathrm{T}}, b_k = b_k + r_t v_j, F_k = F_k + 1$
$\theta_k = {H_k}^{-1} b_k, P_k = \frac{F_{\hat{u}}}{t} \forall \hat{u} \in C_k$

Algorithm 4 Switching module

Compute $G(F) = \sqrt{\frac{1+\ln(1+F)}{1+F}}$
if $\|p_{u_t} - p_k\| < \gamma G(F_{u_t})$ **then**
  $X = X + 1$
  if $X = 10$ **then**
    Set $Signal = 1$
    Input current information of clusters:
    $H_k = I + \sum_{u \in C_k} (H_u - I)$
    $b_k = \sum_{u \in C_k} b_u$
    $\theta_k = {H_k}^{-1} b_k$
  end if
end if
if $\|p_{u_t} - p_k\| > \gamma G(F_{u_t})$ **then**
  $X = 0$
end if

Algorithm 5 Clustering module 1

Compute $G(F) = \sqrt{\frac{1+\ln(1+F)}{1+F}}$
Split:
if $\|\theta_{u_t} - \theta_k\| > \beta(G(F_{u_t}) + G(F_k))$ or $\|p_{u_t} - p_k\| > \gamma G(t)$ **then**
  $\hat{k} \leftarrow maxK + 1$
  $K(u_t) = \hat{k}$
  Split user u_t from k
  Remove user u_t to new cluster $C_{\hat{k}}$
  Update:
  $H_k = H_k - H_{u_t} + I, b_k = b_k - b_{u,t}, F_k = F_k - F_{u,t}$ $C_{\hat{k}} = C_{\hat{k}} - \{u_t\}$
  $H_{\hat{k}} = H_{u_t}, b_{\hat{k}} = b_{u_t} F_{\hat{k}} = F_{u_t}, C_{\hat{k}} = \{u_t\}$
end if
Merge:
if any two marked clusters k_1 and k_2 satisfying $\|\theta_{k_1} - \theta_{k_2}\| < \frac{\beta}{2}(G(F_{k_1}) + G(F_{k_2}))$ and $\|p_{k1} - p_{k2}\| < \gamma G(t)$ **then**
  Add information of cluster k_2 to cluster k_1
  Update:
  $c_{k_1} = c_{k_1} \cup c_{k_2}$
  $F_{c_{k_1}} = F_{c_{k_1}} + F_{c_{k_2}}$
  $H_{k_1} = H_{c_{k_1}} + H_{c_{k_2}} - I$
  $b_{k_1} = b_{c_{k_1}} + b_{c_{k_2}}$
  Delete c_{k_2}
end if

Algorithm 6 Clustering module 2

Re-assign the user to the closest cluster $C_{\hat{k}}$:
$\hat{k} = argmin_{\hat{k}=1,\cdots,K} \|\theta_{u_t} - \theta_{\hat{k}}\|$
If $\hat{k} \neq k$, move u_t from C_k to $C_{\hat{k}}$
Re-compute coefficients θ_k and $\theta_{\hat{k}}$:
$\theta_k = {H_k}^{-1} b_k$
$\theta_{\hat{k}} = {H_{\hat{k}}}^{-1} b_{\hat{k}}$

the update strategy. We adopt the metric $G(F) = \sqrt{\frac{1+\ln(1+F)}{1+F}}$ to evaluate the performance of the current clustering. If the relevant parameters of clustering meet the conditions specified by Eq.(9) within the designated rounds, it indicates the initial construction of user clusters, obviating the need for further merging operations among the clusters. Then, SWCB switches to the clustering module 2 for long-term rewards clustering.

$$\|p_{u,t} - p_k\| < \gamma G(t) \tag{9}$$

Since the basic component of user clusters has been roughly constructed by splitting users and merging clusters, we can directly obtain the number k of current clusters without setting it manually. In each round t, the clustering module 2 reassigns user u to the current cluster that is closest to its bandit parameter θ_u. And clustering module 2 provides better clustering results when the user has enough interactions with the items.

V. Experiments

A. Dataset description

The experiments on synthetic dataset aim to show the clustering quality of proposed method via a visual way. The synthetic dataset consists of 100 users and 1,000 items, both of them are categorized into ten clusters. Thus, the size of each user cluster is 10. Each user maintains a parameter θ, which is acquired during the interaction with items. The feature vectors of both users and items exist in a 6-dimensional feature space. Regarding article feature vectors, a varying number of 0's are randomly assigned to the first five dimensions as a seed vector of a group, while the values of other non-zero dimensions are generated from a Gaussian distribution. During the recommendation sequence, the size of the candidate item set is 25, and the served user u_t is generated uniformly at random from the 100 users.

The two real-world datasets Last.FM and MovieLens are employed to evaluate the recommendation performance of various algorithms. In this study, we resample these two datasets by removing invalid users and items. Table 1 presents a comprehensive description of each dataset after preprocessing.

TABLE I
Real-world dataset summary

Dataset	# Users	# Items	# Tags
MovieLens	162541	62423	97635
Last.FM	5458	5595	5389

B. Compared algorithms

we compare the performance of our proposed method with the state-of-the-art bandit algorithm. The baselines are listed as follows:

- **ϵ-greedy.** It is a commonly used strategy in reinforcement learning that balances exploration and exploitation. It works by choosing a random action with a probability of ϵ and choosing the action with the highest estimated value with a probability of $(1 - \epsilon)$.
- **Lin-One.** It is a single instance of the LinUCB. Lin-one customizes an identical Bandit for all users and all users share the parameters of that Bandit model.

- **Lin-Ind.** It creates an independent instance of LinUCB for each user. This means that a Bandit model is customized for each user, and each user belongs to a separate cluster. Compared to Lin-one, Lin-Ind allows for personalized recommendations.
- **DynUCB.** It divides the user community into multiple clusters and customizes bandits for each cluster. Such clusters are dynamic and users can switch from one cluster to another depending on their changing preferences.
- **SCLUB.** It uses graphs as a set representation of user clusters and updates the user clusters by splitting and merging operations to speed up the process of identifying the underlying clusters.
- **DCMAB.** It lets users within the same cluster collaborate in evaluating the expected benefits of arm.

C. Experimental setup

We set the number of rounds T to 10000 per episode and record the cumulative reward for each round. The corresponding exploration parameters α of baseline are set to $\{0.1, 0.3, 0.5, 0.7\}$. In our experiments, the cumulative rewards are averaged after five episodes as the performance of each compared method.

D. Experimental results

Synthetic dataset experiment We conducted a visual analysis of SWCB on a synthetic dataset. Figure 3, Figure 4, and Figure 5 respectively provide visualizations of the clustering representations obtained by SCLUB, DynUCB, and SWCB on a synthetic dataset. From the visual effects of the three models, it can be observed that SCLUB and DynUCB don't perform well in clustering some individual users who are far from the clusters they belong to, whereas SWCB exhibits a clear clustering effect, indicating that switching and clustering module can result in better clustering representations.

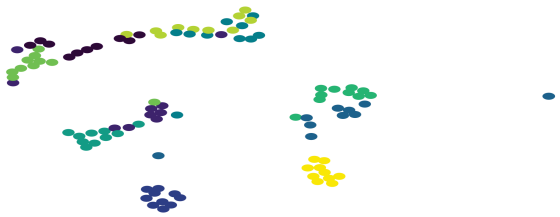


Fig. 2. Visualization of the cluster representation derived from SCLUB on synthetic dataset.

Real-world datasets experiment We compare and analyze the average cumulative reward of SWCB and other baselines during the initial recommendation stage, the experimental results are shown in Table 2, Table 3, Figure 6, and Figure 7 respectively. Overall, experimental results indicate that SWCB is more efficient in the early stages of recommendation (specifically, it achieves higher cumulative rewards in fewer recommendation rounds compared to other baselines).

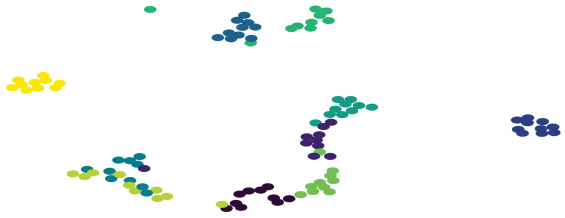


Fig. 3. Visualization of the cluster representation derived from DynUCB on synthetic dataset.

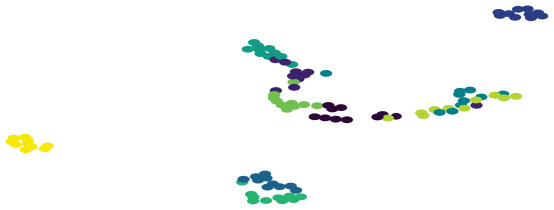


Fig. 4. Visualization of the cluster representation derived from SWCB on synthetic dataset..

From Table 2 and Table 3 we can find that on two real-world datasets, SWCB achieves the highest average cumulative rewards compared with other five methods. The average cumulative reward of SWCB on MovieLens is 6871 which is 338 higher than that of the best-performing baseline and on Last.FM is 2610 which is 358 higher than that of the best-performing baseline. These results show the effectiveness of switching clustering in the early stage of cold start for online recommendation.

Figure 6 and Figure 7 illustrate the trend of reward curves obtained from SWCB. It can be seen that before switching clustering the performance of SWCB is similar to those of the baselines. However, after switching, the efficiency of reward acquisition is improved. This result also reflects the effectiveness of our proposed model.

TABLE II
CUMULATIVE REWARDS ON MOVIELENS.

Algorithm	mean	max	min
Egreedy	5459±84.7	5595	5389
Lin-one	5329±175.2	5632	5192
Lin-ind	5444±31.5	5489	5392
SCLUB	6533±181.6	6586	5981
DynUCB	6495±97.3	6648	6339
DCMAB	6671±82.9	6781	6423
SWCB	**6871**±73.4	**6912**	**6631**

Ablation experiment We conduct an ablation analysis for SWCB on two real-world datasets to demonstrate the effectiveness of switching clustering algorithms during the recommendation process. In this experiment, SWCB-2 represents the SWCB with clustering module 2 removed. SWCB-1 represents the SWCB with clustering module 1 removed, and both of them only use one clustering algorithm without

TABLE III
CUMULATIVE REWARDS ON LAST.FM.

Algorithm	mean	max	min
Egreedy	616±31.7	654	576
Lin-one	222±26.1	243	178
Lin-ind	197±13.2	213	181
SCLUB	1620±172.9	2147	1591
DynUCB	2252±142.6	2560	2157
DCMAB	2352±113.6	2512	2219
SWCB	**2610**±67.2	**2617**	**2417**

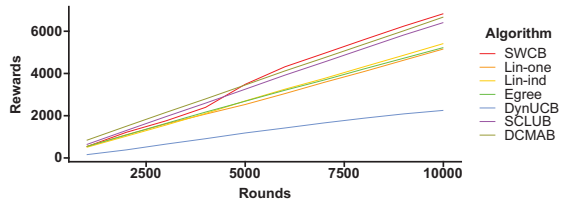


Fig. 5. Cumulative rewards of the proposed model and baselines on MovieLens.

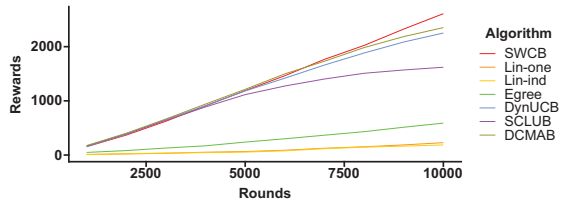
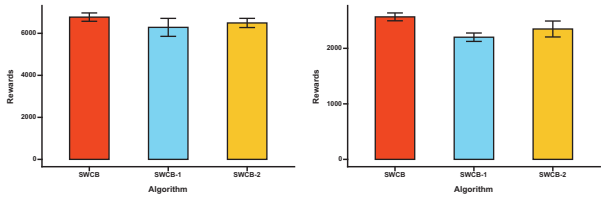


Fig. 6. Cumulative rewards of the proposed model and baselines on Last.FM.



(a) MovieLens (b) Last.FM

Fig. 7. Cumulative rewards for SWCB and SWCB components

switching during the recommendation process. The remaining experimental settings are consistent with Real-world datasets experiments. The experimental results are shown in Figure 8 and Figure 9.

From Figure 7, we can see that compared to the algorithms without clustering switching, SWCB achieves better performance, this indicates the effectiveness of our proposed clustering switching approach. Experimental results also show that the performance of SWCB-2 is better than that of SWCB-1. This may be because SWCB-1 is more suitable for scenarios where there is sufficient user interaction information, while SWCB-2 is the opposite. Therefore, in the case of a short recommendation round, using SWCB-2 throughout the process results in a higher average cumulative reward than using SWCB-1. This also demonstrates the effectiveness of our clustering switching design.

VI. CONCLUSION

In this paper, we propose an online recommendations model named SWCB, that is designed under the bandit framework and combines clustering algorithms to cope with the cold start problem. Unlike the traditional clustering-based bandits relying on one clustering algorithm, SWCB improves the efficiency of recommendations by switching the clustering algorithms. When the clustering metrics reach a certain threshold, SWCB alters the way of user clustering, thus speeding the information retrieval in the early stage and helping to retain new users of online recommendations. In the future, we consider making further improvements to the switching mechanism of the clustering algorithm in SWCB. This involves enhancing the switching process not only based on the quality of the current user clusters but also responding to changes in popular items on the item side.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grants No. 62171391). Shaorong Fang and Tianfu Wu from Information and Network Center of Xiamen University are acknowledged for the help with the GPU computing.

REFERENCES

[1] L. Chuanhao, W. Qingyun, and W. Hongning, "When and whom to collaborate with in a changing environment: A collaborative dynamic bandit solution," Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021: 1410-1419.
[2] K. Christakopoulou and A. Banerjee, "Learning to interact with users: A collaborative-bandit approach," Proceedings of the 2018 SIAM International Conference on Data Mining, 2018: 612-620.
[3] A. Barraza-Urbin, "The exploration-exploitation trade-off in interactive recommender systems," Proceedings of the eleventh acm conference on recommender systems, 2017: 431-435.
[4] S. Shams, D. Anderson and D. Leith, "Cluster-based bandits: Fast cold-start for recommender system new users," Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021: 1613-1616.
[5] P. Auer, N. Cesa-Bianchi and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," Machine learning, 2002, 47: 235-256.
[6] L. Bui, R. Johari and S. Mannor, "Clustered bandits," arXiv preprint arXiv:1206.4169, 2012.
[7] S. Li, C. Gentile and A. Karatzoglou, "Graph clustering bandits for recommendation," arXiv preprint arXiv:1605.00596, 2016.
[8] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," arXiv preprint arXiv:1402.6028, 2014.
[9] S. Li, A. Karatzoglou and C. Gentile, "Collaborative Filtering Bandits," Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016: 539-548.
[10] T. Nguyen, T. Trong and W. Hady, "Dynamic Clustering of Contextual Multi-Armed Bandits," Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 1959-1962.
[11] J. Sanz-Cruzado, P. Castells and E. López, "A simple multi-armed nearest-neighbor bandit for interactive recommendation," Proceedings of the 13th ACM conference on recommender systems. 2019: 358-362.
[12] C. Yan, H. Han and Y. Zhang, "Dynamic clustering based contextual combinatorial multi-armed bandit for online recommendation," Knowledge-Based Systems, 2022, 257: 109927.