Contextual bandits to increase user prediction accuracy in movie recommendation system

Yizhe Chen*

Faculty of Science, University of Hong Kong, 999077, Hong Kong, China

Abstract. Cold-start problems are inevitable phenomena where recommendation systems fail to accurately predict users' favour and cause the loss of new users. The typical Multi-Armed Bandit (MAB) models are widely adopted as recommendation systems to solve cold-start problems, but standard MAB takes much more recommendation trials than new user's tolerance. This study adopts Contextual Multi-Armed Bandit (CMAB) to alleviate such situations and compares the performance of CMAB and typical MAB models at an early stage of the cold phase. Overall, CMAB generated better results in 15 trials in terms of cumulative regret and discounted cumulative gain. The optimal number of groups is 10, which alleviates cold-start problems efficiently, and sustains the efficiency of the off-line recommendation system under collaborative filtering. This paper suggests a possible selection of CMAB for recommendation systems to alleviate the cold start problem and estimates the tuned parameters for the MovieLens dataset. The evaluation metric in this paper provides a possible method of analyzing the general performance of a hybrid recommendation system, instead of adopting multiple evaluation metrics respectively, these metrics also provide estimates of the optimal value of parameters.

1 Introduction

A movie recommendation system is a strategy to mitigate information overload. It faced the dilemma of exploration and exploitation: either recommending new movies to the user to explore user preference or recommending movies that are previously interacted with to ensure user satisfaction. This dilemma is a typical problem in Multi-Armed Bandit (MAB), first introduced by Robbins [1]. In MAB problems, decision-makers are presented with k arms (action) and must select one at each time step, for each selection, a stochastic result is observed from a fixed but unknown distribution. The decision maker would refer to the historical observation and make the next move accordingly. The MAB problem aims to construct a sequential decision strategy that balances the inherent value of exploration and exploitation to minimize the theoretical cost of not selecting the optimal arm.

In real scenarios of movie recommendation problems, the agent is provided with contextual information including the user's watching history and ratings, the performance of other users, and the large number of arms available for recommendation [2]. If movies are

* Corresponding author: chen1204@connect.hku.hk

treated as individual arms, or users are treated individually, this would significantly increase the computation complexity of the system when new users and new movies are introduced to the system [3]. To reduce complexity and increase efficiency, researchers have made much progress in modifying the movie recommendation system. The existing movie recommendation systems are either based on collaborative filtering, content filtering, or hybrid/mixed methods [4]. Content-based filtering is by extracting features from a single user behavior with pre-defined features, while collaborative filtering does not specify the name of the feature, rather, it is based on latent feature extraction from the overall item-user matrix that acts similarly as a weighted average. Among these strategies, all of them are effective methods of minimizing regret and enhancing the prediction accuracy of the system. However, they do not alleviate the cold start problem in real scenarios.

To alleviate the above issues, Contextual Multi-Armed Bandit (CMAB) has been introduced and adopted as a recommendation system. CMAB considers the trade-off between exploration and exploitation and meanwhile, this paper proposes comparative evaluations between existing recommendation models on various standards, and construct methodology that introduced CMAB to the recommendation system. We compare similarities and differences, point out the advantages of CMAB over typical MAB, and address the limitations of the model. This paper focuses on problems faced by traditional recommendation systems and evaluates the result of introducing the CMAB model to the system. This paper will also compare the performance of context-free MAB and CMAB in terms of prediction accuracy and complexity. For the remainder of this paper: Section 2 addresses related work mostly from 2020 to 2024, that is related to our study; Section 3 introduces the methodology of our recommendation model by first listing keywords and formulas, introducing the dataset, and comparing results; Section 4 and Section 5 analyze the comparison between different bandits and proposes some future work.

2 Related work

2.1 Cold start problem

The typical content-based filtering model and collaborative filtering cannot effectively alleviate the cost of the cold-start problem, where a new user or new movie is introduced to the system. Since there is no historical data that could refer to this user/item, therefore, typical recommendation systems fail to accurately predict the user's favour or identify the movie's features during the cold start phase [5]. Furthermore, the standard recommendation system underestimates the potential of such new items. In this case, the platform would suffer from the loss of new users, this problem is similar to the lack of knowledge to reward distribution in the MAB problem, which makes the bandit a suitable method to improve prediction accuracy during the cold start period. However, the typical MAB model considers users individually and treats each movie as an individual arm, without considering the contextual information (user-item matrix, side-information of the user), therefore, the use of Contextual Multi-Armed Bandit after modification would have advantages of both collaborative filtering and MAB to solve cold start problem [6]. Related studies adopt Linear Upper Confidence Bound (LinUCB) as a recommendation system, [7] formulate a hybrid model, and formulate banditMF which combines MAB and Matrix Factorization to improve the overall performance of the system [8].

2.2 Contextual multi-armed bandit

The standard MAB model considers a fixed and unknown reward distribution, at the

beginning of the experiment, there is no additional information about the environment, which enhances the value of exploration at the beginning [2]. However, for movie recommendation systems, platforms have existing information, including the record of other users and side information of new users including age, gender, and occupation. Unlike MAB, Contextual multi-armed bandits could make use of such information to reduce the need for exploration, such strategies could increase the overall performance of the system [5, 6]. A widely used CMAB algorithm is LinUCB, as shown in Algorithm 1. For each user entering the system, the experiment first extracts a feature vector of dimension 79 from the 'User' and 'Movie' sets from MovieLens, which includes the user's features including age, gender, user occupation, and movies' features including genres, tags, and average ratings. For each arm, initialize the identity matrix with dimension 79, with a corresponding zero vector b. Compute the vector θ_a and value $p_{t,a}$ accordingly. For each time step t, the system selects arm i (clusters of users-movie matrix) with the largest p value and observes the reward r_t and update context matrix A_i, b_i. Similar to the standard Upper Confidence Bound (UCB) algorithm, LinUCB is based on deterministic strategies, where the next arm is determined by the value of p, while Epsilon-greedy and Thompson Sampling (TS) are based on stochastic arm selections.

Algorithm 1 LinUCB with disjoint linear models.

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ ***do***
2: Observe features of all arms $\alpha \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4: **if** a is new **then**
5: $\mathbf{A_a} \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
6: $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-dimensional zero vector)
7: **end if**
8: $\widehat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
9: $p_{t,a} \leftarrow \widehat{\boldsymbol{\theta}}_a^{\top}\mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^{\top}\mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$
10: **end for**
11: Choose arm $a_t = \text{argmax}_{a\in\mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
12: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^{\top}$
13: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$
14: **end for**

3 Methodology

3.1 Dataset

This paper adopts the MovieLens 1M dataset, which contains 6040 users and 3952 movies. Each combination of user and movie is tagged with a whole 5-star rating. Each user rated at least 20 movies. The age, gender, and occupation of the user are available and are further vectorized to form a feature vector in the LinUCB algorithm. This study mainly focuses on solving cold start problems with the use of contextual bandits, therefore, 90% of the users are assigned to a training model to become contextual information and constructed into a user-item matrix, and the rest of them are used to represent cold users that the recommendation system has completely no information.

3.2 Algorithm

Data used for training is first constructed into the user-movie matrix, where rows denote UserID and columns represent MovieID. This user-movie matrix contains many more missing values than the actual ratings. After Funk Singular Value Decomposition matrix factorization, the user-movie matrix is separated into user-feature matrix (m * k) and movie-feature matrix (k * n) to produce the matrix of predicted ratings (m * n), this alleviates the problem of data sparsity and missing values if the value of k is tuned [5]. Then the predicted matrix undergoes user-clustering which transforms similar users into k groups. This clustered matrix (k * n) is the base context of the recommendation system, the paper focuses on improving the use of this matrix to solve cold-start problems.

The movie recommendation system is separated into two parts based on whether the user is cold: online recommendation and offline recommendation. Figure 1 is a flow chart of BanditMF, which combines collaborative filtering with clustering as off-line recommendation adopts MAB for online recommendations. When a new user enters the system, MAB randomly picks one of the user clusters and recommends movies that satisfy the previous users. This step is achieved by ranking the movie's weighted average rating and then selecting the top-rated movie that the target user has not watched. The system then receives this new user's reward, updates the user profile, and checks if the user is still cold: if so, the system remains in online recommendation; if not, the user is transferred to the offline recommendation, where she is appended to one of the user clusters and identified as a non-cold user, subsequently, system updates the clustering matrix, the future recommendation for this user would be based on off-line collaborative filtering.

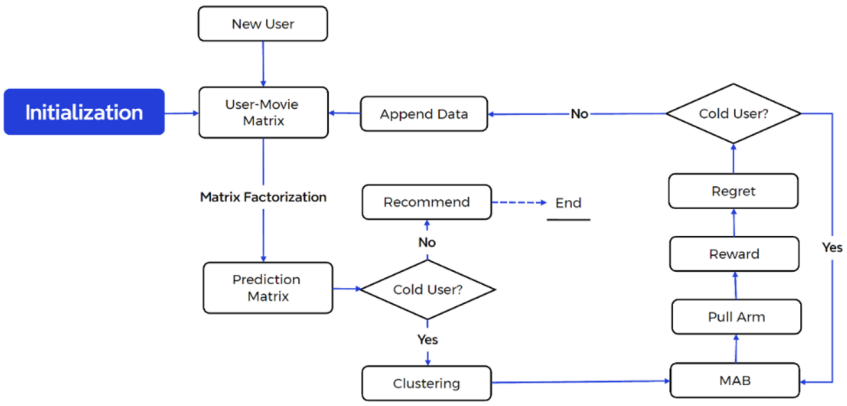

Fig. 1. Flow Chart of BanditMF (Photo/Picture credit: Original)

In the online recommendation process, UCB, Epsilon-greedy, Thompson Sampling, and LinUCB are selected with tuned parameters [8-11]. Another popular algorithm, Explore-Then-Commit is not selected because its arm selection is trivial and occurs only once during the experiment, while the cold-start problem requires frequent changes in arm selection to estimate and predict users' preferences. The rest of the algorithms provide distinct methods for balancing the trade-off between exploration and exploitation, LinUCB is context-based and the rest of the models are context-free. The comparison among these models not only provides insight into the value of CMAB to recommendation systems but also measures the performance differences.

3.3 Evaluation metrics

Rewards of an arm are based on ratings from 1 to 5, and further standardized into the interval [0, 1] from Formula (1). This paper compares the use of different bandits in the online recommendation process, since we focus on alleviating the cold-start problem, where the user's rating number is not large, we mainly focus on the first 5-10 recommendations and measure rewards and regrets [6].

$$r_{u,i} = \frac{Observed\ Rating}{Maximum\ Rating} \tag{1}$$

To emphasize the importance of correct predictions in the early stage of the algorithm, this paper also adopts two more evaluation metrics: discounted cumulative gain (DCG) and normalized discounted cumulative gain (NDCG). From (2) and (3) Ideal Discounted Cumulative Gain (IDCG) is the optimal permutation of rewards that maximizes the theoretical DCG values, the fraction of DCG and IDCG values outputs a normalized value, allowing comparison between algorithms [12].

$$DCG(u) = r_{u,1} + \sum_{t=2}^{T} \frac{r_{u,t}}{log_2(t)} \tag{2}$$

$$NDCG(u) = \frac{DCG(u)}{IDCG(u)} \tag{3}$$

From Formula (2), the reward of the later prediction is discounted to increase the weight of the first several predictions. Equation (3) normalizes DCG into interval [0, 1]. These metrics provide better observation of MAB performance on cold-start problems.

3.4 K-means clustering

K is set to 3 initially, indicating that users are clustered into 3 groups. This study adopts Euclidean distance to measure the clustering procedure [13]. The raw predicted rating matrix undergoes 20 iterations of clustering with a randomly selected initial centroid, then repeats this process on 10 different centroids and picks the cluster with the lowest Root Mean Squared Error. Figure 2 indicates the Root Mean Squared Error (RMSE) decline after each iteration.

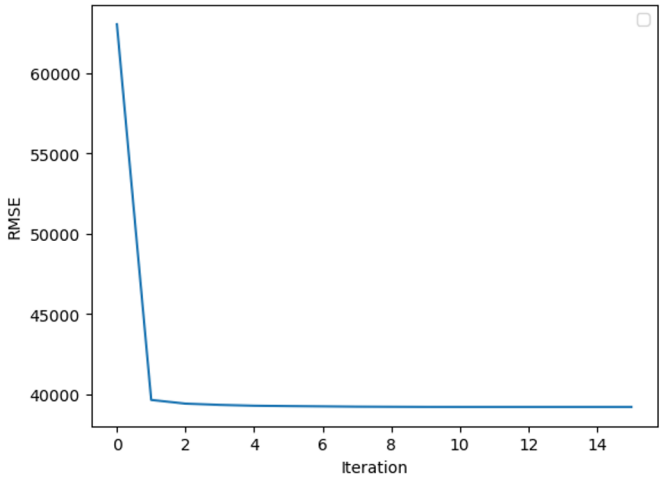

Fig. 2. Decline of RMSE after 15 Iterations (Photo/Picture credit: Original)

4 Results & analysis

This study tests the performance of different bandits with distinct recommendation periods (T). The tables below show the average NDCG and Cumulative Regrets of 50 cold users, which represent the performance of bandits on general users. From Table1, 2, 3, with T = 15, algorithms have similar cumulative regrets, which is intuitively normal, since the logarithmic curve of cumulative regrets would only occur at a significantly large T. In general experiments, LinUCB has lower cumulative regrets and rather smaller standard deviations. Among algorithms, there is no significant difference in NDCG value, LinUCB has low standard deviations of NDCG, indicating a rather stable performance of achieving higher reward at the beginning of the experiment. This could be attributed to the effect of pre-defined contexts that provide extra information for LinUCB.

The comparison across tables measures the effect of selecting a different number of clusters (k). The general average cumulative regrets achieve the minimum at k = 10, whereas LinUCB also achieves the lowest average cumulative regrets. This measure reflects that user clustering groups k = 10 is a possible value that optimizes the online recommendations.

Table 1. NDCG & Cumulative Regrets ($T = 15$, $N = 50$, $k = 3$)

	NDCG	**std**	**Cumulative Regret**	**std**
UCB	0.979976248	±0.002081295	3.66202423	±1.163195
TS	0.983072127	±0.002788222	3.5524631	±1.137956
LinUCB	0.977403072	±0.001840605	3.37830091	±1.092272
ϵ-greedy	0.983487047	±0.002537924	3.66350369	±1.155109

The experiment fixed Period T to be 15 and estimated the performance of 50 cold users. From Table 1, LinUCB has lowest cumulative regret = 3.3783 and lower standard deviation

than other algorithms, which reflects the advantages of including contextual information to the algorithm. This indicates an advantage of adopting LinUCB at small value of k (user clusters).

Table 2. NDCG & Cumulative Regrets (T = 15, N = 50, k = 5)

	NDCG	**std**	**Cumulative Regret**	**std**
UCB	0.960623884	±0.00326680	3.73231872	±0.975485833
TS	0.957844304	±0.00342019	3.81466838	±1.04241418
LinUCB	0.958229495	±0.00222586	3.56436444	±0.999912674
ϵ-greedy	0.958622290	±0.00338849	3.68489675	±1.01011221

Table 3. NDCG & Cumulative Regrets (T = 15, N = 50, k = 10)

	NDCG	**std**	**Cumulative Regret**	**std**
UCB	0.984250784	±0.00280675	3.50778381	±1.09590408
TS	0.97747411	±0.0036339	3.50726015	±1.07879191
LinUCB	0.97619576	±0.00349322	3.23152054	±1.08066565
ϵ-greedy	0.97721851	±0.0036943	3.50118035	±1.15437115

The above tables illustrate the change of cumulative regrets and NDCG value on different selections of k, and period T, however, these evaluation metrics measure the prediction accuracy during the online recommendation period based on the MAB model, instead of collaborative filtering in the off-line part. To provide a comprehensive measurement of the general performance of the system, this paper analyzes the selection of k to off-line recommendation. The equation (4) below measures the performance of offline recommendations for non-cold users. In the context of collaborative filtering, the system extracts K movies that have the top rating to the users. The precision at K measures the ratio of the successful recommendations to the total number of recommendations.

$$Precision\ at\ K = \frac{Successful\ Recommendations}{Total\ Recommendations\ (K)} \tag{4}$$

When a non-cold user enters the system, given that the system contains enough historical data on that user, it is assigned to off-line recommendations automatically according to Figure 1. Table 4 below measures the average precision at K of 50 non-cold users, given the number of movies watched per user is not large, the precision value is generally below 0.2. The missing value in the user-movie matrix also indicates such a phenomenon. However, this paper studies the selection of k, which is the number of clusters when formulating the

prediction matrix. The parameter K here indicates the length of the recommendation list and is fixed at 15.

Table 4. Average Precision at *K*

***k* (*K* = 15, *N* = 50)**	**Precision**	**± Standard Deviation**
k = 3	0.1466666666	±0.1813529401
k = 5	0.152000000	±0.157220298
k = 10	0.169333333	±0.13997460
k = 20	0.148000000	±0.173481987

The result from Table 4 indicates that selecting cluster number at k = 10 is a possible optimal choice, which maximizes the average precision for general users and also has the minimum standard deviation. The study does not include a selection of other parameters such as K, since the number of movies per recommendation is usually pre-defined and limited to multiple variables including the webpage size and size of movie posters.

5 Limitations & outlook

This study mainly focuses on online recommendations particularly and builds comparisons on different bandits (context-free & contextual), such evaluation provides insight into MAB model selection for movie recommendation systems, however, this paper does not standardize and evaluate the definition of "cold-users". From Figure 2, this paper does not provide evaluation metrics on whether users are still cold-users and suitable to be appended to the user-item matrix. Without such evaluations, the algorithm may overestimate the period *T* required to gather enough information for cold users, the actual optimal *T* may be much less than 15 or even 10 movies. Furthermore, the results of NDCG do not show significant differences between CMAB and typical MAB models. This might be attributed to the untuned *K* value in clustering, the optimal K could not be determined intuitively but rather through further experiments. Furthermore, this study lists the age, gender, and occupation of the new user as contextual information related to their movies' favors, however, there is no guarantee that these factors are directly related to users' behaviors of movie rating.

The future work is to provide standardized evaluation methods of users that could determine the "coldness", such metric could avoid redundant computations of MAB and reduce the computation complexity of the system. Furthermore, to simulate the overall performance of recommendation systems instead of the online recommendation part, future works could combine offline and online recommendations under a universal evaluation. For optimizing the selection of cluster group numbers, K, one might adopt a dynamic clustering method to achieve better observations.

6 Conclusion

This paper evaluates the impact of using Contextual Multi-Armed Bandits on the online recommendation part in the movie recommendation system, the research focus is on alleviating the cold-start problem to reduce new-user loss in real scenarios. Under NDCG and cumulative regrets, at low experiment periods, $T = 10$, epsilon-greedy shows better

performance, However, after 5 iterations, the LinUCB algorithm consistently outperforms epsilon-greedy. This shift in performance can be attributed to the fact that LinUCB leverages contextual information, enabling more informed arm selection, particularly in the early stages of the experiment. This study assures that LinUCB has advantages over typical MAB algorithms for solving cold-start problems, with tuned parameters, LinUCB might be a more efficient model for alleviating user loss. The experiment also measures the performance of both off-line and online recommendation system on different values of user clusters, and the performance achieves optimality at $k = 10$, which balances the efficiency on solving cold-start problems and sustains the efficiency of off-line recommendations. The evaluation metrics adopted by this paper could be a possible method of estimating the general performance of the recommendation system.

References

1. H. Robbins, Some aspects of the sequential design of experiments. Bull. Amer. Math. Soc. **58**, 527-535 (1952). https://www.ams.org/journals/bull/1952-58-05/S0002-9904-1952-09620-8
2. C.A. Gomez-Uribe, N. Hunt, The Netflix recommender system. ACM Trans. Manag. Inf. Syst. **6**, 1-19 (2015). https://doi.org/10.1145/2843948
3. D.J. Foster, A. Krishnamurthy, H. Luo, Model Selection for Contextual Bandits, in the Proceedings of Machine Learning Research vol **125**:1–5, (2020). https://doi.org/10.48550/arXiv.1906.00531
4. Q. Kang, W.P. Tay, R. She, Multi-armed linear bandits with latent biases. Inf. Sci. **660**, 120103 (2024). https://doi.org/10.1016/j.ins.2024.120103
5. Li, W. Chu, J. Langford, A contextual-bandit approach to personalized news article recommendation, in the Proceedings of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, April 26-30 (2010). https://doi.org/10.48550/arXiv.1003.0146
6. T. Cunha, A. Marchini, A hybrid meta-learning and multi-armed bandit approach for context-specific multi-objective recommendation optimization, *arXiv preprint*, (2024). https://doi.org/10.48550/arXiv.2409.08752
7. S. Xu, BanditMF: Multi-armed bandit based matrix factorization recommender system, *arXiv preprint*, (2022). https://doi.org/10.48550/arXiv.2106.10898
8. R. Cañamares, M. Redondo, P. Castells, Multi-armed recommender system bandit ensembles, in Proceedings of the the 13th ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, United States, September 16-20 (2019). https://doi.org/10.1145/3298689.3346984
9. A. Dzhoha, I. Rozora, Multi-armed bandit problem with online clustering as side information. J. Comput. Appl. Math. **427**, 115132 (2023). https://doi.org/10.1016/j.cam.2023.115132
10. H.B. Xie, H. Gu, Z. Qi, Efficient algorithms for multi-armed bandits with additional feedbacks: Modeling and algorithms. Inf. Sci. **633**, 453-468 (2023). https://doi.org/10.1016/j.ins.2023.03.060
11. Dynamic clustering based contextual combinatorial multi-armed bandit for online recommendation. Knowl.-Based Syst. **257**, 109927 (2022). https://doi.org/10.1016/j.knosys.2022.109927

12. A.M. Ikotun, A.E. Ezugwu, L. Abualigah, et al., K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Inf. Sci. **622**, 178-210 (2023). https://doi.org/10.1016/j.ins.2022.11.139
13. F.M. Harper, J.A. Konstan, The MovieLens Datasets: History and Context. ACM Trans. Int. Int. Sys. **5**, 1-19, (2015). https://doi.org/10.1145/2827872