Multi-Armed Bandits to Recommend for Cold-Start User

Crícia Z. Felício[1,2], Klérisson V. R. Paixão[2], Celia A. Z. Barcelos[2], Philippe Preux[3]

[1] Federal Institute of Triângulo Mineiro, IFTM, Brazil
[2] Federal University of Uberlândia, UFU, Brazil
cricia@iftm.edu.br, {klerisson, celiazb}@ufu.br
[3] University of Lille & CRIStAL, France
philippe.preux@inria.fr

Abstract. To deal with cold-start user, recommender systems often rely on prediction models that exploit various sources of information. Such models are valuable to compensate the lack of ratings, but the abundance of them opens an important problem of model selection. So far, several methods have been proposed to deal with cold-start problem in recommender systems. However, facing a set of models, very little work exists on selecting the model to cope with a given cold-start user. To address this gap, this work in progress quantitatively investigates the implementation of multi-armed bandits for model selection during the cold-start phase. We present an encouraging preliminary experiment.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: cold-start problem, multi-armed bandits, recommender systems

1. INTRODUCTION

In a recommendation system, different models are commonly used to deal with different stages of a user experience. For example, a particular model works better in earlier stages when the recommender system does not know the user's tastes yet. However, in later stages, a different model should be more effective, and therefore one switches to the more effective model. Originally, switching methods [Burke 2002] were designed to handle cold-start problem. The idea is to switch from one model to an other once the system has enough data about the user, so he is not cold anymore.

While the concept of switching models [Billsus and Pazzani 2000] is not new for recommender systems (henceforth RS), the availability of several cold-start methods provides enriched resources to *model selection*. Applied to cold-start stage, a model selection method may be seen as a framework to alternate among prediction model in order to find a more suitable one. Few works have sought to empirically assess the efficacy of a model selection specifically within the cold-start stage. Based on this gap, the aim of this work in progress is to explore how a model selection can be useful to provide better recommendations. Our hypothesis is that recommendation model fails in part of its predictions, therefore a model selection that maximizes the recommendation gain might be more precise.

In this paper, we pose one research question and report preliminary results to identify the role of a feedback-oriented method for model selection. In particular, we investigate whether *bandit algorithms* are useful for this model selection task. These algorithms weigh models, so that the worst performing models end up with a very little weight. Therefore, the overall recommendation takes advantage of different models and might be better by selecting the best to use to make a particular recommendation, based on their past performance.

C. Z. Felício would like to thank the Federal Institute of Triângulo Mineiro for study leave granted. We also thank the Brazilian research agencies CAPES, CNPq and FAPEMIG for supporting this work.

2. BACKGROUND AND LITERATURE REVIEW

To understand our approach, called MAB-Rec, we introduce some RS formalism:

Let U be a set of users and I be a set of items. Each user $u \in U$ and each item $i \in I$ has a unique identifier. The user-item rating matrix is $R = [r_{u,i}]_{m \times n}$, where each entry $r_{u,i}$ is the rating given by user u on item i, and m is the number of users, and n is the number of items. The recommendation task is based on the predictions of the missing values of the user-item rating matrix. Then, prediction models are used to recommend those top-k ranked.

Multi-Armed Bandit (MAB) problem can be understood as a sequential decision problem where an algorithm continually chooses among a set of arms (in this paper, we assume the set of arms is finite). In each step t, an arm a is selected and pulled which leads to a reward $X_a(t)$. This reward is distributed according to a certain unknown law. Here, we consider that the goal is to learn, as fast as possible, through repeated arm pulls, the arm that returns the maximum expected reward.

In this work, we assume a set of prediction models as the arms from MAB problem. Then, our bandit algorithm is sequentially applied to choose a prediction model either the best performing one at the moment (exploitation), or an other arm to learn how it performs (exploration). We rely on the ϵ-Greedy algorithm to implement our model selection. ϵ-Greedy maintains the mean reward of each arm (prediction model) a, denoted by $\bar{X}_a$. Each time t that the arm a is played, the mean reward $\bar{X}_a$ is updated. The mean reward of the arm a at time t is represented by $\bar{X}_a(t)$.

We represent the probability of selecting arm a at time t as $\mathbb{P}_a(t)$. In each round t the ϵ-Greedy algorithm selects the arm with the highest mean reward with probability 1 - ϵ, and selects an arm uniformly at random with probability ϵ.

Related Work. This paper follows the line on applications of bandits in RS [Mary et al. 2015].

Li et al. [2010] reports on personalized recommendation of news articles as a contextual bandit problem. They propose LinUCB, an extension of the UCB algorithm. It selects the news based on mean and standard deviation. It also has a factor α to control the exploration / exploitation trade-off. Moreover, Caron and Bhagat [2013] incorporate social components into bandit algorithms to tackle the cold-start problem. They designed an improved bandit strategy to model the user's preference using multi-armed bandits. Several works model the recommendation problem using a MAB setting in which the items to be recommended are the arms [Bouneffouf et al. 2012; Girgin et al. 2012].In a different way, Lacerda et al. [2013; 2015] model users as arms to recommend daily-deals. They consider strategies for splitting users into exploration and exploitation.

In comparison, the goal of MAB-Rec is the selection of existent prediction models that might offer better recommendations for cold-start users. Our MAB setting is also different, whereas the arms are the prediction models.

3. MAB-REC APPROACH

MAB-Rec is made of 3 phases: (i) computation of the prediction models, (ii) sort prediction models, and (iii) recommendation. To define the set of prediction models, we applied three steps: Rating prediction, Preference clustering, and Consensus computation as described in [Felício et al. 2016]. We obtain $M = \{M_0 = (C_1, \hat{\theta}_1), \ldots, M_K = (C_K, \hat{\theta}_K)\}$, the set of prediction models where each M_s is composed of a cluster of users C_s and its consensual preference vector $\hat{\theta}_s$.

Table I shows an example of how a prediction model is built. In Table Ia we have a user-item rating matrix example with 2 users and 7 items. With BiasedMF algorithm[1] [Koren 2008] we obtain the predicted rating matrix, see Table Ib. Clustering the predicted rating matrix rows and considering that the two users is in the same cluster, we present the consensual preference vector $\hat{\theta}_1$ in Table Ic.

[1]The name BiasedMF comes from the LibRec library that we use in the experiments.

Table I. (a) Example of a user-item rating matrix. "-" means that the user has not rate the item. (b) Predicted rating matrix. (c) Consensual preference vector.

(a)

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	5	2	4	-	5	1	-
u_2	4	-	5	-	5	-	1

(b)

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4.6	2.09	4.23	4.24	4.84	1.07	1.0
u_2	4.2	3.8	4.42	5.0	4.86	2.28	1.2

(c)

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4.6	2.09	4.23	4.24	4.84	1.07	1.0
u_2	4.2	3.8	4.42	5.0	4.86	2.28	1.2
$\hat{\theta}_1$	4.4	2.94	4.32	4.62	4.85	1.67	1.1

After obtaining the prediction models, we sort the consensual preference vectors according to their ratings. So, for each $\hat{\theta}_s$ we have a $\hat{\theta}'_s$ that represents the consensual preference vector in a sorted order. The idea is to recommend the items with high ratings in each model first. We hypothesize that this strategy can contribute to learn users preference faster. For instance, the correspondent $\hat{\theta}'_1$ to $\hat{\theta}_1$ in Table Ic will have the sorted list of items equal to $\{i_5, i_4, i_1, i_3, i_2, i_6, i_7\}$.

Making Recommendations: at each time t a recommendation for a user u is made according to a bandit algorithm $\mathcal{B}$ following these steps:

(1) Select a prediction model M_s using the bandit algorithm $\mathcal{B}$;
(2) Select the next item i not recommended yet from $\hat{\theta}'_s$ (consensual preference vector of M_s sorted by ratings);
(3) Recommend item i to user u;
(4) Receive a rating $r_{u,i}$ as feedback from u;
(5) Return the reward;
(6) Update the prediction model statistics in $\mathcal{B}$.

We consider a binary reward where $X_{M_s} = 1$ if $r_{u,i} \geq r_{max} - \beta$, $\beta \in [1, 2]$, otherwise $X_{M_s} = 0$; r_{max} is the max rating in the dataset. Then, the reward is based on the proximity of user rating and r_{max}.

4. PRELIMINARY FINDINGS

4.1 Research Method

This section describes our methodology by outlining our research question, our dataset, and our analysis method. We structure our work around the following research question:

RQ: How effective are Multi-Armed Bandits to select initial recommendations for cold-start users?

To answer the above question, we mimic a cold-start scenario. This is done using the standard *leave-one-out cross-validation*, where the number of folds is equal to the number of instances in the dataset. Therefore, the selected prediction model is applied once for each instance, using all other instances as a training set, the remaining one being used as a single-user test set; then the performance are averaged over all users, each being used as a test set. Note that, to simulate a realistic cold-start scenario, we do not provide for train any preference, for instance, movie ratings from the test users and that is why we called this protocol **0-rating**.

The experiments were performed on a real-world dataset collected from Facebook users [Felício et al. 2015]. The dataset has 49,729 ratings from 498 users over 169 movies and with 40.9% of sparsity. Here, we took users that rated at least 20 items, because we want to evaluate MAB-Rec against *nDCG* metric for the firsts 20 items recommended.

We extended the LibRec [Guo et al.] implementation of BiasedMF to build the prediction models and incorporate the bandit algorithm in recommendation process. Experiments were executed with 10 latent factors and 100 iterations. The optimal number of consensual prediction models is 3.

4.2 Results

Table II presents the *nDCG* at rank size of 5, 10, 15, and 20 using ϵ-Greedy as the bandit algorithm. MAB-Rec achieves until 0.8620 for ndcg@5 with $\beta = 2$ and 0.8592 with $\beta = 1$. The difference is

quite small between the results using different ϵ values and between the two ways to compute reward. The explanation for this can be the dataset features where we have few items (169 movies) and a small number of prediction models (optimal values was got for 3 consensual prediction models, 3 clusters). Future work will investigate the MAB-Rec approach in others dataset and with others bandits algorithms.

Table II. nDCG results with ϵ-Greedy: (a) Binary Reward with $\beta = 1$; (b) Binary Reward with $\beta = 2$

(a)

ϵ	Rank size			
	@5	@10	@15	@20
0.1	0.8564	0.8474	0.8463	0.8449
0.2	**0.8592**	0.8495	0.8484	0.8479
0.3	0.8555	**0.8506**	0.8477	**0.8480**
0.4	0.8590	0.8504	0.8483	0.8479
0.5	0.8585	0.8504	**0.8495**	0.8476

(b)

ϵ	Rank size			
	@5	@10	@15	@20
0.1	0.8565	0.8481	0.8472	0.8468
0.2	0.8563	0.8481	0.8470	0.8467
0.3	0.8591	0.8496	0.8475	0.8473
0.4	0.8592	0.8501	0.8481	**0.8480**
0.5	**0.8620**	**0.8514**	**0.8490**	0.8476

5. FINAL REMARKS

We presented preliminary results on recommendation model selection through multi-armed bandit algorithm. Our proposed approach focus on the important problem of recommending for cold-start users. While prior works on model selection mainly aim at discovering when a user is not cold anymore, then switch to a new model, we are investigating new ways to foster RS when they have few or none information about the user. Our preliminary experimental results reached 86% of accuracy levels in terms of nDCG@5. We plan to look at different datasets, others bandits algorithms and different strategies to filter the set of prediction models.

REFERENCES

Billsus, D. and Pazzani, M. J. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction* 10 (2): 147–180, 2000.

Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. A contextual-bandit algorithm for mobile context-aware recommender system. In *Proc. IntĊonfón Neural Information Processing*. ICONIP. pp. 324–331, 2012.

Burke, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12 (4): 331–370, 2002.

Caron, S. and Bhagat, S. Mixing bandits: A recipe for improved cold-start recommendations in a social network. In *Proc. Workshop on Social Network Mining and Analysis*. SNAKDD. ACM, pp. 11:1–11:9, 2013.

Felício, C. Z., Paixão, K. V. R., Alves, G., and de Amo, S. Social prefrec framework: leveraging recommender systems based on social information. In *Proc. Symposium on Knowledge Discovery, Mining and Learning*. KDMiLe. pp. 66–73, 2015.

Felício, C. Z., Paixão, K. V. R., Barcelos, C. A. Z., and Preux, P. Preference-like score to cope with cold-start user in recommender systems. In *Proc. IEEE Int. Conf. on. Tools with Artificial Intelligence*. ICTAI, 2016.

Girgin, S., Mary, J., Preux, P., and Nicol, O. Managing advertising campaigns – an approximate planning approach. *Frontiers in Computer Science* 6 (2): 209–229, Apr., 2012.

Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. Librec: A java library for recommender systems. In *Proc. User Modeling, Adaptation, and Personalization*. UMAP.

Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*. ACM SIGKDD. Las Vegas, Nevada, USA, pp. 426–434, 2008.

Lacerda, A., Santos, R. L. T., Veloso, A., and Ziviani, N. Improving daily deals recommendation using explore-then-exploit strategies. *Information Retrieval Journal* 18 (2): 95–122, 2015.

Lacerda, A., Veloso, A., and Ziviani, N. Exploratory and interactive daily deals recommendation. In *Proc. ACM Conference on Recommender Systems*. RecSys. ACM, New York, NY, USA, pp. 439–442, 2013.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the International World Wide Web Conferences*. ACM, pp. 661–670, 2010.

Mary, J., Gaudel, R., and Preux, P. Bandits and recommender systems. In *Machine Learning, Optimization, and Big Data*. Lecture Notes in Computer Science, vol. 9432. Springer International Publishing, pp. 325–336, 2015.