Neurocomputing 267 (2017) 635–643

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Long-term performance of collaborative filtering based recommenders in temporally evolving systems[☆]

Xiaoyu Shi[a,1], Xin Luo[a,b,*], Mingsheng Shang[a], Liang Gu[c,d,1]

[a] Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
[b] College of Computer Science and Engineering, Shenzhen University, Shenzhen 518060, China
[c] Jinan University, Guangzhou, Guangdong 510632, China
[d] Sangfor Technology Incorporation, Shenzhen, Guangdong 518057, China

ARTICLE INFO

*Article history:*
Received 7 February 2017
Revised 4 June 2017
Accepted 12 June 2017
Available online 29 June 2017

Communicated by Nianyin Zeng

*Keywords:*
Learning system
Recommender system
One-step recommendation
Long-term effect
Temporally evolving system
Bipartite network

ABSTRACT

Recommender systems benefit people at every moment in their daily life. Considerable attentions have been drawn by performance in one-step recommendation and static user–item network, while the performances of recommenders on temporally evolving systems remain unclear. To address this issue, this paper first describes an online commercial system by using a bipartite network. On this network, a recommendation-based evolution method is proposed to simulate the temporal dynamics between a recommender and its users. Then the long-term performance of three state-of-the-art collaborative filtering (CF)-based recommenders, i.e., the user-based CF (UCF), item-based CF (ICF) and latent factor-based model (LFM), is evaluated on the generated temporally evolving networks. Experimental results on two large, real datasets generated by industrial applications demonstrate that 1) optimization-based CF models like the LFM enjoy their high-prediction accuracy in one-step recommendation; and 2) entity relationship-based CF models like the ICF benefit the recommendation diversity, as well as the system health on a temporally evolving network. It turns out that in a temporally evolving system, an efficient recommender should consider both the one-step and long-term effects to generate satisfactory recommendations. Thus, it is necessary to adopt heterogeneous models, e.g., trade-off between optimization-based model and entity relationship-based model, in real systems to grasp various users' behavior patterns to improve their experiences.



## 1. Introduction

In the last two decades, people have witnessed the rapid development of the information technologies (especially the artificial intelligence [1-6]) and the Internet economy, nowadays there are a massive numbers of resources (e.g., pictures, videos, goods and etc.) on the Internet [7-8]. As a result, people are often overwhelmed with information overload that available information is too much to fit out the key part. In such context, various information filtering and optimization tools have been emerged [9-14]. Among them, recommender systems, which predict the user's potentially interesting items via analyzing user historical behaviors, are considered as a promising way to extract valuable data [15-16]. Therefore, they have become attractive for many fields. For example, Amazon, Netflix and YouTube, adopt recommender systems to gain a better user satisfaction and user experience to benefit their profits further [17-20].

In the last two decades, different kinds of recommenders have been proposed based on various ideas and concepts. Among them, collaborative filtering (CF) is one of the most popular and effective recommender [21-23]. The research on the CF recommenders with varying ideas can be divided into the Neighborhood Based Model (NBM) and the Latent Factor Model (LFM) respectively. For

☆ This work was supported in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences, in part by the International Joint Project through the Royal Society of the U.K. and the National Natural Science Foundation of China under grant 61611130209, by the National Natural Science Foundation of China under grant 61402198, grant 61602434, grant 91646114, and grant 61672136, in part by the Young Scientist Foundation of Chongqing under grant cstc2014kjrc-qnrc40005, in part by the Chongqing Research Program of Basic Research and Frontier Technology under grant cstc2015jcyjB0244 and Youth Innovation Promotion Association CAS, no. 2017393.

* Corresponding author at: Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China.

*E-mail addresses:* xiaoyushi@cigit.ac.cn (X. Shi), luoxin21@cigit.ac.cn, luoxin21@gmail.com (X. Luo), msshang@cigit.ac.cn (M. Shang), os.liang@gmail.com (L. Gu).

[1] X. Shi and L. Gu contributed equally to this work.

http://dx.doi.org/10.1016/j.neucom.2017.06.026
0925-2312/

the NBM, it is a kind of entity relation-based method. It focuses on building the neighborhoods of target entity (i.e., user or item) based on the similarity between different entities, then makes commendations according to the historical behaviors of target's neighbors. It further includes two different types of forms that the user-oriented (U-NBM) and item-oriented (I-NBM) recommendation [20].

Compared with NBM, LFM is one of numerical optimization method, it first maps both users and items into a joint latent factor space, then trains this user–item model with feature factors by using the known ratings, after that the unknown ratings is estimated by calculating the inner products of corresponding user-factor and item-factor vectors pair [24]. Motivated by Netflix prize, matrix factorization (MF)-based techniques for latent factor (LF) analysis on sparse matrices arise and become the most successful approach for LFM . The existing literature on this topic embodies a variety of approaches, including the singular value decomposition (SVD) ++ model [25], probabilistic MF model [26], Nonnegative MF [27] (NMF) and nonparametric MF model [28]. In summary, these two kinds of models (i.e., NBM and LMF) can be employed in different occasions depending on detailed requirements.

Regarding for designing the healthy recommender system, most prior works focus on evaluating the one-step recommendation accuracy on the static dataset, which assumes that rating matrix is fixed without incremental data. For real-world online applications and systems, this assumption is lack of practicability since the mutual feedback between system and the user decisions are on-going [29]. Thus, the new ratings happen continuously in recommender systems, and the real-world sites with recommender system are temporally evolving systems.

Most importantly, it is one of the most important goals but not the ultimate goal that making good rating predictions for unrated items. The original intention of employing recommender system is that not only mining some useful information to customer, but also bringing profits to the sites [30-31]. Hence, the ultimate goal of employing the recommender system is to broaden the scope of customer interest, in addition to recommend unrated items with high predicted rating to user.

In summary, the quality metrics of a healthy recommender system should include accuracy, novelty, diversity and other utilities. Furthermore, it is significant to evaluate the quality of recommender system in a more comprehensive way that the combination of short- and long-term effects.

Unfortunately, the long-term effects of a CF-based recommender in the temporally evolving system are rarely discussed up to date. For example, it is still unclear whether a well-performed CF-based recommender in a one-step recommendation can also provide high performance in the long-term recommendation. Meanwhile, it is not sure that an accurate-oriented recommender (e.g., an LFM-based one) can also share the diversity of recommendation and the health of temporally evolving system. Last but not least, there are several unknown questions about: whether it will hurt the health of temporally evolving system by using highly accurate recommender; and which recommender will lead to this effect; what extent this effect will be during the temporal evolving process.

In such context, some works have been proposed to address the aforementioned issues. In order to deal with customers' preference for products which are keep drifting over time, Koren [29] proposed a time-aware factor model that introducing the temporal dynamics of user rating criteria rating and item's popularity. Based on this model, Koenigstein et al. [17] proposed a rich bias model with consideration of items' type to improve the recommendation accuracy in Yahoo! Music dataset.

Since the data of historical user behavior are ever growing in real-world applications, Luo et al. proposed an incremental collaborative filtering model to deal with the problem of data explosion [32]. For the heterogeneity environment, Rosaci and Sarnè proposed a new agent-based system with consideration of the effect of the device exploited by the user [33]. Shi et al. [34] designed a user preference-oriented recommender to deal with the temporal dynamics in the online system. However, these works only focus on improving the accuracy of recommender without considering other important performance indexes, such as diversity of recommendation and health of system. Different with above works, we give a comprehensive performance study of recommender in temporally evolving system.

Moreover, there are several studies along this line. Ekstrand et al. [35] proposed a novel MovieLens movie recommender that allows users to choose recommender and studies its performance in the long-term evolving process. Experimental results showed that how customers make use of this power. In order to evaluate the long-term effects of using recommender system, Zeng et al. [36] proposed an evolution model to simulate the interaction between recommender system and its users, but it not considered the temporal dynamics. Hu et al. [37] investigated the recommendation accuracy of different recommenders in evolving networks without considering the temporal dynamics. Zhao et al. [38] reported the performance of different recommender on the evolution of user–item bipartite network. However, the paper focuses mainly on studying the performance of entity relationships-based recommender instead of involving numerical optimization-based ones (e.g., LFM). In addition, the recommendation with time is iterated for a small number of rounds, making the long-term effects of co-evolution between recommender and users decisions hard to detect.

This paper focuses on investigating the long-term performance of CF-based recommenders in temporally evolving systems. To do so, we first model a recommender system with an evolving user–item bipartite network, and then propose a recommendation-based evolution method to simulate the temporal dynamics between recommender and the decisions of its users. Meanwhile, the network is evolving driven by recommender. In such settings, we study the performance of three well-known CF-based recommenders, i.e., U-NBM, I-NBM and LFM. For thoroughly evaluating the performance of involved models, we have adopted four performance metrics, i.e., Gini coefficient for ecosystem health, intra-similarity and popularity for recommendation diversity, and the root mean squared error (MASE) for recommender accuracy.

To summarize, the main motivation of this work is to explore the long-term performance of CF-based recommenders in the temporally evolving system, in order to make them more practical in real-world application. The main contributions include:

- We will propose a recommendation-based evolution method to capture the temporal dynamics between recommender system and users' behaviors in the temporally evolving system.
- We will investigate the ecosystem health and diversity of recommendation as well as recommender accuracy. Through empirical study, we find that numerical optimization-based LFM has the lowest accuracy loss during the temporal evolving process, while entity relationships-based ICF shows a relatively better performance than LFM in terms of recommendation diversity and ecosystem health.
- We will conduct experiment on two real datasets for studying the performance of CF-based recommenders with the proposed method.

The rest of this paper is organized as follows: Section 2 presents the preliminaries. Section 3 develops the proposed method. Section 4 illustrates the experiments and discusses the results. Finally, Section 5 concludes this paper.

## 2. Preliminaries

A CF-based recommender system generally models historical user behaviors into a user–item rating matrix defined as:

**Definition 1.** *The user–item rating matrix*. Formally, the user–item rating matrix R is the inner product of user set $U$ and item set $I$, which can be represent as $|U| \times |I|$. The element $r_{u,i}$ means the user $u$ rated on item $i$.

**Definition 2.** *The problem of CF*. Suppose that $R_K$ and $R_U$ are the known and unknown rating sets in $R$, then the problem of CF can be describe that constructing an estimator $\hat{R}$ to achieve or approximate $\arg\min(\sum_{(u,i)\in R_U} |\hat{r}_{u,i} - r_{u,i}|)$.

Entity relationships-based NBM models are classical popular kind of recommenders. Depending on which relationship is to be modeled, they can be further divided into U-NBM and I-NBM.

*U-NBM*: UCF is the classical implementation of U-NBM, and the basic idea of UCF is the common principle of *word-of-mouth*. The UCF algorithm follows these processes below. First, it calculates the similarities between the target user $u$ and the rest of users $v$ that have rated on item $i$. Then it sorts the users in a descend way, according to the calculated similarities, and picks the top K users as the K-nearest neighbors of target user $u$. Finally, for target user u, the predicting score of unrated item $i$ (i.e., $\hat{r}_{u,i}$) is calculated by:

$$\hat{r}_{u,i} = \frac{\sum_{v=1}^{K} w(u,v) r_{v,i}}{\sum_{v=1}^{K} |w(u,v)|} \tag{1}$$

where $K$ is the $u$'s neighborhood size and $w(u,v)$ denotes the similarity between user $u$ and $v$. Here we adopt the cosine-based similarity measure:

$$w(u,v) = \frac{\sum_{i\in R_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i\in R_u} r_{u,i}^2 \sum_{j\in R_v} r_{v,j}^2}} \tag{2}$$

where $R_{u,v}$ denotes the items set rated by both user $u$ and user $v$. $r_{u,i}$ denotes the rating of user u on item $i$.

*I-NBM*: ICF is the classical implementation of I-NBM. While UCF relies on the opinion of like-minded users to predict a rating, ICF focus on ratings given to similar items. Instead of considering users' similarities in UCF, ICF first calculates the similarity $w(i,j)$ between item $i$ and item $j$ for each item pair, then it gets the K-nearest neighbors of item $i$ according to the calculated similarity. Hence, for target user $u$ the predication score of an unrated item $i$ is calculated as:

$$\hat{r}_{u,i} = \frac{\sum_{j=1}^{K} w(i,j) r_{u,j}}{\sum_{j=1}^{K} |w(i,j)|} \tag{3}$$

Like UCF, the similarity equation can be expressed as:

$$w(i,j) = \frac{\sum_{u\in R_{i,j}} r_{u,i} r_{u,j}}{\sqrt{\sum_{u\in R_i} r_{u,i}^2 \sum_{u\in R_j} r_{u,j}^2}} \tag{4}$$

where $R_i$ is the set of users who have rated item $i$. $R_{i,j}$ denotes the set of users who have rated both item $i$ and item $j$.

*LFM*: Regarding to LF-based model, it is a classical type of numerical optimization method. First, it decomposes the rating matrix R into two low-dimension matrices (i.e., P and Q) with rank-$f$, where $P \in \mathbb{R}^{|u|\times f}$ and $Q \in \mathbb{R}^{|I|\times f}$, $f \le \min(|U|, |I|)$. Then the predicted rating of user $u$ rated on item $I$ can be defined as the inner product of the corresponding user–item feature vector pair given by:

$$\hat{r}_{u,i} = p_u^T q_i \tag{5}$$

Further study [39] indicates that in order to improve prediction accuracy, the user bias $b_u$, item bias $b_i$ and overall average rating $\mu$ can be added into (5):

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u^T q_i \tag{6}$$

In order to learn these parameters in (6), we use the stochastic gradient decent (SGD) algorithm to minimize the regularized squared error (RSE) on $R_K$:

$$\sum_{(u,i)\in R_K} e_{ui}^2 = \sum_{(u,i)\in R_K} \left(r_{ui} - \mu - b_u - b_i - p_u^T q_i\right) + \lambda\left(b_u^2 + b_i^2 + \left\|p_u^2\right\| + \left\|q_i^2\right\|\right) \tag{7}$$

where $\|\cdot\|$ is the Euclidean norm and $\lambda$ is the Tikhonov regularizing term for avoiding overfitting. In detail, the elements in $p_u$, $q_i$, $b_i$, and $b_u$ update through a training step using the following equations:

$$p_{uk} \leftarrow p_{uk} + \lambda(2e_{ui}q_{ik} - \beta p_{uk}) \tag{8}$$

$$q_{ik} \leftarrow q_{ik} + \lambda(2e_{ui}p_{uk} - \beta q_{ik}) \tag{9}$$

$$b_i \leftarrow b_i + \lambda(2e_{ui} - \gamma b_i) \tag{10}$$

$$b_u \leftarrow b_u + \lambda(2e_{ui} - \gamma b_u) \tag{11}$$

where $\beta$ and $\gamma$ are the regularization parameters.

## 3. Recommendation-based evolution method

For investigating the performance of recommenders on temporally evolving system, we first describe the recommender system as a user–item bipartite network. A simple user–item bipartite network is shown in Fig. 1. In detail, the circle represents the user node and the square represents the item node. The edge between circle and square means user has rated on item. Taking node $A$ for example, user node $A$ is connected with item node $a$, $b$, and $d$, which means user $A$ has rated on item $a$, $b$, and $d$, respectively.

For capturing the temporal dynamics in the dataset, different with dividing the dataset into training and test subset, we divide the dataset into a series of subsets by using the dataset partitioning method as shown in Fig. 2. Specifically, the edges between user and item are sorted in order of time first. Then the first 20% of dataset ($C_0$) constructs an initial user–item network. On one hand, as test set, the subsequent dataset is equally divided into $n$ subsets ($C_i$, $i = 1\ldots.\ n$) according to time. On the other hand, the edges in subsequent dataset will be added to the network in one macro-step by the recommender. After each macro-step, the new recommendation edges ($R_1 \ldots R_{n-1}$) are added into the original dataset ($C_0$). Hence, the new evolving network is constructed. Meanwhile,

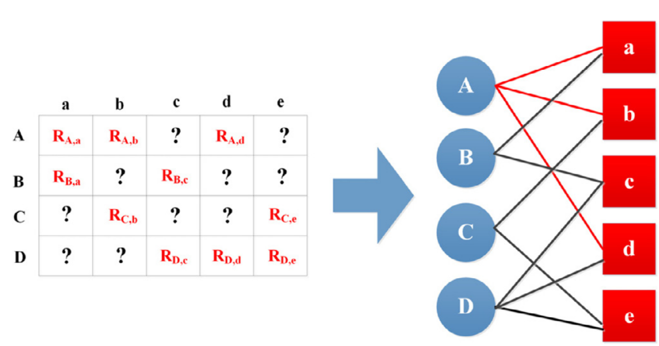



**Fig. 1.** A user–item bipartite network.

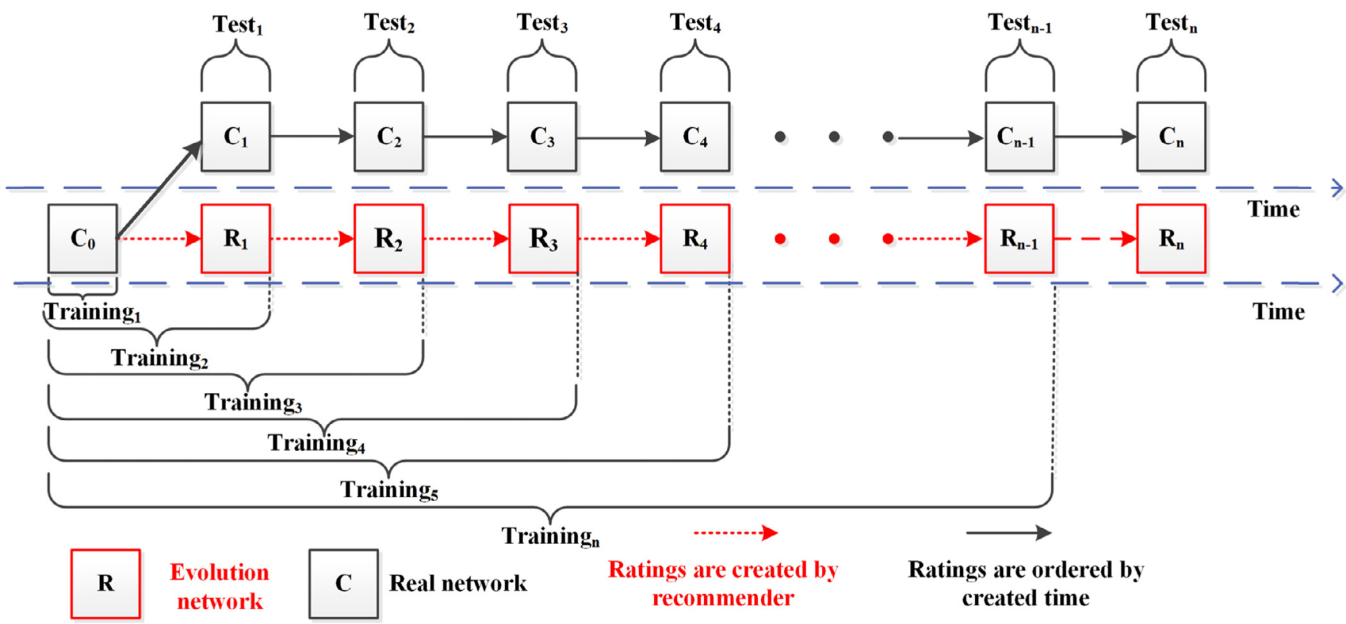



**Fig. 2.** The dataset partitioning.

the evolving network is considered as the new training set for next-step recommendation.

After that, we propose a recommendation-based evolution method to simulate the temporally evolving process of recommender system, due to the mutual feedback between recommender system and its user decisions. Obviously, the network is evolving driven by recommender. The number of added edges is determined by the degree increase of the real data ($C_i$) in the corresponding macro-step. For simplicity, we assume that all of users would accept recommendations (i.e., the items with highest prediction score), resulting in the network is evolving in a deterministic way. We perform realization of network evolution with each recommender.

Note that this proposed network evaluation method includes the temporal dynamics between the recommender system and its users, since both of the dataset partitioning and the evolving process are based on the real data with time information. In each macro-step, the user degree in the evolution network is the same as that in real network. However, due to the effect of recommendation, the item degree is different from the real network. As mentioned before, three state-of-the-art CF recommenders will be studied in this paper. For analyzing the long-term performance of recommender in temporally evolving network, we will compare the evolution networks to the real network. In this paper, we do not refer to the growth of the network size, since it introduces the cold start issue. The whole process of network evolution method is described in Fig. 3.

For the computational complexity of the proposed recommendation-based evolution methods, it depends on the used recommender and the parameter *Max_step* (i.e., *n*). For UCF, the computational complexity of one-step recommendation is $O(|I|^*(E_i/|I|)\hat{}2)$, where $E_i = \sum_{j=1}^{i} C_0 + R_j$ is the known ratings at the macro-step $i(i=1\ \ldots,\ n-1)$. Hence, the computational complexity of UCF-based evolution method is $O(Max\_step * |I| * (E_i/|I|)\hat{}2)$. For the ICF-based evolution method, the computational complexity of one-step recommendation and whole evolving process are $O(|U|^*(E_i/|U|)\hat{}2)$ and $O(Max_step * |U| * (E_i/|U|)\hat{}2)$respectively. For LFM, the computational complexity of one-step recommendation is $O(E_i{}^*f^*S)$, where $f$ is the number of latent factor, and $S$ is the total number of iteration for training data. Hence, the computational complexity of LFM-based evolution method $O(Max\_step * E_i * f * S)$.

## 4. Experiments and results

In this section, we first introduce some evaluation metrics and the details of experiment setting. Then we conduct some experiments to evaluate the proposed method and give the corresponding analyses.

### 4.1. Evaluation metrics

For evaluating the performance of a recommender system in temporally evolving network, we adopt several types of measures as follow:

*RMSE*: the accuracy of recommendation is one of the most important evaluation metrics for recommender. In this paper, we focus on evaluating the prediction accuracy of recommender in terms of ratings predictions. Root mean squared error (RMSE) is the most popular metrics used in evaluating accuracy of CF models [42]. Formally, the RMSE of a recommender is given by:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i\in T}(r_{u,i} - \hat{r}_{u,i})^2}{|T|}} \tag{12}$$

where $T$ denotes the validation dataset.

*Intra-similarity*: For a healthy recommender system, it should diversify the recommendations to each specified user, otherwise it is not meaningful to receiving many recommendations from same topic. In this paper we concern the diversity of recommendation list for each user, hence we adopt the intra-similarity of recommendation list to measure the diversity [42]. The intra-similarity of each user is defined as:

$$I_l = \frac{1}{L(L-1)}\sum_{i\neq j} w(i,\ j) \tag{13}$$

where $w(i,\ j)$ is the similarity between item $i$ and item $j$, as shown in (2) and (4).

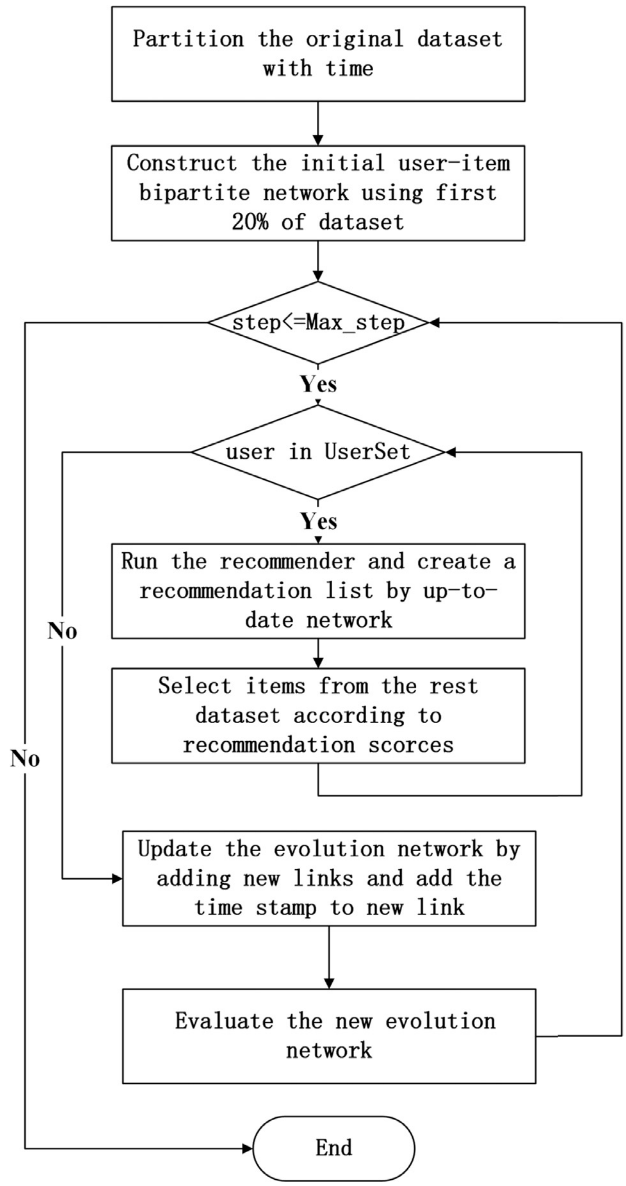



**Fig. 3.** Processing flow of the recommendation-based evolution method.

Thus the intra-similarity of the whole recommender system is described as:

$$I = \frac{1}{m}\sum_{l=1}^{m} I_l \tag{14}$$

The lower intra-similarity of the whole system has, the higher diversity of the system has.

*Popularity*: Under the condition of approximately same accuracy, it can bring additional sales that a recommendation list includes more novelty items, rather than that only includes high popular items. Hence, a healthy recommender system should suggest items with which the user is unfamiliar. A common method of measuring popularity is that calculating the average degree $\langle k\rangle$ over all recommendations.

$$\langle k\rangle = \frac{1}{nL}\sum_{u=1}^{n}\sum_{i\in R^u} k(i) \tag{15}$$

where $L$ is the length of recommendation list, $n$ is the number of users, and $k(i)$ is the number of users who collect the item $i$, respectively.

*Gini coefficient*: In our experiments, one of main corners is to evaluate the health state of system during the temporally evolving process. Hence, we measure the inequality of the item popularity distribution by adopting the Gini coefficient, which is widely used in economic and social fields [40,41]. It can be computed as:

$$G = \frac{2\sum_{i=1}^{M} iK_i}{M\sum_{i=1}^{M} K_i} - \frac{M+1}{M} \tag{16}$$

where $K_i$ is the popularity of items and has been sorted in the ascending order. The range of $G$ is [0, 1], where 0 corresponding to equal popularity of all items and 1 corresponding to zero popularity of all items but one, respectively. The smaller Gini coefficient is, the healthier the system is. $M$ is the number of items.

### *4.2. Datasets*

In this paper, two datasets with time information are used:

(1) D1(MovieLens): Collected through the MovieLens system (available at[2]), D1 contains 1000,209 ratings in the range of [1,5], by 6040 users on 3706 movies. Its rating density is 4.47%.
(2) D2(Netflix): It is a subset of the original dataset released from Netflix system (available at[3]). The subset contains 93,945 ratings in the scale of [1,5] from 2000 users on 4088 movies. Its rating density is 1.17% only.

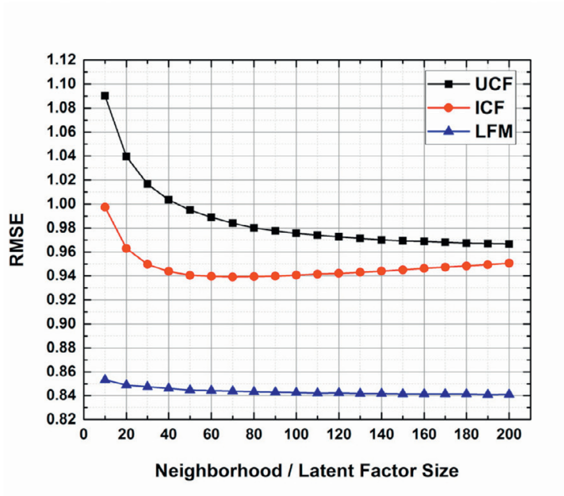
These datasets are often used in the recommender system community. According to the introduced dataset partitioning method, each dataset is ordered by ascending time. The initial network consists of the first 20% of the edges. After that, the subsequent edges are equally divided into 40 subsets according to time while $n = 40$.

### *4.3. Experiments with neighborhood and latent factor sizes*

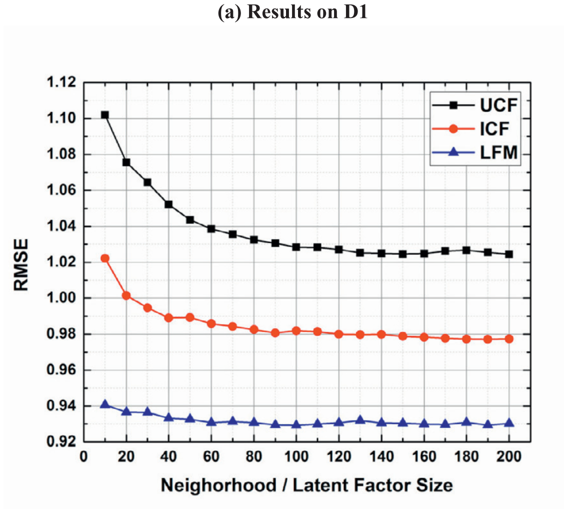
The size of the neighborhood ($K$) or latent factor ($f$) has significant impact on the prediction quality. For determining the sensitivity of these parameters and keep experiment results comparable across all recommenders, we performed a set of experiments where we varied the neighborhood size of NBM and the latent factor size of LFM to compute corresponding RMSE's, respectively. Our results in D1 and D2 are shown in Fig. 4. We can observe that the neighborhood size affects the prediction quality of NBM (i.e., ICF and UCF) both in D1 and D2. Specifically, different recommenders show different type of sensitivity.

In D1 as shown in Fig. 4(a), the curve of ICF decreases significantly as increasing the neighborhood size from 10 to 50. After that, the rate of decrease diminishes and the curve tends to be stable. Similarity with ICF, the curve of UCF experiences the same tendency, but the optimal neighborhood size of UCF is 100 (RMSE is 0.975). On the other hand, the performance of LFM is not sensitive to the latent factor size varies as shown in Fig. 4. Moreover, the value of RMSE keeps around 0.85 in D1, when latent factor size changes from 10 to 200. The same phenomenon also happened in D2. Therefore, compared to NBM (i.e., UCF and ICF), LFM has two obvious advantages: First, LFM can guarantee the prediction accuracy without increasing the latent factor size, that is the performance of LFM is not sensitive to the latent factor size. Second, compared to the NBM, the LFM with a well-designed loss function is more competitive in terms of accuracy.

[2] http://grouplens.org/datasets/movielens
[3] http://www.netflixprize.com

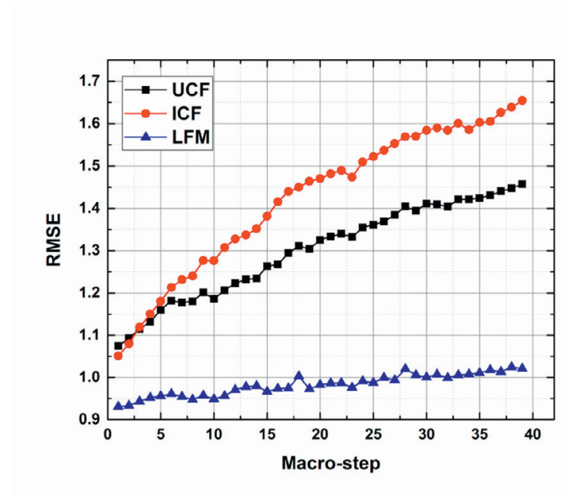



**(a) Results on D1**

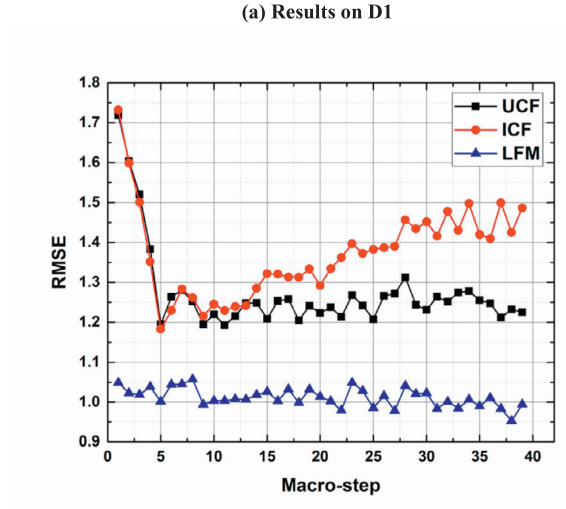



**(b) Results on D2**

**Fig. 4.** The RMSE comparison between all tested recommenders on each dataset with varying neighborhood size or latent factor size.




**(a) Results on D1**




**(b) Results on D2**

**Fig. 5.** The RMSE comparison between all tested recommenders during the evolution processing.

### *4.4. Prediction accuracy of recommenders on the evolving model*

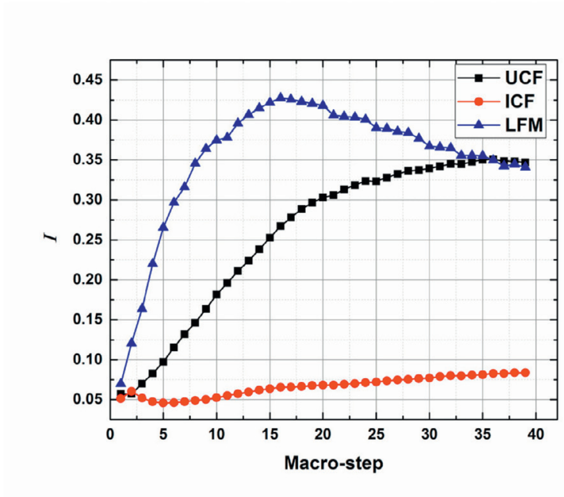
Prediction accuracy is by far the most discussed property in the recommender systems literature. Thus, it is necessary to measure accuracy index (i.e., RMSE) to observe the effect of temporally evolving network on the performance of each recommender. In order to get objective and unbiased results, we first set the optimal neighborhood size for NBM and late factor size for LFM as demonstrated in Fig. 4. That is $K = 100$ for UCF, $K = 40$ for ICF and $f = 20$ for LFM. After that, we run the evolution model with different recommenders. Note that we set the *Max_step* is 39 in our evolution model, according to the number of subsets (i.e., $n = 40$).

Fig. 5 shows the accuracy tendency of each recommender in terms of RMSE during the whole temporal evolving process. Regarding to D1, although there is an increase tendency for all of tested recommend in term of RMSE, LFM has the lowest accuracy loss among ICF and UCF during the whole evolving process. The reason is that performance of LFM is based on learning method and it establishes the optimal model by solving a carefully designed loss function, while the ICF and UCF are statistical based methods and do not have learning process. Regarding to D2, it shows that there is a sharp decrease tendency for the curves of ICF and UCF from beginning to 5th macro-step. Unfortunately, after that, both of them increase slowly along with fluctuation process. Compared to NBM (i.e., ICF and UCF), LFM still enjoys the lowest accuracy loss and the curve has a fluctuation process but the RMSE keeps stable around 1. The reason can be inferred that due to the feature of temporal dynamics in D2 (i.e., Netflix), it will lead to an accuracy loss when the customers' preferences on products are drifting during the evolving process in D2.

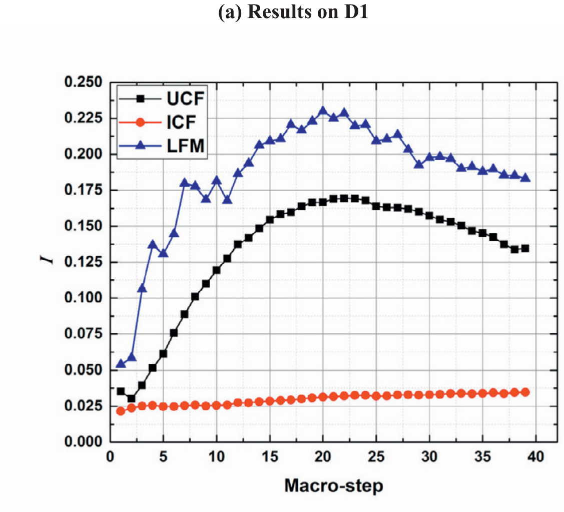
Overall, this experiment demonstrates that compared to the NBM (i.e., ICF and UCF), LFM can guarantee the long-term prediction accuracy in the temporal evolving online system. Moreover, it enjoys the high accuracy in one-step recommendation.

### *4.5. Diversity of recommendation on the temporally evolving network*

It is well known that under the condition of approximately same accuracy, a healthy recommender system should recommend more novelty items to each user, rather than only recommend the popular item with high score. Hence, for showing the impact of evolving network on the diversity of recommendation produces by different recommenders, we also measure the intra-similarity and popularity of recommendation during the temporal evolving process, respectively. Fig. 6 shows the evolution of intra-similarity produced by different recommenders. Clearly, ICF has the lowest intra-similarity of recommendation among all tested recommender.

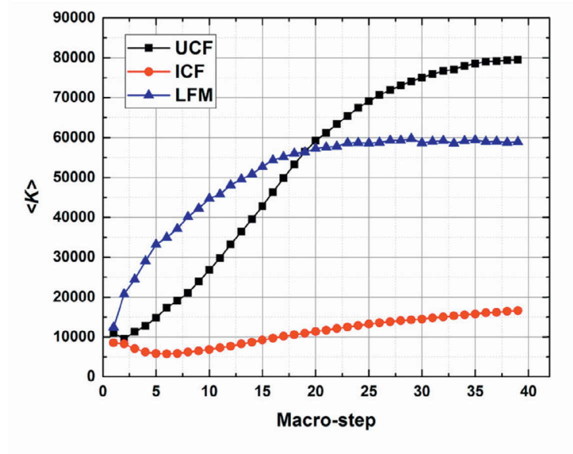



(a) Results on D1

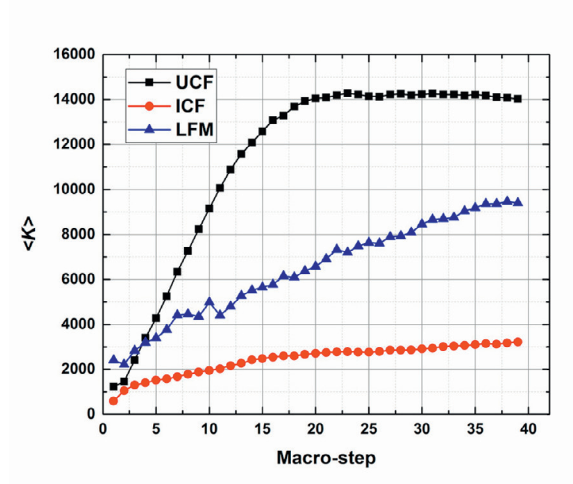



(b) Results on D2

**Fig. 6.** The intra similarity of recommendation produced by different recommenders during the evolving process.




(a) Results on D1




(b) Results on D2

**Fig. 7.** The popularity of recommendation produced by different recommenders during the evolving process.

The following is UCF while LFM has the highest intra-similarity of recommendation. Hence, it means that compared to LFM, the recommendation list produced by ICF yields good recommendation diversity.

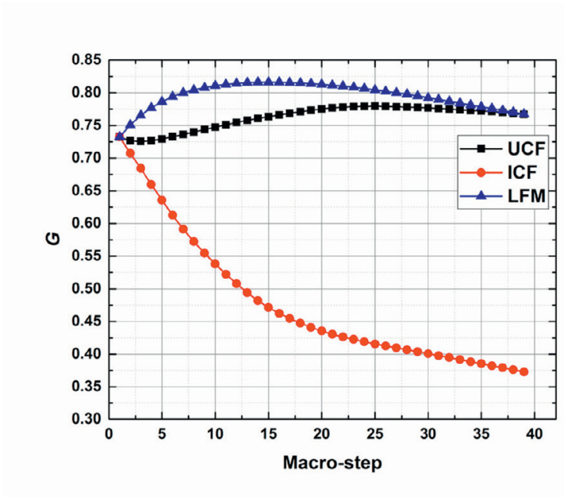
Fig. 7 shows the popularity of recommendations achieved by each recommender. It demonstrates that ICF again produced the most novel (i.e., less popular) recommendations. Clearly, the curves of LFM and UCF have an obvious increasing tendency during the evolving process, which indicate that both of them tend to give popular item with high prediction score. As a result, these recommendation items will be mostly popular items, especially for UCF. Hence, it demonstrates that recommendation list produced by ICF is more likely to recommend niche items, compared to UCF and LFM.

### *4.6. Distribution of item popularity on the temporally evolving network*

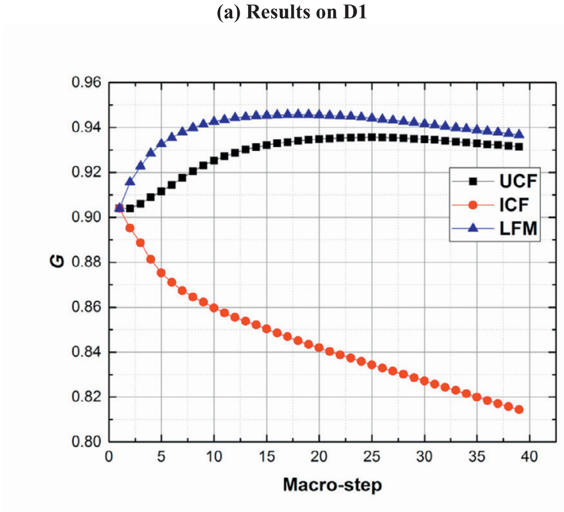
One main concern of this work is to observe the health state of ecosystem with recommender in the evolving network. Therefore, we use the network structure metrics such as Gini coefficient which is a common tool to study the topology state of a complex network.

For obtaining a quantitative comparison, the Gini coefficient is calculated after each macro-step of evolution. The higher the value is, the more uneven the distribution will be, as well as the greater harmful for ecosystem health. The change of Gini coefficient during the whole evolving process is shown in Fig. 8. Taking D1 for example, the curves of UCF and LFM increase gradually during the evolution process, which indicates that the distribution of item degree become more and more uneven during the temporal evolving process. Hence, it demonstrates that LFM and UCF essentially strengthen the long tail during the temporal evolving process. That leads to the loss of biodiversity in an online system. However, the Gini coefficient of ICF reduces significant during the whole evolving process, due to that it always recommends novelty items with relatively smaller degree. One of the reasons is that in UCF the prediction scores of unrated items are proportional to the degree of the items, while in ICF the prediction scores of unrated items have a strongly relationship with the method of items' similarity measure. The same phenomenon also happens in D2.

For UCF and LFM, although it is not a serious problem for one-step recommendation, it is harmful to the health of system when it happens recursively (e.g., recommendation in the temporal evolving system). As a result, the popular item account a larger part of the whole sales, and it is not meaningful for improve site's profits further. On the other hand, its recommendation accuracy is lower than LFM while ICF eliminates the long trail both in D1 and D2.




(a) Results on D1




(b) Results on D2

**Fig. 8.** The evolution of Gini coefficient using different recommenders during the temporally evolving process.

## 5. Conclusions

Collaborative filtering (CF) has proven to be effective and efficient in building recommender systems. However, the long-term performance of CF-based recommenders in a temporally evolving system remains unclear [21]. In this paper, we have investigated the long-term performance of several different CF-based recommenders on a temporally evolving online system. To do so, we first describe a recommender system as a user–item bipartite network, and then adopt a network evolution method to capture the mutual feedbacks between user's choices and recommenders in an evolving network. Then based on the proposed method, the performances of three state-of-the-art CF-based recommenders (i.e., U-NBM, I-NBM, LFM) in temporally evolving networks built on two industrial datasets were investigated. Finally, besides the frequently adopted RMSE which evaluates the prediction accuracy of involved recommenders, we have also adopted their intra-similarity, popularity and Gini coefficient to study their overall performance.

With such innovative evaluation settings, several important phenomena have been captured: 1). The prediction accuracy of all recommenders is decreasing in temporal evolving process if all users are entirely trust with the recommendation list produced by CF recommender, even though the recommender (i.e., LFM) has a well-accuracy in a one-step recommendation. 2). Although LFM proves to be highly accurate and scalable on static network, in evolving network, ICF has a relatively better performance in terms of recommendation diversity and ecosystem health. Furthermore, as we know, our work is the first one to investigate the long-term influence of different CF-based recommenders on the temporal evolving system. Our findings prove that it is necessary to adopt heterogeneous models in real systems to grasp various users' behavior patterns to improve their experiences.

## References


[1] N. Zeng, Z. Wang, Y. Li, M. Du, X. Liu, A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models, IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (2) (2012) 321–329.
[2] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[3] S. Li, M. Zhou, X. Luo, Z. You, Distributed Winner-take-all in Dynamic Networks, IEEE Trans. Autom. Control 62 (2) (2017) 577–589.
[4] L. Jin, S. Li, H.M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, doi: 10.1109/TIE.2017.2674624, In press.
[5] N. Zeng, Z. Wang, H. Zhang, F.E. Alsaadi, A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay, Cogn. Comput. 8 (2) (March 2016) 143–152.
[6] N. Zeng, Z. Wang, B. Zineddin, Y. Li, M. Du, L. Xiao, X. Liu, T. Young, Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach, IEEE Trans. Med. Imaging 33 (5) (May 2014) 1129–1136.
[7] M.N. Moreno, S Segrera, V.F. López, M.D. Muñoz, Á.L. Sánchez, Web mining based framework for solving usual problems in recommender systems. A case study for movies' recommendation, Neurocomputing 176 (February) (2016) 72–80.
[8] Y.-N. Xia, M.-C. Zhou, X. Luo, S.-C. Pang, Q.-S. Zhu, A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters, IEEE Trans. Syst. Man Cybern.: Syst. 45 (1) (January 2014) 1.
[9] J. Xuan, X. Luo, G. Zhang, J. Lu, Uncertainty analysis for the keyword system of web events, IEEE Trans. Syst. Man Cybern.: Syst. 46 (6) (June 2015) 829–842.
[10] N. Zeng, Z. Wang, Y. Li, M. Du, X. Liu, Inference of nonlinear state-space models for sandwich-type lateral flow immunoassay using extended Kalman filtering, IEEE. Trans. Biomed. Eng. 58 (7) (January 2011) 1959–1966.
[11] W. Zhou, J. Wen, Q. Xiong, M. Gao, J. Zeng, SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems, Neurocomputing 210 (October) (2016) 197–205.
[12] Y. Xun, J. Zhang, X. Qin, FiDoop: parallel mining of frequent itemsets using MapReduce, IEEE Trans. Syst. Man Cybern.: Syst. 46 (3) (March 2016) 313–325.
[13] Z. You, M. Zhou, X. Luo, S. Li., Highly efficient framework for predicting interactions between proteins, IEEE Trans. Cybern. 47 (3) (March 2017) 731–743.
[14] N. Zeng, Z. Wang, Y. Li, M. Du, J. Cao, X. Liu, Time series modeling of nano-gold immunochromatographic assay via expectation maximization algorithm, IEEE. Trans. Biomed. Eng. 60 (12) (December 2013) 3418–3424.
[15] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (3) (March 1997) 56–58.
[16] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (April 2005) 734–749.
[17] N. Koenigstein, G. Dror, Y. Koren, Yahoo! Music recommendations: modeling music ratings with temporal dynamics and item taxonomy, in: Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, Illinois, USA, 2011, pp. 165–172.
[18] J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications, in: R. Kohavi, F. Provost (Eds.), Applications of Data Mining to Electronic Commerce, Springer US, Boston, MA, 2001, pp. 115–153.
[19] J. Davidson, B. Liebald, J. Liu, P. Nandy, T.V. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, D. Sampath, The YouTube video recommendation system, in: Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 2010, pp. 293–296.
[20] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (1) (January 2003) 76–80.
[21] J.-M. Yang, K.F. Li, Recommendation based on rational inferences in collaborative filtering, Knowl. Based Syst. 22 (1) (January 2009) 105–114.
[22] J. Bobadilla, F. Serradilla, A. Hernando, Collaborative filtering adapted to recommender systems of E-learning, Knowl. Based Syst. 22 (4) (May 2009) 261–265.
[23] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Comput. 42 (8) (August 2009) 30–37.
[24] X. Luo, M. Zhou, M. Shang, S. Li, Y. Xia, A novel approach to extracting non-negative latent factors from non-negative big sparse matrices, IEEE Access 4 (April) (2016) 2649–2655.
[25] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, USA, 2007, pp. 5–8.

[26] R.R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, Advances in Neural Information Processing Systems 2011, New York, USA, 2011, pp. 1–8.
[27] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, Q. Zhu, A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method, IEEE Trans. Neural Networks Learn. Syst. 27 (3) (March 2016) 579–592.
[28] K. Yu, S. Zhu, J. Lafferty, Y. Gong, Fast nonparametric matrix factorization for large-scale collaborative filtering, in: Proceedings of the 32nd ACM SIGIR International Conference on Research and Development in Information Retrieval, Boston, MA, USA, 2009, pp. 211–218.
[29] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (April 2010) 89–97.
[30] J.D. Ekstrand, M.D Ekstrand, First do no harm: Considering and minimizing harm in recommender systems designed for engendering health, in: Engendering Health Workshop at the RecSys 2016 Conference, Boston, Massachusetts, USA, ACM, 2016, pp. 1–2.
[31] D. Jannach, P. Resnick, A. Tuzhilin, M. Zanker, Recommender systems: beyond matrix completion, Commun. ACM 59 (11) (October 2016) 94–102.
[32] X. Luo, M. Zhou, Y. Xia, Q. Zhu, An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering, IEEE Trans. Autom. Sci. Eng. 13 (1) (January 2016) 333–343.
[33] D. Rosaci, G.M.L. Sarnè, Recommending multimedia web services in a multi--device environment, Inf. Syst. 38 (2) (April 2013) 198–212.
[34] X. Shi, M.-S. Shang, X. Luo, A. Khushnood, J. Li, Long-term effects of user preference-oriented recommendation method on the evolution of online system, Physica A 467 (February) (Feb. 2017) 490–498.
[35] M.D. Ekstrand, D. Kluver, F.M. Harper, J.A. Konstan, Letting users choose recommender algorithms: an experimental study, in: Proceedings of the Ninth ACM Conference on Recommender Systems, Vienna, Austria, 2015, pp. 11–18.
[36] A. Zeng, C.H. Yeung, M. Medo, Y.-C. Zhang, Modeling mutual feedback between users and recommender systems, J. Stat. Mech: Theory Exp. 2015 (7) (June 2015) P07020.
[37] X. Hu, A. Zeng, M.-S. Shang, Recommendation in evolving online networks, Eur. Phys. J. B 89 (46) (February 2016) 1–7.
[38] D.-D. Zhao, A. Zeng, M.-S. Shang, J. Gao, Long-term effects of recommendation on the evolution of online systems, Chin. Phys. Lett. 30 (11) (November 2013) 118901.
[39] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, Q. Zhu, H. Leung, An efficient second-order approach to factorizing sparse matrices in recommender systems, IEEE Trans. Ind. Informatics 11 (4) (August 2015) 946–956.
[40] W. Halffman, L. Leydesdorff, Is inequality among universities increasing? Gini coefficients and the elusive rise of elite universities, Minerva 48 (1) (March 2010) 55–72.
[41] L. Wittebolle, M. Marzorati, L. Clement, A. Balloi, D. Daffonchio, K. Heylen, P. De Vos, W. Verstraete, N. Boon, Initial community evenness favors functionality under selective stress, Nature 458 (7238) (April 2009) 623–626.
[42] X. Luo, M. Shang, S. Li, Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications, in: Proceedings of the 16th IEEE International Conference on Data Mining (ICDM), Barcelona, Spain, 2016, pp. 311–319.

**Xiaoyu Shi** received the B.S. degree in computer science from the PLA Information Engineering University, Zhengzhou, China, in 2007, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2015. His research interests include recommender system, cloud computing, green computing and big data applications.

**Xin Luo** (M'14) received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2016, and is currently a professor of computer science and engineering. His research interests include big data analysis and intelligent control. He has published 60+ papers (including 10+ IEEE TRANSACTIONS papers) in his related areas.

**Mingsheng Shang** received his B.E. degree in Management in Sichuan Normal University in Chengdu, China in 1995, and Ph.D. degree in Computer Science from University of Electronic Science and Technology of China in Chengdu, China in 2007. He is currently a professor at the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China. His research interests are in complex network analysis and big data applications.

**Liang Gu** received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2011. He worked as postdoctoral researcher at Yale University from 2011 to 2013 and then was promoted as the associate research scientist from 2013 to 2015. He is currently the Chief Technology Officer of R&D at Sangfor Technology Inc. and adjunct research professor at Jinan University. His research interests include network security and cloud computing.