Exploratory Recommender Systems Based on Reinforcement Learning for Finding Research Topic

Li Yu
School of Information
Renmin University of China
Beijing, China
buaayuli@ruc.edu.cn

Zhuangzhuang Wang
School of Information
Renmin University of China
Beijing, China
w1194690657@ruc.edu.cn

Hongrun Xin
International School
Beijing University of Posts and Telecommunications
Beijing, China
xinhongrun@bupt.edu.cn

Wei Zhang
School of Information
Central University of Finance and Economics
Beijing, China
kddzw@163.com

Xuefeng Li
School of Information
Central University of Finance and Economics
Beijing, China
xuefeng@cufe.edu.cn

Haiyan Wang
School of Information
Beijing Forestry University
Beijing, China
haiyan@bfu.edu.cn

***Abstract*—Traditional recommender systems try to select few items from some candidate items to users. Unfortunately, a user often hope recommender system help him to make a decision or finish a task based on his uncertain preference. For example, a researcher could hope recommender system to help him to find an advanced research topic by recommending literatures paper and refining his research interest and. In this paper, we develop an exploratory paper recommender system based on reinforcement learning, which can navigate a researcher to identify research topic by recommending papers continuously. In order to refine and focus user's research preference, as a reinforcement learning method, Multi-Armed Bandit (MAB) is employed for navigating recommendation paper. And two improved MAB methods are proposed, including ε-Greedy Stochastic Perturbation (ε-Greedy-SP) and Continuous Upper Confidence Bound (Con-UCB). Also, a weighted-LDA method is proposed for constructing the topic tree. A prototype system is developed and used to make experiments. Empirical research is made to analyze the change process of users' preference. The results show that the system is very effective for focusing and finding research topic.**

Keywords- Recommender Systems; Multi-Armed Bandit; Reinforcement Learning; Research Topic

I. INTRODUCTION

Along with the development of Internet and the explosive growth of information, recommender system has become a hot research issue in the past ten years. However, the goal of most recommender systems is just to recommend some to user from a lot of candidate items. A few researches are made on recommender system oriented to task. For example, recommender systems help a user to decide a research topic. It is a fact that a researcher often needs to read a lot of literatures in order to know state of the art and to find a research topic. Although lots of researches on paper recommendation are made, there are a few recommender systems whose goal is to help a user to find a research topic.

In this paper, we explore the application of reinforcement learning to exploratory recommender systems, whose goal is to help user to find research topic. The contributions of this paper are as follows: 1) A weighted LDA (Latent Dirichlet Allocation) method is proposed to build multi-layer topic tree; 2) Two exploratory recommendation methods based Multi-Armed Bandit (MAB) are proposed, respectively including ε-Greedy Stochastic Perturbation (ε-Greedy-SP) and Continuous Upper Confidence Bound (Con-UCB) ; 3) A paper recommender system oriented to finding research topic is developed, and its performance is tested and evaluated.

This paper is organized as following. Next, we survey the state of art on reinforcement learning and recommender system. In section 3, overall framework of developed paper recommender system is proposed. In section 4, two recommendation methods based on MAB are presented. In section 5, the experiments are made and system performance is tested. Finally, conclusions and future research is discussed.

II. RELATED WORK

Finding a research topic often starts by reading a large number of papers, so paper recommender system is very useful for scholars to select their research fields. Tang et al. used focused collaborative filtering which is added with users clustering for paper recommendations [1]. Lee used collaborative filtering methods to develop a paper recommender system [2]. Beel et al. compared several different evaluations for research paper recommendation [3]. Melnick focused on how to display a research paper [4], and the result showed that organic recommendations performed better than commercial recommendations.

DOI reference number: 10.18293/SEKE2018-063
The research is supported by National Natural Science Foundation of China (No.71271209, No.71331007).

Personalized recommendation means to recommend objects, such as goods, music, websites or papers, based on analysis of unique user through the recommendation process. In the past ten years, machine learning methods have been introduced to recommendation field. Wang J. et al. [5] combined Convolutional Neural Network and Wide & Deep model to recommend articles and applied attention model to solve the sequential problem. Tajima A et al. [6] used Factorization Machine to extract features and Gated Recurrent Unit to recommend news for large amount of users. Yang C. et al. [7] combined CF and semi-supervised learning to recommend POIs. However, all these methods assume that there are some labeled data for filling the matrix (CF methods) or training the network (NN methods), so the cold start problem is still not solved very well and the fluctuation of users' preferences cannot be evidently detected.

The conception of reinforcement learning is firstly proposed by Barto [8], who defined reinforcement learning as a goal-oriented learning from interaction. Multi-Armed Bandit problem is a classical problem in reinforcement learning, and the research about MAB has lasted for decades. The latest achievements are as follows. Xu [9] used MAB models to balance exploiting user data and protecting user privacy in dynamic pricing. Shahrampour [10] proposed a new algorithm for choosing the best arm of MAB, which outperforms the state-of-art. Lacerda [11] proposed an algorithm named as Multi-Objective Ranked Bandits for recommender systems.

III. System Design of Exploratory Recommender System

A. System Overview

The overview of our proposed recommendation method is shown in Fig.1. It includes three key modules, respectively *Topic Tree Building Module*, *Recommendation Module* and *User Preferences Updating Module*.

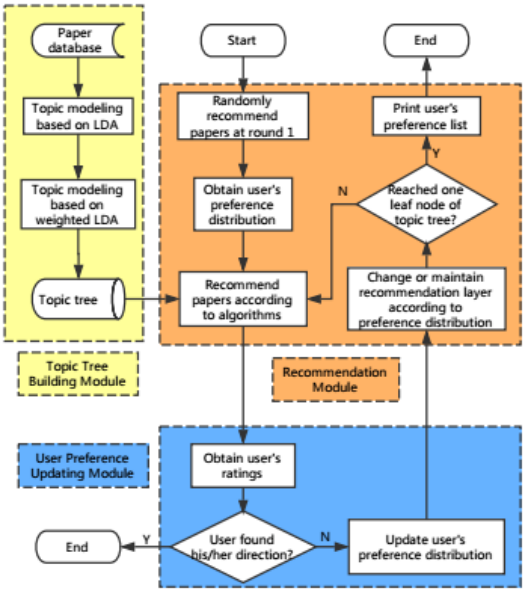

Figure 1. Exploratory Recommendation Process

In the topic tree building module, firstly, all literatures are separated into *N* topics of 1st layer based on Latent Dirichlet Allocation (LDA) method and get the Distribution Matrix (*DM*) and Belong Matrix (*BM*) of 1st layer. Then, a weighted-LDA method proposed in section 3.2 is used to subdivide each topic of 1st layer into *N* topics, so we get *N***N* topics at 2nd layer. We generalize weighted-LDA to more layers and get *DM* and *BM* of all layers. So, a topic tree is built, in which every non-leaf node has *N* child nodes. The process will be described in detail in Section III(B).

Recommendation module is the core module of our system. For a new user, we select papers from different topics at 1st layer randomly as the recommendation of 1st round. Then we obtain user's preference distribution of *N* topics at 1st layer according to feedback (ratings to the recommended papers). Afterwards, recommendations are carried out in the following steps.

Step 1. Recommend papers based on preference distribution (Section 4);
Step 2. Obtain user's ratings to the recommended papers;
Step 3. User has found research topic? If yes, move to ***Step 6***; if no, move to ***Step 4***;
Step 4. Obtain user's new preference distribution from **User Preferences Updating Module**;
Step 5. Judge how the recommended layer should change (Section 3.4). User preference conforms to tracking condition and no lower layers? If yes, move to ***Step 6***; if no, move to ***Step 1***.
Step 6. Get user's preference list and the process end

Figure 2. Process of Recommendation Module

The function of user preferences updating module is to update user preferences of different layers according to user's ratings to the recommended papers, and its procedure is detailed in Section 3.3.

B. Topic Modeling

The structure of the topic tree is shown in Fig. 3.

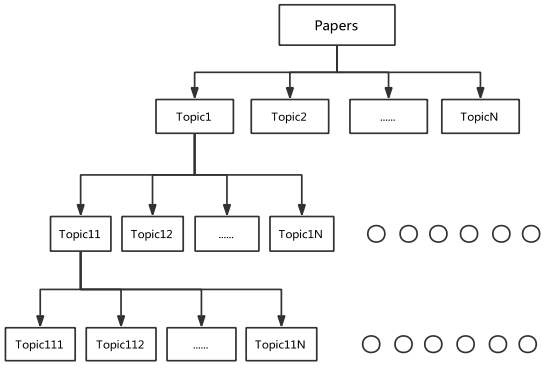

Figure 3. Structure of Topic Tree

Topic modeling of 1st layer based on LDA. Latent Dirichlet Allocation (LDA) is a topic discovery model for documents. According to LDA, each word from a paper obeys the following process: select a topic related to this paper with some probability and select a word from the selected topic with some probability. The process of LDA can be explained as factorizing the known Document-Word Matrix.

So, each paper p_m is mapped to a *N*-dimensional vector using basic LDA method, represented by $(d_{m,T_1}, \dots, d_{m,T_n}, \dots, d_{m,T_N})$. And we will get topic distribution matrix (denoted by TDM) at 1st layer, shown in Table 1.

TABLE I. TOPIC DISTRIBUTION MATRIX (1ST LAYER)

	T_1		T_n		T_N
p_1	d_{1,T_1}		d_{1,T_n}		d_{1,T_N}
..	…		…		…
p_m	d_{m,T_1}		d_{m,T_n}		d_{m,T_N}
..	…		…		…
p_M	d_{M,T_1}		d_{M,T_n}		d_{M,T_N}

Based on topic distribution matrix, we can determine the topic that paper p_m belongs to as following,

$$T_m^* = indexof\left(argmax\{d_{m,T_i} | i = 1, \dots, N\}\right) \quad (1)$$

Thus, we get a belong-to matrix at 1st layer (denoted by *BM*(1st layer)), where each row of *BM* (1st layer) is a *N*-dimensional vector contains of a one and *N*-1 zeros. It will be used in *randSelect*() function in recommendation methods.

Subtopic modeling based on weighted-LDA. We describe the process of 2nd layer and it's easy to promote to all the lower layers. At first, which papers should be brought into the subdivision of which topics is determined as follows: For the distribution vector of

$$p_m = (d_{m,T_1}, \dots, d_{m,T_n}, \dots, d_{m,T_N}) \quad (2)$$

We sort it in descending order and get an adjusted vector

$$(d_{m,T_{adj_1}}, \dots, d_{m,T_{adj_n}}, \dots, d_{m,T_{adj_N}})$$

Then accumulate the vector until the sum is greater than a threshold θ. And we can get the related topic of paper p_m,

$$\text{Topic}(p_m) = \{T_{adj_1}, \dots, T_{adj_{count}}\} \quad (3)$$

TABLE II. AN EXAMPLE OF SUBDIVISION MEMBER DETERMINATION

$paper\_id$	d_{T_1}	d_{T_2}	d_{T_3}	d_{T_4}	d_{T_5}
p_1	0.072	0.113	0.496	0.236	0.083
p_2	0.129	0.041	0.062	0.728	0.041
p_3	0.192	0.137	0.253	0.283	0.136

For example, as shown in Table 2, when $\theta = 0.6$, p_1 is related to topic T_3 and T_4, p_2 is related to T_4, p_2 is related to T_4, T_3 and T_1. After determining the participants, we use weighted-LDA for topic discovery. In the topic discovery process of basic LDA, every word in every document is not separated by the conception of weight, and is considered just as count 1. We assume that p_1, p_2 are both participants of subdivision of T_1, but $d_{1,T_1} = 0.8$ and $d_{2,T_1} = 0.2$. In this situation, p_1 and p_2 should apparently be distinguished, because p_1 belongs to T_1 more than p_2 does. The rule is defined as: the more a paper belongs to a topic, the higher weight its words will get in subdividing the topic. This is the thought of weighted-LDA. In subdivision of T_n, for every participant p_m, its Document-Word Matrix is multiplied by d_{m,T_n}, and this adjusted matrix will be the input of LDA. In this way, we will get *N* distribution matrices and *N* belong-to matrices, denoted by *DM* (2nd layer) and *BM* (2nd layer).

C. User Preference Updating

We map the user's ratings to the recommended papers to user's scores to topics through DM, which is also denoted as *d*. Figure 4 shows the updating process. At the beginning of tth round, assume the current layer is *L*, which means the user has got clear preference from layer 1 to *L*-1, we will get a preference list with the length of *L*-1, denoted as $pre^{(t-1)}$. $pre_i^{(t-1)}$ represents the most preferred topic at layer *i*. If $pre = [2,1]$, it means the user likes T_{21} at the end of (t-1)th round and the current recommendations are among the subtopics of T_{21}, which are T_{211}~T_{21N}.

Correspondingly, we maintain a *L*-length list named as $US^{(t-1)}$, which stores the user's preference scores to different layers from 1 to *L*. $US_i^{(t-1)}$ represents the scores to the topics at ith layer. It should be noticed that $User\_score$ just stores the scores alongside the user's preference path, so every element in US is an *N*-dimensional vector and $User\_score$ should be explained in conjunction with pre. When $pre = [2,1]$, US_1 represents the scores to T_1~T_N, $User\_score_2$ represents the scores to T_{21}~T_{2N}, and $User\_score_3$ represents the scores to T_{211}~T_{21N}. $Paper^{(t)}$ is the union of the recommended papers at tth round, which consists of *K* papers denoted as

$$\{Paper_1^{(t)}, Paper_2^{(t)}, \dots, Paper_k^{(t)}, \dots, Paper_K^{(t)}\}$$

The distribution of $Paper_k^{(t)}$ at ith layer is ($d_{k(t),T_{pre_1 pre_2 \dots pre_{i-1}1}}, \dots, d_{k(t),T_{pre_1 pre_2 \dots pre_{i-1}N}}$),where $k(t)$ means the id of $Paper_k^{(t)}$. At the situation of $pre = [2,1]$, the distributions of $Paper_k^{(t)}$ from 1st layer to 3rd layer are $d_{k(t),T_1}$~$d_{k(t),T_N}$, $d_{k(t),T_{21}}$~$d_{k(t),T_{2N}}$ and $d_{k(t),T_{211}}$~$d_{k(t),T_{21N}}$. User's ratings to the *K* papers are $Rating^{(t)} = \{R_1^{(t)}, R_2^{(t)}, \dots, R_k^{(t)}, \dots, R_K^{(t)}\}$, and ac is the attenuation coefficient.

Input: d (the shorthand of *DM*)
$pre^{(t-1)}$ (user's preference path at (t-1)th round)
$U^{(t-1)}$ (user's preference score at (t-1)th round)
$Rating^{(t)}$ (user's ratings to recommended papers at tth round)
ac (attenation coefficient)
Output: $US^{(t)}$

For $k = 1$ to K
$\widehat{R_k^{(t)}} = R_k^{(t)} - 2.5$
End for
$L = length(pre^{(t-1)}) + 1$
For $l = 1$ to L
$Temp\_score_l^{(t)}$
$= (\sum_{k=1}^{K} \widehat{R_k^{(t)}} d_{k(t),pre_1 pre_2 \dots pre_{l-1}1}^{(t)}, \dots, \sum_{k=1}^{K} \widehat{R_k^{(t)}} d_{k(t),pre_1 pre_2 \dots pre_{l-1}N}^{(t)})$
$US_l^{(t)} = US_l^{(t-1)} * ac + Temp\_score_l^{(t)} * (1 - ac)$
End for
Return $US^{(t)}$

Figure 4. Preference Updating

It is important to notice that $R_k^{(t)}$ is limited in [0,1,2,3,4,5] and $\widehat{R_k^{(t)}}$ is a revise of $R_k^{(t)}$. Without this process, if the score of a specific topic is close to 0, we can hardly distinguish that whether the user dislikes the topic or there are not enough recommendations from this topic, and these two situations cannot be confused. After the revise, when the score is close to 0, the confusion is the user neither likes nor dislikes the topic or not enough chances for the topic, and it's acceptable. The key is we can easily separate the topics which are preferred by the user (a relatively large positive number) and those topics the user dislikes (a relatively small negative number).

D. Backtracking and Tracking

Backtracking condition indicates that user's preference becomes not so clear at the upper layer, so the recommendation process should trace back to upper layer. On the contrary, tracking condition shows that user's preference at current layer is clear enough and the recommendation process should traverse down alongside the topic tree. The two conditions are defined as follows: Backtracking condition. At the end of t^{th} round, if $L > 1$ and

$$max_{n=1,\dots,N}\, US_{L-1,n}^{(t)} - secondMax_{n=1,\dots,N}\, US_{L-1,n}^{(t)} < \theta_2 \quad (4)$$

which means the score of the most preferred topic at upper layer is not obviously larger than the second one, we pop the last element of pre and the last *N*-dimensional vector of User_score, and let $L = L - 1$. Tracking condition. At the end of round *t*, if $L < Max\_layer$ and

$$max_{n=1,\dots,N}\, US_{L,n}^{(t)} - secondMax_{n=1,\dots,N}\, US_{L,n}^{(t)} > \theta_3 \quad (5)$$

which means the difference between the score of the most preferred topic at current layer and the score of the second one is clear enough, the recommendation process remote to lower layer, and pre should be added with indexof($max_{n=1,\dots,N}\, US_{L,n}^{(t)}$) and we add an *N*-dimensional zero vector to the tail of US, $L = L + 1$, θ_2 and θ_3 are the thresholds of the two conditions.

E. System Interfaces

Several interfaces of our system are shown as follows. Fig.5 shows the login interface. Fig.6 is the main recommendation interface, which contains of the information of recommended papers and the buttons for user to give the rating. Fig. 7 shows the word cloud generated after each round of recommendation.

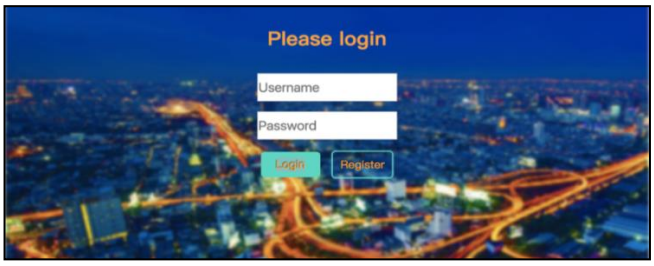

Figure 5. Login Interface of System

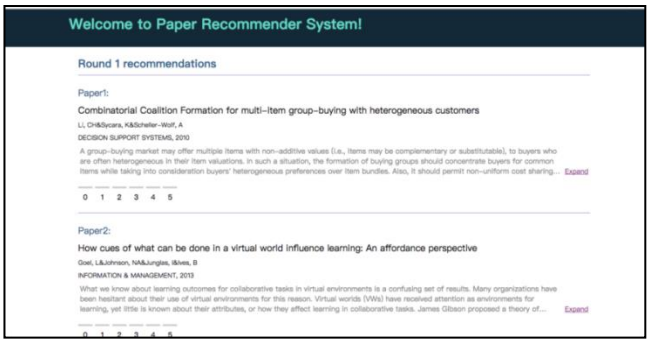

Figure 6. Recommendation Interface of System

Figure 7. Generated Word Cloud for Each Round

IV. Exploratory Recommendation Methods Based on MAB

A. ε-Greedy Stochastic Perturbation (ε-Greedy-SP)

ε-Greedy is a classic method of solving MAB model. Give a threshold named as ε, and generate a random number named as ξ, if $\xi > \varepsilon$, the arm with the highest average profit will be chosen, when $\xi \le \varepsilon$, a random arm will be selected. We apply ε-Greedy method to the situation of exploratory recommendation in the following two ways.

Classic ε-Greedy. Fig.8 shows the process of ε-Greedy. *L* is the current layer, if the generated random value $a > \varepsilon$, the most preferred topic at current layer will be chosen as T^*, else if $a \le \varepsilon$, one of the *N* topics at current layer will be chosen randomly as T^*. The purpose of *randSelect()* function is to determine related topics T^* which a paper which belongs to.

Input: *b* (the shorthand of *BM*),
$pre^{(t-1)}$ (user's preference path at (*t*-1)$^{\text{th}}$ round),
$User\_score^{(t-1)}$ (user's preference score at (*t*-1)$^{\text{th}}$ round),
ε (threshold of ε-Greedy)
Output: $RecResult^{(t)}$

$RecResult^{(t)} = \{\}$
$L = length(pre^{(t-1)}) + 1$
For $k = 1$ to K
  $a = Rand()$;
  If $a > \varepsilon$ Then $T^* = T_{pre_1 pre_2 \dots pre_{L-1}\{n|\, \max_{n=1,2,\dots,N} User\_score_{L,n}^{(t-1)}\}}$
  Else $T^* = T_{pre_1 pre_2 \dots pre_{L-1} random(1,N)}$
  $RecResult^{(t)} \leftarrow randSelect(b_{m,T^*} = 1)$
End for
Return $RecResult^{(t)}$

Figure 8. Classic ε-Greedy Algorithm

ε-Greedy_SP. Based on classic ε-Greedy algorithm, we add a stochastic perturbation to the current preference scores in ε-Greedy_SP for catching user's preference as soon as possible. As show in Fig.9, the function *randVector(ε)* is used to generate a *N*-dimensional vector, consist of one ε and *N*-1 zeros. The design of *randVector(ε)* ensures the exploration of the recommendation process.

Input: b (the shorthand of *BM*)
$pre^{(t-1)}$ (user's preference path at (t-1)th round),
$US^{(t-1)}$ (preference score at (t-1)th round)
ε (threshold of ε-Greedy)
Output: $RecResult^{(t)}$

$RecResult^{(t)} = \{\}$
$L = length(pre^{(t-1)}) + 1$
For $k = 1$ to K
$US'^{(t-1)}_{L,n} = US^{(t-1)}_{L,n} + randVector(\varepsilon)$
$T^* = T_{pre_1 pre_2 \ldots pre_{L-1}\{n|\max_{n=1,2,\ldots,N} US'^{(t-1)}_{L,n}\}}$
$RecResult^{(t)} \leftarrow randSelect(b_{m,T^*} = 1)$
End for
Return $RecResult^{(t)}$

Figure 9. ε-Greedy_SP Algorithm

The design of ε. For any method derived from ε-Greedy, the value of ε is apparently an important aspect. We believe that there is a close connection between the value of ε and user's preference. When the preference is not so clear, ε should be relatively large for a greater degree of exploration. On the contrary, with the difference between the preferred topics and the disliked ones is large enough, ε should decrease to a relatively small value. We design ε as follows in our system.

$$\varepsilon = 1 - S(gap) \tag{6}$$

$$gap = \max_{n=1,\ldots 1N} US^{(t-1)}_{L,n} - \underset{n=1,\text{on}N}{\text{second}\max} US^{(t-1)}_{L,n} \tag{7}$$

where $S()$ is the Sigmoid function. $S()$ can be stretched or shrunk in both axes according to the actual situation.

B. Continuous Upper Confidence Bound (Con-UCB)

The classic Upper Confidence Bound (UCB) method is to express exploitation and exploration as two parts of a total score. The basic formula of UCB is $\text{Score}_i = \overline{\text{Gain}_i}(t) + \sqrt{\frac{2\ln t}{T_{i,t}}}$, in which the first part represents the average gain of an arm and the second part represents the possibility of the arm, where t is the round numbers, $\overline{\text{Gain}_i}(t)$ denotes the average gain of arm i and $T_{i,t}$ is the times that arm i used. For exploratory recommendation, UCB method needs to combining the average $User\ score$ and how many times the topics are recommended which is calculated through *BM*. For example, if a recommended paper belongs to T_1 according to *BM* (1st layer), the $T_{i,t}$ of T_1 in the basic UCB formula will be added by 1.

Continuous-UCB. Continuous-UCB is a UCB based method by considering the weights of topics. The difference in Continuous-UCB is that we alternate *BM* in classic UCB with *DM*, that is to say in Continuous-UCB, we sum the distributions on T_1 of all the recommended paper as the $T_{i,t}$ of T_1 in the UCB formula. The process is shown in detail in Figure 10. This method also gives an idea to the situation that there are complicated matches between recommendation items and categories.

Input: d (the shorthand of *DM*)
b (the shorthand of *BM*)
$pre^{(t-1)}$ (user's preference path at (t-1)th round)
$UsS^{(t-1)}$ (user's preference score at (t-1)th round)
$round$ (rounds made at current layer),
$Paper$ (all the recommended papers)
Output: $RecResult^{(t)}$

$RecResult^{(t)} = \{\}$
$L = length(pre^{(t-1)}) + 1$
For $k = 1$ to K
For $n = 1$ to N
$UCB\_score^{(t)}_{L,n} = User\_score^{(t-1)}_{L,n} + \sqrt{\frac{2 * \ln(K * round)}{\sum_{time=t-round}^{t-1} \sum_{num=1}^{K} d_{paper^{(time)}_{num}, T_{pre_1 pre_2 \ldots pre_{L-1} n}}}}$
End for
$T^* = T_{pre_1 pre_2 \ldots pre_{L-1}\{n|\max_{n=1,2,\ldots,N} UCB\_score^{(t)}_{L,n}\}}$
$RecResult^{(t)} \leftarrow RandSelect(b_{m,T^*} = 1)$
End for
Return $RecResult^{(t)}$

Figure 10. Con-UCB Algorithm

In Fig.10, *round* represents order of rounds when recommendation are made at current layer, $Paper$ represents the set of all the recommended papers, $paper^{(t)}_k$ represents the k^{th} paper at round *t*.

V. EXPERIMENTS

A. Experiment Dataset and Design

In this paper, Web of Science journal articles in from 2009 to 2013 are used, and they are from the Science Citation Index Expanded (SCI- EXPANDED), Social Sciences Citation Index (SSCI) and Arts and Humanities Citation Index(AHCI).

In order to preliminarily test our developed system and compare two proposed MAB methods, 5 undergraduates are invited to participate in experiments and to determine their research topics. Three of them use recommender system based on ε-Greedy_SP while other two students use recommender system based on Con-UCB.

Number of topics at every layer is set to 5, *Max_layer* is 2 and *K* is 5. We will compare the two methods in several indices and analyze the focusing process of user preferences through the change of their ratings in Section 5.2. Besides, the empirical analysis of two typical user shows the flexibility of our system to difference type of users.

B. Experiment Result

Overall Result. First, average ratings of five invited students are shown in Figure 11, where user1, user2 and user3 employ ε-Greedy_SP while user4 and user5 employ Con-UCB.

It is easy to find that users' ratings increase gradually and the preferences are focused little by little. It shows that the proposed paper recommender systems can catch and focus user's preference gradually.

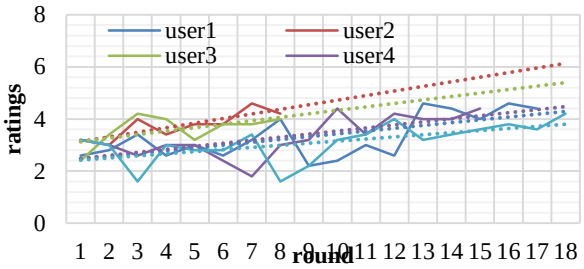

Figure 11. Average ratings of five students

More detailed result is shown in Table 3. It shows that ε-Greedy_SP performs better than Con-UCB in the case of insufficient samples. Due the lack of samples, the result can be also caused by the individual difference, so the result of the algorithm comparison here is just for reference and is not on a high confidence level.

TABLE III. COMPARISON OF E-GREEDY_SP AND CON-UCB

Algorithm	Average round	Average rating	Average variance ratio
ε-Greedy_SP	11	3.491	6.250%
Con-UCB	16.5	3.2	5.455%

Empirical Analysis. In order to understand focusing processing of user's preference, we select user 3 as for empirical analysis. The recommendation process will promote to 2nd layer only if *US* conforms to tracking condition at least once, the preference at 2nd layer has no value at 1st round, and it could be discontinuous because of backtracking condition, so we choose the topics at 1st layer to analyze the focusing process. Perference Score of user3 to $T_1 \sim T_5$ is shown in two diagrams from Figure 12 to Figure 13. It is visible that after 2nd round, user3 has a comparatively preference of T_5. The score of T_5 is growing steadily and opens up a gap with other topics gradually until user3 finds the direction of research. This result shows that user3 has a roughly concept of his research topic, and our system here is a concrete refinement tool for user3.

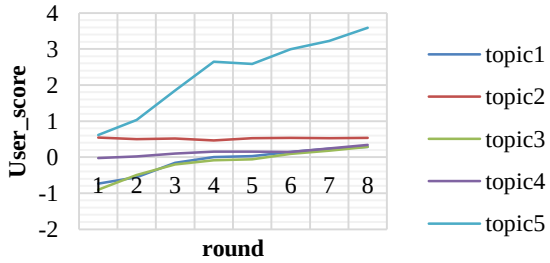

Figure 12. Line chart of user3's preference score

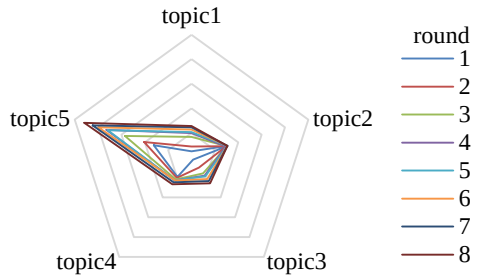

Figure 13. Radar chart of user3's preference score

VI. CONCLUSION

Recommender system devote to finding individual items to users. Traditional recommender system does not work well when helping users to finish a task. In this paper, we propose a novel paper recommender system for finding research topic, where the user only needs to give feedbacks of recommended items. Two exploratory recommender methods based on MAB models are proposed. A prototype system is developed, and show good performance. In the future, the system will be tested by more invited students.

REFERENCES

[1] Tang T. Y., Mccalla G.: Mining implicit ratings for focused collaborative filtering for paper recommendations. The Workshop on User and Group MODELS for Web-Based Adaptive Collaborative Environments, International Conference on User Modeling, 45-56 (2003)

[2] Lee J., Lee K., Kim J. G.: Personalized Academic Research Paper Recommendation System. Computer Science (2013).

[3] Beel J., Langer S.: A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. Research and Advanced Technology for Digital Libraries (2015).

[4] Melnick S. L., Shahar E., Folsom A. R.: Sponsored vs. Organic (Research Paper) Recommendations and the Impact of Labeling. International Conference on Theory and Practice of Digital Libraries, 395-399 (2013).

[5] Wang J.: Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors' Demonstration. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2051-2059 (2017).

[6] Tajima A.: Embedding-based News Recommendation for Millions of Users. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1933-1942 (2017).

[7] Yang C., Bai L., Zhang C.: Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1245-1254 (2017).

[8] Barto A. G.: Reinforcement learning. Springer Berlin Heidelberg, 665-685 (1998).

[9] Xu L., Jiang C., Qian Y.: Dynamic Privacy Pricing: A Multi-Armed Bandit Approach With Time-Variant Rewards. IEEE Transactions on Information Forensics & Security 12(2), 271-285 (2017).

[10] Shahrampour S., Noshad M., Tarokh V.. On Sequential Elimination Algorithms for Best-Arm Identification in Multi-Armed Bandits. IEEE Transactions on Signal Processing, (2017).

[11] Lacerda A.: Multi-Objective Ranked Bandits for Recommender Systems. Neurocomputing (2017), 246.