Paper:

Consideration of Strategy for Selecting Probing Items in Interactive Recommendation

Yasufumi Takama[1], Mingman Xu, Hiroki Shibata[2]

Graduate School of System Design, Tokyo Metropolitan University
E-mail: ytakama@tmu.ac.jp[1], hshibata@tmu.ac.jp[2]

This paper discusses a strategy to determine probing item. The definition of probing item is provided in the paper: the probing item is an item by showing it to a user the system can obtain effective feedback for further recommendation. This paper compares several strategies for determining probing items. Probing items are presented to users especially at their Sign-up process to obtain their feedback, from which initial user profiles are constructed. Therefore, how to determine probing items are critical for designing recommender systems. This paper reports the results of comparing several strategies by simulating user behaviors using real dataset.

Keywords: Recommender system, Online Recommendation, Cold-start, Probing items, Rating matrix

1. Introduction

This paper discusses a strategy to determine probing item used in order to construct a user profile in the situation of online recommendation. Services that include recommendation functionality determines items that are considered to be recommended based on the information such as user preference and interest in items. Especially, services employing so-called Collaborative Filtering determines recommending items based on the information related to the interaction between users and the system, such as the history of user's behavior like browsing, purchasing, and reviewing. For a study on this Collaborative Filtering, generally, a large scale of interaction data prepared in advance is used. However, to make the system capable of use on actual service, it must be assumed that interaction data is provided sequentially (in online manner) through the use of service by users.

As one approach to collect interaction data, it can be considered to make the system construct the user profile when the user begins to use the service by obtaining the feedback through some questions to the user related to items. This approach is called sign-up process in this paper. At the sign-up process, a strategy to determine an initial set of items to be shown to a user is critical because it affects the quality of obtained user profile that is used also in the subsequent process.

This paper defines a probing item as following: by showing a probing item to a user the system can obtain effective feedback from the user. Strategies to determine a probing item in the situation that the system must recommend a new user items in online manner are discussed. In detail, a method that keeps updating the user profile in online manner with the iteration of recommending items and obtaining user's evaluation as the feedback to the items is proposed. The whole process managed in the proposed method is divided into 2 parts, the sign-up process that constructs an initial user profile, and the recommendation process that uses Collaborative Filtering.

The experiment to validate the method is conducted on dataset so-called rating matrix, rather than user-experiments, which needs the help of collaborators. Instead of asking a real user to try the system, a simulated user's behavior that is modeled based on the rating matrix is used. This means of experiments reduce the cost of experiments compared with user experiments and make the experiment scalable, to the level of real recommender systems. From the point of view that recommender systems must archive accurate recommendations and minimize the user effort at the sign-up process, this paper shows comparisons and discussions on strategies to determine probing items.

2. Related work

2.1. Collaborative Filtering

Methods of recommendation are categorized into 2 groups. One of them is content-based filtering that employs features of items. The other one is Collaborative filtering that employs the behavioral history of users' evaluation to items. In Collaborative Filtering, the user's evaluation for items that the user has not seen is predicted taking the rating matrix that is composed of users' behavioral history mentioned above as an input.

Collaborative Filtering is further divided into 3 categories, memory-based method, model-based method, and hybrid method of these 2 methods. A Memory-based method predicts the user's evaluation using similarity coefficient among users or items. That similarity coefficient is defined based solely on a rating matrix. As a representative method of memory-based methods, k-NN (k-Nearest Neighbor) [3] can be mentioned. It determines a set of users that resemble the target user who is recommended, based on the Pearson correlation coefficient, then using

the ratings assigned by users in the set and the correlation coefficient among them, it predicts ratings by the target user to items.

Model-based method predicts ratings using the model that is built from a rating matrix or item information. Specifically, Factorization-based methods that generate latent feature vectors of user and item from a rating matrix [8, 13], and Deep Learning-based methods have been proposed [4, 5].

2.2. Cold-start Problem

Cold-start has been discussed because it is one of the major problems in Collaborative Filtering, in which after a new user or item is added to the system database the system cannot recommend accurately or at all unless enough amount of behavioral history of users have been accumulated.

Various methods have been proposed as a solution for the Cold-start problem. Those methods are categorized into either personalized or non-personalized methods. Non-personalized methods include random recommendation [6], and methods recommend items based on Entropy or popularity [10, 11]. As examples of personalized methods, recommendation based on questionnaire [1, 18], method employing kind of means that extrapolate the rating matrix [7, 14], and methods that employ social information such as the history of user activity on social network services.

2.3. Interactive Recommender System

Most of the evaluations for recommender system are conducted based on errors such as MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) between predicted ratings and real ratings, or indexes such as recall and precision that are calculated from list of recommended items. It can be said that these indexes are intended to be used for the evaluation of the capability of the system to recommend just one item using accumulated user's behavioral history.

However, in the realistic cases that recommender system runs, because user's behavioral history can be obtained sequentially through item recommendation, suitable set of items to be recommended to a user affect the future status of the system. Focusing on this case in which feedback from the interaction between users and the system must be utilized, studies on Interactive Recommender System have been done.

There are some studies that construct the interactive recommender system in the framework of Multi-armed Bandit Problem in which the sequential determination process of recommending item on the feedback from users is considered so that it solves the trade-off between exploitation and exploration. Zhoa et al. have proposed an interactive collaborative filtering method by extending Probabilistic Matrix Factorization [13]. It determines recommending items using Thompson Sampling, and updates user profile sequentially based on feedback (ratings). Christakopoulou et al. have shown comparisons among various presentation strategies of items such as UBC (Upper Confidence Bound), Greedy as well as Thompson Sampling [2].

As another example, Sun et al. have proposed recommender system that interact with users in natural language [16]. It is capable of both 2 interactions, i.e., interaction to extract the conditions or properties that item satisfies, and interaction for actual recommendation. A strategy to determines which interaction is suitable at each time is learned. Wegba et al. have proposed interactive movie recommender system that assumes to recommend unless users satisfy. Considering the iterative process unless users satisfy as an exploration over the dataset, it aims to improve its user experience on recommendation by providing better exploration experience by the story telling that uses representative characters for each movie.

3. Construction of simulated experiments

For evaluation of recommender systems on sequential recommendation case, 2 approaches, online user experiments with collaborators and offline experiments based on dataset (rating matrix), are considered. While former approach has the merit of which it can obtain feedback from users, it is not suitable to large scale experiments and comparisons on various systems or algorithms. On the other hand, although offline experiments has disadvantages such that it cannot evaluate the prediction performance on items that have not been rated, it can be used on experiments to see the effect of large amount of users and comparison on various system in the same condition. Therefore, this study considers the offline experiment is a valid choice for the purpose such as enumerating possible algorithms, hypothesis to be investigated, before actual user-experiment based on dataset.

From the above-mentioned point of view, this paper employs offline experiments using dataset. This section describes dataset that is used, process of sequential recommendation, and a strategy to determine probing item to obtain effective feedback from users.

3.1. Dataset

This paper employs MovieLens 100K Dataset[1]. There are 100000 ratings assigned one of integers 1, 2, ..., 5 by 943 users to 1682 items.

A certain amount of ratings for each user are needed to construct the user model from which system can simulate the user's behavior. Therefore, this paper employs only the histories corresponding user of which gives rating to at least 200 items, for the sake of appropriate evaluation of sequential recommending strategy. There are 149 users in the dataset who satisfy this constraint. The data by the other users are employed for recommendation itself.

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

3.2. Process of Sequential Recommendation

It is assumed that users to be recommended are such one who have never used the system. Namely, there is no behavioral history for such new users in the system to determine the items that should be recommended. Therefore, this paper proposes to recommend upon the following 2 steps.

1. Sign-up process: the system does the recommendation for the purpose to establish user profile.
2. Recommendation process: the system does the recommendation for the purpose to recommend items that a target user seems to like.

Sign-up process aims to obtain the feedback that can be used to establish effective user profile in the recommendation process. Item that is presented in this process is called probing item in this paper. At the recommendation process, commonly used user-based collaborative filtering (KNNWithMeans[2]) is employed.

General notation in this paper is explained here first. For a finite set A, $|A|$ denotes the number of elements in A, and $\{a;P(a)\}$ denotes a set of element a that satisfies a proposition (condition) $P(a)$. $(a_1,a_2,...,a_n)$ denotes a tuple or family in formal set theory in which the order has the meaning and the same element can exist, while $\{a_1,a_2,...,a_n\}$ denotes a set in which order does not have meaning and duplicated element does not exist.

Letting $Q=\{1,2,...,5\}$ be a set of possible ratings, $\varnothing$ be a missing value (no rating is given), U be a set of users, $U'\subset U$ be a set of target users who have rated at least 200 items, V be a set of items, $n=0,1,...$ be the steps of the process, r^0 $(\in R)$ be an given rating matrix (dataset) where $R=(Q\cup\{\varnothing\})^{U\times V}$ is a set of all possible rating matrix. In addition, let r^n $(\in R)$ be the rating matrix at step n, and $\hat{r}^n$ $(\in R)$ be the predicted rating matrix using KNNWithMeans based on r^n. Note that using the definition r^0, U' is formally defined as,

$$U'=\left\{u;u\in U,\left|\left\{v;v\in V,r^0_{u,v}\neq\varnothing\right\}\right|\geq 200\right\}.$$

As another example of the relation between these definitions and the dataset, it can be seen that if $A=\left\{(u,v);r^0_{u,v}\neq\varnothing,u\in U,v\in V\right\}$, then $|A|$ is a number of ratings in a dataset, therefore, for MovieLens 100K $|A|=10^5$.

For each target user u $(\in U')$, the sequential recommendation defined for each step $n=1,2,...$ is done as following.

1. Initialization ($n=0$): determine real rated items V^0_u, and initial set of sequentially rated items V^1_u, so that $V^0_u=\left\{v;r^0_{u,v}\neq\varnothing\right\}, V^1_u=\emptyset$. Rating matrix is initialized as $\forall u'\in U\ v'\in V$,

$$\left(u=u'\wedge v'\in V^0_u\right)\Rightarrow r^1_{u',v'}=\varnothing,$$
$$\left(u\neq u'\vee v'\notin V^0_u\right)\Rightarrow r^1_{u',v'}=r^0_{u',v'}.$$

That is, rating histories of the target user is removed from the dataset, and it cannot be used to predict the rating until it is observed through interaction.

2. Sign-up process ($n=1,2,...,10$):
   a. Obtain N-length sorted list v as
   $$v=(v_i;v_i\in V-V^n_u,i=1,2,...,N),$$
   where the order is defined as, $\forall i,j=1,2,...,N$
   $$i<j\Leftrightarrow v_i<^n v_j,$$
   and present the list v to the target user. Order relation $<^n$ for this sort is determined following means described later around Eq. (1) in section 3.3.
   b. Find $\hat{v}$ $(\in V)$ such that,
   $$\hat{v}=\min\left\{v_i;v_i\in V^0_u,i=1,2,...,n\right\}.$$
   In other words, it is the first item in the list which u have rated. Order relation $<^n$ is used to define above min.
3. Recommendation process ($n=11,12,...,20$):
   a. Generate predicted ratings using the recommendation method KNNWithMeans, and an item $\hat{v}$ $(\in V)$ such that
   $$\hat{r}^n_{u,\hat{v}}=\max\left\{\hat{r}^n_{u,v'};v'\in V^0_u-V^n_u\right\}$$
   is presented to u. Order relation $<^n$ is used to define above max.
4. Update ($n=1,2,...,20$): obtaining the rate $r^{n+1}_{u,\hat{v}}$ as feedback of $\hat{v}$ and making the other ratings invariant, update the states so that, $\forall u'\in U\ v'\in V$,
   $$\left(u'\neq u\vee v'\neq\hat{v}\right)\Rightarrow r^{n+1}_{u',v'}=r^n_{u',v'},$$
   and $V^{n+1}_u=V^n_u\cup\{\hat{v}\}$. Keep the iteration.

3.3. Strategy to Determine Probing Item

To determine probing item at the sign-up process, this paper proposes 9 strategies and shows a comparison among them. All strategies differ only in sorting measure $f:V\to\mathbb{R}$ from which order relation $<^n$ is defined. Here first necessary definitions to define the strategies are described.

Sorting measure $f:V\to\mathbb{R}$ is defined so that it satisfies $\forall v',v''\in V$,

$$v'<^n v''\Leftrightarrow f(v')\geq f(v''). \quad . . . . . (1)$$

That is, the sort is conducted by following the order $<^n$ defined by f here, in which the higher value $f(v')$ has, the higher rank (low value of ranking index) v' has in the presented list v to a user. Letting a subset of users who rated item v be $U^k_v=\left\{u;u\in U,r^k_{u,v}\neq\varnothing\right\}$, The mean value,

$$\overline{r^k}_v=\frac{\sum_{u'\in U^k_v}r^k_{u',v}}{|U^k_v|},$$

2. http://surpriselib.com/

is defined. $H_A(v)$ is entropy on the domain A $(\subset Q \cup \{\varnothing\})$ that is defined as,

$$H_A(v) = -\sum_{j \in A} p_{v,j} \log(p_{v,j}), \quad \ldots\ldots\ldots \quad (2)$$

where

$$p_{v,j} = \frac{n_{v,j}}{\sum_{l \in A} n_{v,l}},$$

and $n_{v,j}$ is the number of users who rate v as j.

With the above-mentioned definitions, 9 strategies are defined for each $v' \in V$ as followings.

1. Random: define $f(v')$ at random.
2. Popularity: define $f(v')$ to be a popularity $P(v')$ (> 0). Popular items are tended to be recommended.
3. Mean: define f as $f(v') = \overline{r^n}_v$
4. Variance [6]: define f as

$$f(v') = \frac{1}{|U_v^k|} \left(\sum_{u' \in U_v^k} (r_{u',v}^k - \overline{r^n}_v)^2 \right).$$

   This aims to conserve the diversity of items in a list.
5. Pure entropy [6]: define $f(v') = H_Q(v)$, with Eq. (2),
6. Entropy0 [11]: define $f(v') = H_{Q'}(v), Q' = Q \cup \{\varnothing\}$.
7. Ent*Pop [10]: employs the definition

$$f(v') = H_Q(v') P(v').$$

8. Ent*(log Pop) [10]: employs the definition

$$f(v') = H_Q(v') \log P(v').$$

9. HELF [11]: Harmonic mean of Entropy and Logarithm of rating Frequency employs the definition,

$$f(v') = \frac{2F(v')H'(v')}{F(v') + H'(v')},$$

   where,

$$F(v') = \frac{\log P(v')}{\log |T|}, T = \left\{ (u', v''); r_{u',v''}^1 \neq \varnothing \right\}$$

   and

$$H'(v') = \frac{H(v')}{\log |Q|}.$$

4. Experiments

4.1. Evaluation Index

This paper evaluates strategies to determine probing item from the 2 points of view, effort that users take at the sign-up process, and recommendation performance in the recommendation process. As mentioned so far, at the sign-up process by presenting sorted item list to a target user following the determination strategy of probing item, the system keeps asking the user to find a movie that the user have seen already from the top of the list and rate the item. The time to take before the user find the item can be seen as the effort. Therefore, in this experiment, the user's effort defined based on the rank in the presented list of found items that the user have rated in the past. It can be said that the higher rank the found item has, the less effort users have token.

Evaluation of the recommendation performance at the recommendation process is conducted based on RMSE and MAE. Both RMSE and MAE are an index to evaluate the accuracy of prediction for ratings, and are defined as follows respectively for each step $n = 1, 2, \ldots$ for user $u \in U$.

$$\mathrm{RMSE}(n,u) = \left[\frac{1}{\left|V_u^{n+1}\right|} \sum_{v \in V_u^{n+1}} (\hat{r}_{u,v}^n - r_{u,v}^{n+1})^2 \right]^{0.5},$$

$$\mathrm{MAE}(n,u) = \frac{1}{\left|V_u^{n+1}\right|} \sum_{v \in V_u^{n+1}} \left| \hat{r}_{u,v}^k - r_{u,v}^{n+1} \right|.$$

Another index such as precision and recall assuming so-called top k recommendation can be considered, however, this paper employs only these RMSE and MAE as evaluation indexes for the sake of discussing the history of recommendation performance.

4.2. Experimental Results

For each user in 149 users who rated at least 200 times, using the process described in Sec. 3.3, recommendations were conducted 20 times (that is, $|U'| = 149$, and $\hat{r}^n$ calculated for $n = 1, 2, \ldots, 20$). The histories of averaged RMSE,

$$\mathrm{RMSE}(n) = \frac{\sum_{u' \in U'} \mathrm{RMSE}(n, u')}{|U'|} \quad \ldots\ldots \quad (3)$$

and that of MAE,

$$\mathrm{MAE}(n) = \frac{\sum_{u' \in U'} \mathrm{MAE}(n, u')}{|U'|} \quad \ldots\ldots \quad (4)$$

for each strategy are shown in Fig. 1 and 2. As an index to measure the effort users take at the sign-up process, the history average of top ranking among all users for each strategy is shown in Fig. 3.

Based on these results, summary of qualitative comparison among the strategies in terms of user's effort at the sign-up process, at the start of the recommendation process, and the end of the recommendation process is shown

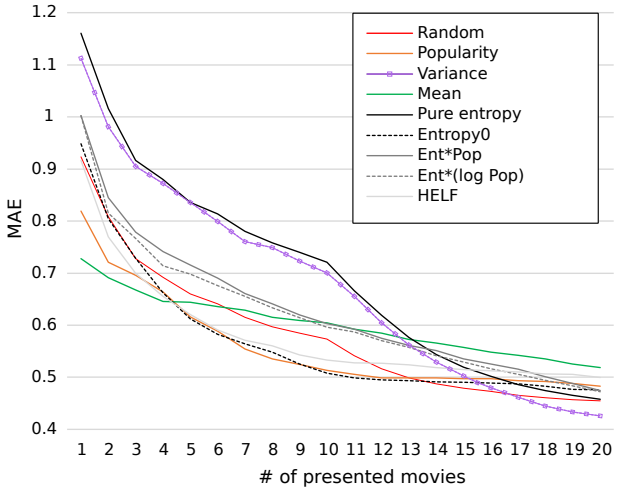


Fig. 1. Histories of MAE (Mean Absolute Error) on each step of the process among various strategies, which denotes the series of MAE$(n), n = 1, 2, ...20$ in Eq. (4).

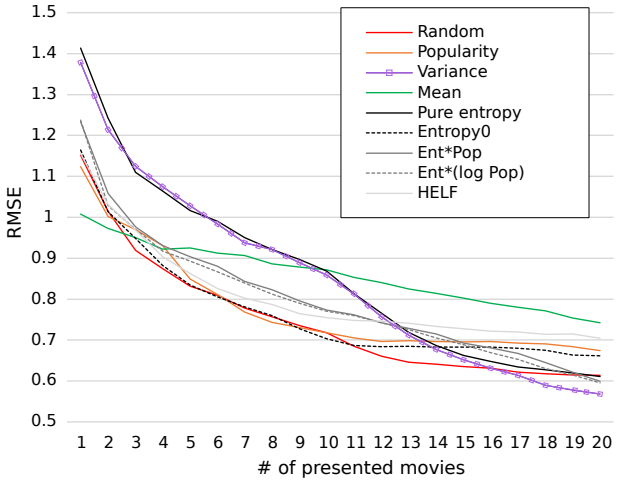


Fig. 2. Histories of RMSE (Route Means Squared Error) on each step of the process among various strategies, which denotes the series of RMSE$(n), n = 1, 2, ...20$ in Eq. (3).

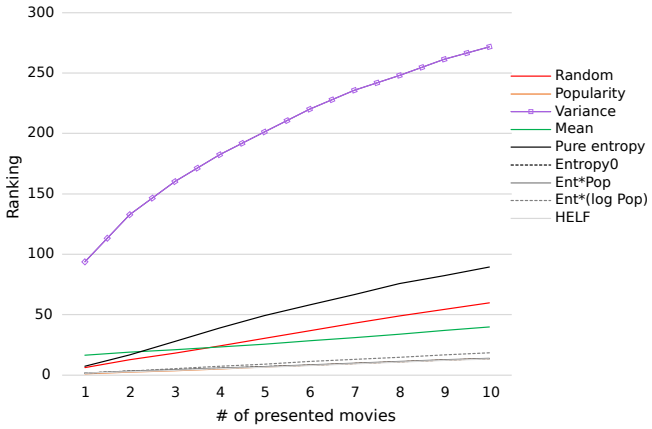


Fig. 3. Histories of index for user's effort on each step of the process among various strategies.

Table 1. Qualitative comparison among the strategies. rec. perf. denotes recommendation performance. 2 means it is good, 1 means it is comparable with the others, 0 means it is inferior to the others.

strategy	user's effort	rec. perf. (at the start)	rec. perf. (at the end)
Random	1	2	2
Popularity	2	2	1
Mean	0	0	1
Variance	1	1	1
Pure entropy	1	0	2
Entropy0	2	2	1
Ent*(Pop)	2	1	2
Ent*(log Pop)	2	1	2
HELF	2	1	1

in Tab. 1. The meanings of the numbers in the table are as follows, 2: means it is good, 1: means it is comparable with the others, 0: means it is inferior to the others.

At the sign-up process, the tendency that user's effort is low on the strategies that take popularity in its account is observed. It is supposed that the high possibility for popular items likely to be seen by most of the users contributes mainly to reducing the user's effort. Entropy0 shows low user's effort. It is supposed to be caused by the property of Entropy0 that it tends to have high value if the dataset has many missing values, which shows the same effect as considering popularity. On the other hand, results on the strategies that only consider the distribution of rating values such as Variance and Pure entropy shows high user's effort. Therefore, it can be considered that combination of Popularity is needed for any case.

For recommendation performance at the start of the recommendation process, Random, Popularity, and Entropy0 show acceptable results. In the case of Random, it is supposed that presentation of various kind of items contributes to establishing a high quality user profile, by which high performance archived. However, the inferior result of Random in terms of user's effort is observed. It is supposed that the result caused by the case that system cannot place items that the user know in the high raked place at the sign-up process.

Recommendation performance at the end of the recommendation process shows more acceptable value among the strategies of Random, Variance, Pure entropy, Ent*Pop, Ent*(log Pop), than the others. It is supposed that the results of Random and Variance are caused by their property that can obtain feedbacks on diverse items. On the other hand, any notable change is observed in the result of Popularity, which gives us the thought that there is a possibility the result is caused by the shortage of feedback on diverse item at the sign-up process. Strategies that take both Entropy and Popularity into account, that is, Ent*Pop and Ent*(log Pop) show acceptable results in both indexes for user's effort and recommendation performance throughout whole the process.

5. Conclusion

This paper showed the comparative experiments on the strategies to determine probing item shown to a user that is used to establish user profile in the situation of sequential recommendation is needed, with dataset. From the result among strategies such as Random selection, selection based on Popularity, and so on., it was confirmed that strategies that take Popularity into account reduce user's effort, and strategies that take both Entropy and Popularity into account show the good result in terms both of recommendation performance and user's effort.

The limitations of this study are followings, 1) the proposed experiment framework can consider only the feedback selected from the dataset composes of users actual rating in the past, 2) the effect of interaction between users and the system for the future state of users cannot be considered in the experiment. The authors suppose that to overcoming these problems it is needed that user experiments with the help of collaborators, however, to reduce the possible strategies that should be validated before user experiments, which needs a lot of cost, the proposed approach can be employed effectively.

For the future work, adaptive strategy during the sigh-up process should be investigated. Using adaptive strategy, it is expected that more suitable probing items are found. Recent formulation of multi-armed bandit problem to express the sign-up process should be also considered. This experiment in this study solely on one dataset, movie-lens 100K, however, experiments on other datasets must be conducted in order to see the detailed property of proposed framework and its limitation.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 19K22896, 21H03553, 22H03698.

References:

[1] M. Aharon, O. Anava, N. Avigdor-Elgrabli, D. Drachsler-Cohen, S. Golan, O. Somekh. Excuseme: Asking users to help in item cold-start recommendations, 9th ACM Conference on Recommender Systems, pp. 83–90, 2015.

[2] K. Christakopoulou, F. Radlinski, K. Hofmann. Towards Conversational Recommender Systems, KDD'16, pp. 815–824, 2016.

[3] T. Cover, P. Hart. Nearest Neighbor Pattern Classification, IEEE Transaction on Information Theory, Vol. IT-13, pp. 21–27, 1967.

[4] T. Ebesu, B. Shen, Y. Fang. Collaborative Memory Network for Recommendation Systems, SIGIR'18, pp. 515–524, 2018.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. S. Chua. Neural Collaborative Filtering, WWW2017, pp. 173–182, 2017.

[6] A. Kohrs, B. Merialdo. Improving Collaborative Filtering for New Users by Smart Object Selection, International Conference on Media Features, 2001.

[7] X. Lam, T. Vu, T. Le, A. Duong. Addressing Cold-Start Problem in Recommendation Systems, 2nd International Conference on Ubiquitous Information Management and Communication, pp. 208–211, 2008.

[8] D.D. Lee, H.S. Seung. Algorithms for Non-negative Matrix Factorization, NIPS'00, pp. 556–562, 2000.

[9] C. Li, F. Wang, Y. Yang, Z. Li, X. Zhang. Exploring Social Networking Information for Solving Cold Start in Product Recommendation, International Conference on Web Information Systems Engineering, pp. 276–283, 2015.

[10] A. Rashid, I. Albert, D. Cosley, S. Lam, S. Mcnee, J. Konstan, J. Riedl. Getting to Know You: Learning New User Preferences in Recommender Systems, 7th International Conference on Intelligent User Interfaces, pp. 127–134, 2002.

[11] A. Rashid, G. Karypis, J. Riedl. Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach, ACM SIGKDD Explorations Newsletter, Vol. 10, Issue 2, pp. 90–100, 2008.

[12] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews, ACM Conference on Computer Supported Cooperative Work, pp. 175–186, 1994.

[13] R. Salakhutdinov, A. Mnih. Probabilistic Matrix Factorization, NIPS'07, pp. 1257–1264, 2007.

[14] M. Saveski, A. Mantrach. Item Cold-start Recommendations: Learning Local Collective Embeddings, 8th ACM Conference on Recommender Systems, pp. 89–96, 2014.

[15] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, J. Christensen. Social Collaborative Filtering for Cold-Start Recommendations, 8th ACM Conference on Recommender Systems, pp. 345–348, 2014.

[16] Y. Sun, Y. Zhang. Conversational Recommender System, SIGIR'18, pp. 234–244, 2018.

[17] K. Wegba, A. Lu, Y. Li, W. Wang, Interactive Storytelling for Movie Recommendation through Latent Semantic Analysis, IUI2018, pp. 521–533, 2018.

[18] M. Zhang, J. Tang, X. Zhang, X. Xue. Addressing Cold Start in Recommender Systems: A Semi-Supervised Co-Training Algorithm, 37th International ACM Conference on Research; Development in Information Retrieval, pp. 73–82, 2014.

[19] X. Zhao, W. Zhang, J. Wang. Interactive Collaborative Filtering, CIKM'13, pp. 1411–1420, 2013.