This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

A Survey of Collaborative Filtering Based Recommender Systems for Mobile Internet Applications

Zhe Yang, Bing Wu, Kan Zheng, *Senior Member, IEEE,* Xianbin Wang, *Senior Member, IEEE,* and Lei Lei, *Member, IEEE*

***Abstract*—With the rapid development and application of the mobile Internet, huge amounts of user data are generated and collected every day. How to take full advantages of these ubiquitous data is becoming the essential aspect of a recommender system. Collaborative filtering (CF) has been widely studied and utilized to predict the interests of mobile users and to make proper recommendations. In this paper, we first propose a framework of CF recommender system based on various user data including user ratings and user behaviors. Key features of these two kinds of data are discussed. Moreover, several typical CF algorithms are classified as memory-based approaches and model-based approaches and compared. Two case studies are presented in an effort to validate the proposed framework.**

***Index Terms*—Mobile Internet, Recommender System, Collaborative Filtering.**

I. Introduction

Along with the rapid development of mobile Internet and cloud computing, massive amounts of data are produced every day by both people and machines. Our society has already entered the era of Big Data [1]. Thanks to the various smart devices and mobile applications, Internet users can acquire all sorts of information about education, shopping, social activity, etc. [2] [3] [4] [5]. However, as the volume of data increases, individuals have to face the problem of excessive information, which makes it more difficult to make the right decisions. This phenomenon is known as information overload [6]. Moreover, limited by the input ability of mobile devices, users are usually unwilling to type in lots of words to describe what they want. Recommender system can alleviate these problems by effectively finding users' potential requirements and selecting desirable items from a huge amount of candidate information. Recommender systems are usually classified into two categories, i.e., content-based and collaborative filtering (CF) [7].

Content-based recommender system utilizes the contents of items and finds the similarities among them. After analyzing sufficient numbers of items that one user has already shown favor to, the user interests profile is established. Then the recommender system could search the database and choose proper items according to this profile. The difficulty of these algorithms lies in how to find user preferences based on the contents of items. Many approaches have been developed to solve this problem in the areas of data mining or machine learning. For example, in order to recommend some articles to a specific reader, a recommender system firstly obtains all the books this reader has already read and then analyzes their contents. Key words can be extracted from the text with the help of text mining methods, such as the well-known TF-IDF [8]. After integrating all the key words with their respective weights, a book can be represented by a multi-dimensional vector. Specific clustering algorithms can be implemented to find the centers of these vectors which represent the interests of this reader.

On the other hand, collaborative filtering (CF) has become one of the most influential recommendation algorithms [9]. Unlike the content-based approaches, CF only relies on the item ratings from each user. It is based on the assumption that users who have rated the same items with similar ratings are likely to have similar preferences. CF is specifically designed to provide recommendations when detailed information about the users and items is inaccessible. Furthermore, it successfully mitigates the problem of over-specialization [10], which is quite common in content-based systems. Over-specialization is the phenomenon that recommended items are always much the same and the diversity of recommendations is neglected. As CF makes recommendations according to the neighborhood (people with similar preferences), the item one user has consumed may be something new to his neighbors. The above features are particularly attractive which make CF algorithms extensively employed in recommender systems.

However, to the best of our knowledge, very few studies have revealed the common features of the various CF algorithms for mobile Internet applications. In addition, most of the existing surveys merely introduce the principles of CF algorithms, ignoring the importance of case study, which can demonstrate the performances of typical algorithms visually and specifically. Therefore, this paper focuses on collaborative filtering based recommender systems for mobile Internet applications. In particular, main contributions of this paper are highlighted as follows:

- We introduce a general framework of CF recommender system. This framework assists recommender developers

Manuscript received April 22, 2016; revised May 23, 2016.

Zhe Yang, Bing Wu, and Kan Zheng are with the Intelligent Computing and Communication (IC^2) Lab, Key Lab of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications (BUPT), Beijing, 100876, China (E-mail: zkan@bupt.edu.cn).

Xianbin Wang is with the Department of Electrical and Computer Engineering, The University of Western Ontario, N6A 5B9, London, Canada.

Lei Lei is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

to utilize the gathered data and to generate proper recommendations. The features of data collected from both user behaviors and user ratings are also discussed and compared.

- CF algorithms are classified. Main procedures of CF are briefly summarized and introduced.
- Two case studies are presented to validate the proposed framework. Evaluations on representative CF algorithms are conducted based on real-world datasets with detailed analysis and comparison.

The rest of this paper is organized as follows. Section II presents the framework of CF. Both classification and main procedures of typical CF algorithms are introduced in Section III. In Section IV, we conduct two case studies based on real-world datasets in order to analyze the performances of CF algorithms. Finally, Section V concludes this paper.

II. Framework of CF Recommender System

As illustrated in Fig. 1, the framework of a typical CF recommender system includes: 1) Data Collection; 2) Pre-processing; 3) Collaborative Filtering. Firstly, user data are collected through wireless networks and stored in the cloud database [11]. Then certain pre-processing operations are imperative for ensuring the data integrity and reliability. Based on these data, CF algorithms are implemented to predict user interests and recommend related items in order to save the time and effort.

A. Data Collection

Data collection is the fundamental of the entire recommender system. The gathered data mainly fall into four categories: demographic data, production data, user behavior and user rating [12].

1) Demographic Data: Many businesses require users to register on their servers and fill in personal information before using the services. The personal information usually includes name, telephone, gender, hobbies, etc. Based on the analysis of the above demographic data, businesses can establish the user profiles and push promotional messages to the mobile clients more specifically.

2) Production Data: Merchants tend to classify their commodities according to their functions, brands, prices, etc. For example, a video website usually adds tags to their videos in order to help the consumers to find what they enjoy more conveniently. Hence, the production data are easy to access by the businesses.

3) User Behavior: While browsing a website or listening to a piece of music, users are likely to be monitored by the server which stores a large amount of behavior data, such as the playing duration of a song, the purchasing date of a book, or even the number of clicks on a webpage. These data are usually of large volumes and need to be analyzed by specific data mining methods.

4) User rating: Some websites provide rating systems and guide consumers to rate items that they have experienced, such as movies, songs and web services. These ratings reflect the preferences of a consumer and receive increasing attention from the businesses. Furthermore, items may have various attributes which need to be rated respectively. Accordingly some rating systems provide users the opportunity to rate items based on multiple criteria which can greatly enrich the rating information [13].

All the data mentioned above can play an important role in the recommender system if being effectively utilized. However, as explained in Section I, collaborative filtering does not need any information about the users (demographic data) and items (production data), it focuses on the user's feedback including the explicit feedback (user rating) and the implicit feedback (user behavior) [14]. Key features of these two kinds of data are summarized in Table I.

B. Pre-processing

With the development of mobile Internet, the collected data are usually in various formats due to the diversity of user equipments and the heterogeneity of networks [16]. Therefore, data pre-processing has become an indispensable part of recommender systems, which is responsible for ensuring the input data of collaborative filtering to be completed and reliable. Pre-processing is usually divided into the following three steps.

1) Data Cleaning: Raw data cannot be directly utilized due to the presence of dirty data which may be produced by possible equipment failures or transmission errors. The error ratio may become very high especially when users are in a high speed [17] [18]. Besides, some consumers may rate the items arbitrarily, such as giving all items the highest rating for saving time, which is likely to reduce the reliability of the rating information on the whole. Specific outlier detection algorithms can alleviate these problems to some extent [15]. For example, after choosing part of the ratings as training data and establishing a classifier model based on machine leaning algorithms, the outliers can be removed with satisfying accuracy.

2) Generation of Implicit Ratings: Most CF recommender systems merely treat explicit user ratings as valuable information. However, a large portion of users do not always rate the items they have already consumed, which leads to the problem of data sparsity [19]. Thanks to the extensively applied mobile clients, specific user behaviors are collected and stored in the cloud with tremendous potential value, which may become the key to mitigate this problem. For instance, recommender systems collect massive volumes of user ratings and behaviors as training set and then implement specific machine learning algorithms on it, such as neural network or decision tree, so as to construct a prediction model which can transform user behaviors into implicit ratings, as illustrated in Fig. 2. The volume of rating data can be increased greatly by this means.

3) Data Integration: Both the explicit and implicit rating data are integrated into a matrix, namely, the rating matrix, as shown in Fig. 3. Obviously, there are still a plenty of missing elements in this matrix which need to be filled in through collaborative filtering.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

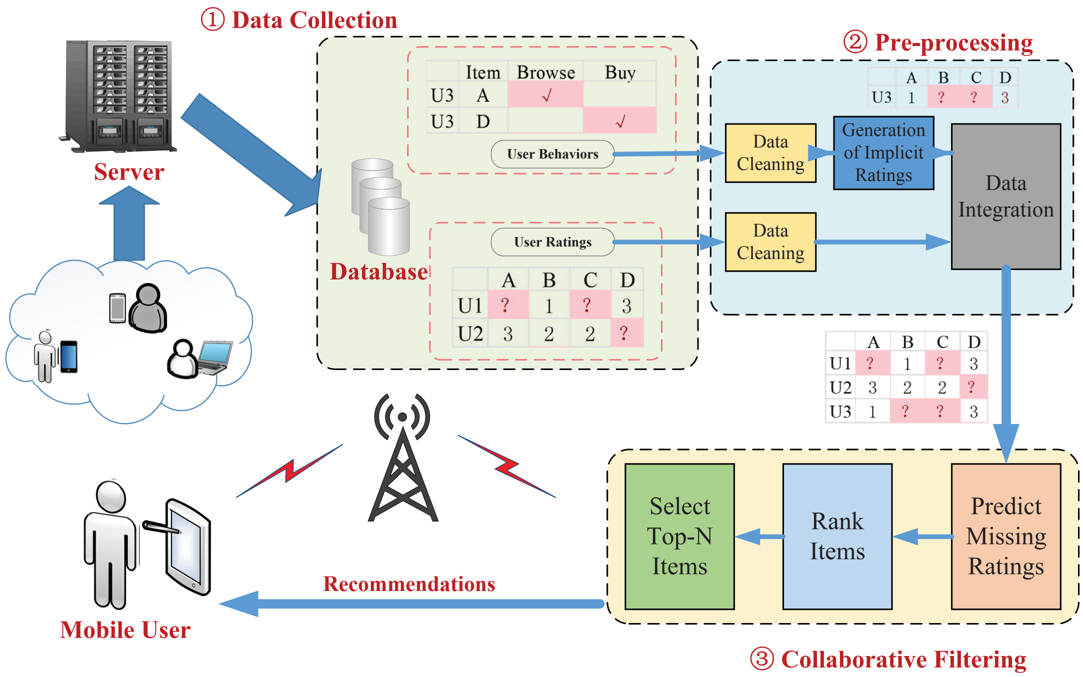


Fig. 1. Framework of CF recommender system.

TABLE I
KEY FEATURES OF THE USER RATING AND USER BEHAVIOR.

Feature	User Behavior	User Rating
Data size	Large	Small
(Non)-structured	Mainly semi-structured or non-structured, usually stored in the log files.	Structured data, which can be easily represented by a matrix.
Coverage	Wide, nearly all the users are recorded.	Narrow, only a part of users have the habit of rating an item after using it.
Objective/Subjective	Objective	Subjective
Easy/Difficult to be utilized	Difficult, certain algorithms are needed to explore the potential value.	Easy, data can be directly input into CF recommender systems.
Reliability	Instable due to the data mining algorithms and the amount of training data.	High, which reflects the user's preference on a certain item.

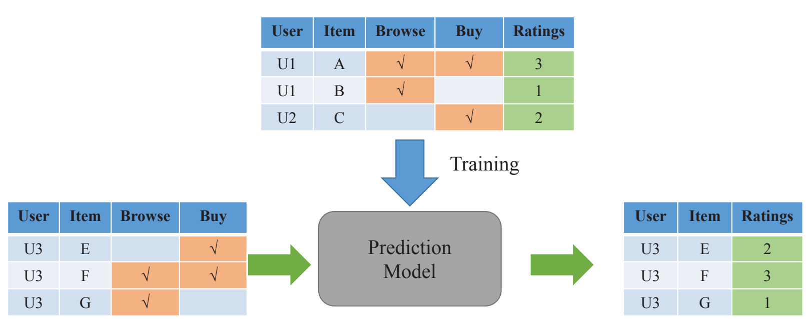


Fig. 2. Generation of implicit ratings based on user behaviors.

C. Metrics of Collaborative Filtering

General procedures of CF include predicting missing values, ranking items and selecting Top-N items. Since the rating matrix is incomplete, the main task of collaborative filtering is to predict these missing elements based on the known data. After that, items are ranked according to predicted ratings and Top-N of them are selected as the recommendations. Once a recommender system is established, another challenge is how to evaluate its performance. The metrics of recommender systems are divided into three categories [20], i.e.

1) Predictive Accuracy Metrics: In order to estimate the accuracy of the prediction, the complete dataset is divided

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

	A	B	C	D
U1	?	1	?	3
U2	3	2	2	?
U3	2	?	1	3
U4	4	4	2	?

Fig. 3. A rating matrix with missing values.

into a training set and a test set. The training set is used to generate predictions while the test set is responsible for evaluating the predictive accuracy. Two predictive accuracy metrics are extensively applied in collaborative filtering, i.e., Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which is defined by Eq. (1) and Eq. (2).

$$MAE = \frac{\sum_{(u,i)\in R_{test}} |R_{u,i} - R'_{u,i}|}{|R_{test}|}, \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} |R_{u,i} - R'_{u,i}|}{|R_{test}|}}, \tag{2}$$

where $|R_{test}|$ represents the number of ratings in test set. $R_{u,i}$ is the predicted rating for user u on item i and $R'_{u,i}$ is the actual rating in test set. A lower MAE or RMSE represents a higher predictive accuracy.

2) Classification Accuracy Metrics: Even though the above metrics are effective and easy to understand, users may not care about the exact figure of the predicted rating. What they only concerned is whether the recommendations are relevant with their interests or not. Under this circumstance, a feasible solution is to transform the ratings into the binary scale by choosing threshold properly. For example, if the ratings range from 1 to 5, items rated greater than 4 can be regarded as relevant and the others as irrelevant. In this way, both the recommendation list and the test set are divided into two parts. Three classification accuracy metrics are widely used to evaluate the relevance between recommendations and user interests, i.e., Recall, Precision and F-1 score [22].

Recall is defined as the proportion between the number of relevant items in the recommendation list and that in the test set.

$$recall = \frac{\sum_u L(N,u)}{\sum_u L(u)}, \tag{3}$$

where $L(N,u)$ and $L(u)$ represent the relevant items for user u in recommendation list and test set respectively. The upper limit of recall is 1, which means all the relevant items in the test set can be found in the recommendation list.

Precision is defined as the percentage of relevant items in the recommendation list.

$$precision = \frac{\sum_u L(N,u)}{UN}, \tag{4}$$

where U is the number of users. The upper limit of precision is 1, which means all the items in recommendation list are relevant.

Sometimes these two metrics may conflict with each other. For example, if the system presents all the items in the dataset to users as recommendations, the precision may be very low while the recall reaches up to 1. However, if the system only recommends one item to user and this item is exactly relevant, then the precision is 1 while the recall is unlikely to be high. In order to synthesize the two metrics, F1-score is proposed which is defined as the harmonic mean between recall and precision [24].

$$F_1 = \frac{2 * recall * precision}{recall + precision}. \tag{5}$$

3) Diversity Metrics: Recently, researchers become increasingly aware that simply pursuing the increase of predictive accuracy may reduce the diversity of recommendations [27]. Thus diversity metrics need to be taken into account while generating recommendations and they can be divided into two parts, i.e., intra-list similarity and inter-list similarity [25]. Intra-list similarity S_u^i measures the similarity between each item in the recommendation list for user u, which is computed as

$$S_{\text{intra}}^u = \frac{2}{n(n-1)} \sum_{i,j\in L_u, i\neq j} Sim(i,j), \tag{6}$$

where L_u is the recommendation list for user u. n is the number of recommendations. $Sim(i,j)$ represents the similarity between item i and j. The details of similarity calculation are provided in the next section.

Inter-list similarity is used to measure the similarity between the recommendations of user u and other users, which is defined as

$$S_{\text{inter}} = \frac{2}{n(n-1)} \sum_{u,v\in U, u\neq v} \frac{|L_u \cap L_v|}{|L_u|}, \tag{7}$$

where n is the number of recommendations for user u. L_u and L_v are the recommendation lists for user u and v. It is evident that a higher intra-list or inter-list similarity indicates a lower diversity of the recommendation list.

III. TYPICAL ALGORITHMS OF CF

As illustrated in Fig. 4, CF algorithms can be roughly classified into two categories, i.e., memory-based CF and model-based CF [28]. Memory-based CF algorithms directly utilize volumes of historical data to predict rating on target item and provide recommendations for the active user. Whenever a recommendation task is performed, the system needs to load all the data into the memory and implement specific algorithms on them. Differently, model-based CF can utilize certain data mining methods to establish a prediction model based on the known data. Once the model is obtained, it does not need the raw data any more while recommending.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

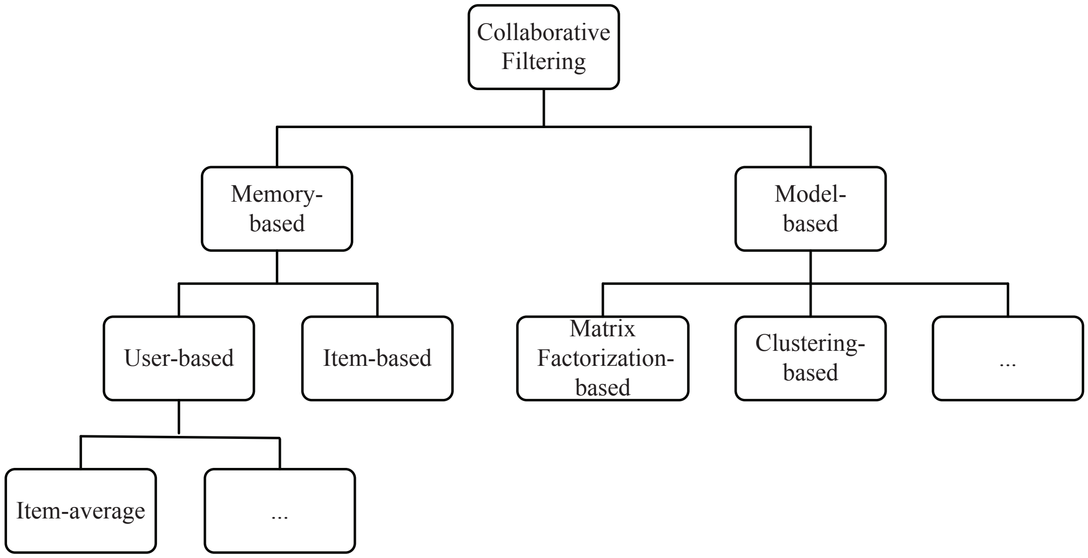


Fig. 4. Classification of collaborative filtering algorithms.

A. Memory-based CF

Memory-based CF is commonly utilized in recommender systems owning to its high-effectiveness and easy-implementation. The performance of memory-based CF is usually satisfactory on both accuracy and diversity. Based on all the ratings in database, the recommender system finds neighbors for certain user or item and calculates the predicted value for the unknown rating.

1) Classification: Memory-based CF algorithms can be divided into two kinds: user-based CF and item-based CF [23]. User-based CF explores the relationship between rows in the rating matrix while item-based CF focuses on the relationship between columns.

a) User-based CF: User-based CF firstly calculates the similarities between the active user and the other users. Common similarity measures in CF include pure cosine, adjusted cosine and Pearson correlation coefficient. Users with high similarities are selected as neighbors of the active user. Then the system utilizes the neighbors' ratings on a specific item to calculate the weighted average which is regarded as the predicted rating, treating the respective similarities as weights. At last, the recommender ranks all the items according to their predicted ratings and selects Top-N items as recommendations. One challenge the user-based CF has to face is the scalability problem. For some famous video websites with millions of registered users, calculating the similarities among all users and choosing neighbors in real-time is difficult to implement. As a result, user-based CF is more suitable when users are not so many and the user group is relatively stable.

Item-average algorithm is a special case of user-based CF, which chooses all the users as neighbors with equal weights, as illustrated in Fig. 5.

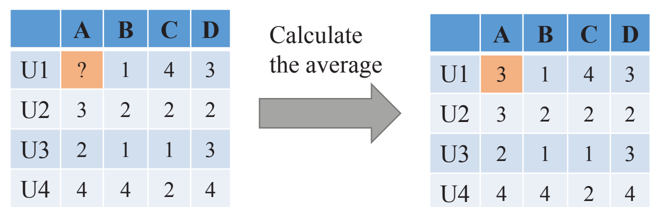


Fig. 5. Rating prediction using item-average algorithm.

b) Item-based CF: Unlike the user-based CF, item-based CF focuses on the similarities among items. It is based on the assumption that items with similar user ratings are likely to be of similar types. Hence the similarities among items are firstly calculated using the same similarity measures with the user-based CF. After choosing the neighbors for the target item and calculating the weighted average, the predicted rating on this item is obtained. It is easy to understand that once the items are too many and change frequently, scalability problem is also difficult to avoid. A comparison between the user-based CF and item-based CF is shown in Table II.

2) General Procedures: Fig. 6 gives five general procedures of memory-based CF which need to be analyzed in detail as follows:

- **Step 1**: Similarity Computation
  Similarity computation between users or items is a critical step in CF. Lots of similarity measures are extensively used in memory-based CF, e.g.
  a) Basic Similarity Measures:
  - Pure Cosine Similarity:
    Pure cosine similarity measures the cosine value of the angle between two vectors. For the user-based CF, users are represented by the rows of the rating matrix with the missing values set to 0. Pure cosine similarity between user vectors is calculated as follows,

$$\omega_{ij}^{u} = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}. \tag{8}$$

    where i and j indicate two user vectors. For the item-based CF, items are represented by the columns of

TABLE II
COMPARISON BETWEEN USER-BASED CF AND ITEM-BASED CF.

Categories of CF	Similarity	Application scenarios
User-based CF	User-user similarity	• The number of items is larger than that of users. • Users do not change frequently.
Item-based CF	Item-item similarity	• The number of users is larger than that of items. • Items do not change frequently.

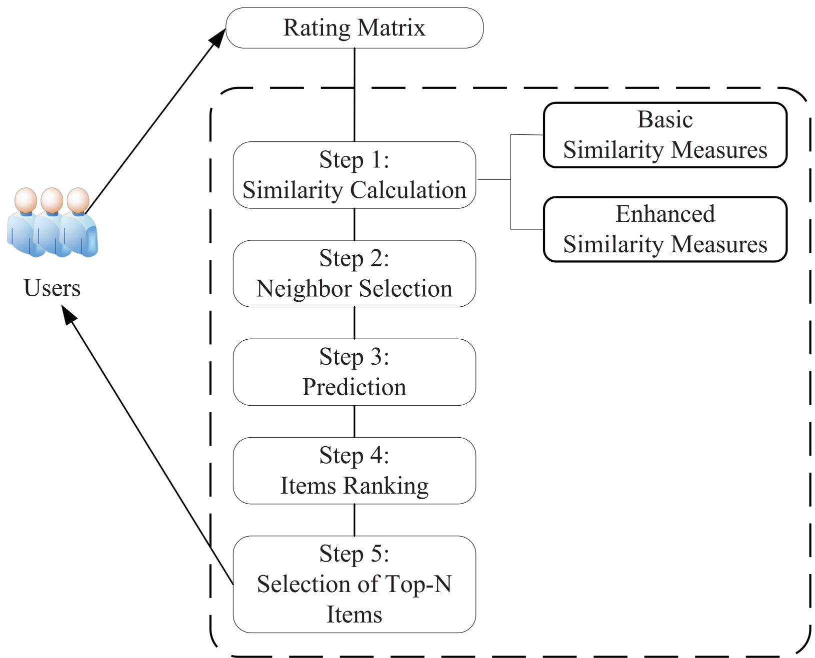


Fig. 6. Main procedures of memory-based CF.

the rating matrix and the similarity between them is defined as,

$$\omega_{kl}^{i} = cos(\vec{k}, \vec{l}) = \frac{\vec{k} \cdot \vec{l}}{||\vec{k}||_2 * ||\vec{l}||_2}. \quad (9)$$

where k and l represent two item vectors.

– Adjusted Cosine Similarity:
As rating scales among users are different, same rating of two users does not mean the same degree of interest. This problem is ignored by pure cosine similarity. Moreover, setting the missing ratings to 0 by default is more or less unreasonable. Adjusted cosine similarity corrects these defects by subtracting the average rating of the user and using co-rated items to establish the vector. Co-rated items are the items rated by both user i and user j. The adjusted cosine similarity for user-based CF is defined as follows,

$$\omega_{ij}^{u} = \frac{\sum_{k\in K}(r_{i,k} - \bar{r_i})(r_{j,k} - \bar{r_j})}{\sqrt{\sum_{k\in K}(r_{i,k} - \bar{r_i})^2}\sqrt{\sum_{k\in K}(r_{j,k} - \bar{r_j})^2}}, \quad (10)$$

where K is the set of the co-rated items. $r_{i,k}$ and $r_{j,k}$ are two ratings on item k from user i and j. $\bar{r}_i$ and $\bar{r}_j$ are the average ratings of user i and user j. For the item-based algorithm, co-rating users are chosen to calculate the similarity between items, which is shown in Eq. (11).

$$\omega_{kl}^{i} = \frac{\sum_{u\in U}(r_{u,k} - \bar{r_u})(r_{u,l} - \bar{r_u})}{\sqrt{\sum_{u\in U}(r_{u,k} - \bar{r_u})^2}\sqrt{\sum_{u\in U}(r_{u,l} - \bar{r_u})^2}}, \quad (11)$$

where U is the set of users who have rated both item k and item l. $r_{u,k}$ and $r_{u,l}$ are two ratings from user u on item k and l. $\bar{r}_u$ is the average rating of user u.

– Pearson correlation coefficient:
Pearson correlation coefficient (PCC) reflects the degree of linear correlation between two variables. Same as the adjusted cosine, PCC selects co-rated items or users to calculate similarities. Pearson correlation coefficients between two users and two items are respectively given by Eq. (12) and Eq. (13).

$$\omega_{ij}^{u} = \frac{\sum_{k\in K}(r_{i,k} - \bar{r_i})(r_{j,k} - \bar{r_j})}{\sqrt{\sum_{k\in K}(r_{i,k} - \bar{r_i})^2}\sqrt{\sum_{k\in K}(r_{j,k} - \bar{r_j})^2}}, \quad (12)$$

$$\omega_{kl}^{i} = \frac{\sum_{u\in U}(r_{u,k} - \bar{r_k})(r_{u,l} - \bar{r_l})}{\sqrt{\sum_{u\in U}(r_{u,k} - \bar{r_k})^2}\sqrt{\sum_{u\in U}(r_{u,l} - \bar{r_l})^2}}. \quad (13)$$

It can be observed that for user-based CF, PCC is the same as adjusted cosine, while for item-based CF,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

they are a little different. The subtrahends of ratings are the average of users and items respectively.

b) Enhanced Similarity Measures:

– Set-based Similarity:
Similarity is likely to be overestimated using adjusted cosine or PCC when the number of co-rated items is very small. A set-based similarity is proposed to alleviate this problem [49]. For user-based CF, the enhanced similarity between user i and user j is defined as follows,

$$\omega'^{u}_{ij} = \frac{2|K_i \cap K_j|}{|K_i| + |K_j|}\omega^{u}_{ij}, \tag{14}$$

where ω^{u}_{ij} is the basic similarity such as PCC. $|K_i|$ and $|K_j|$ are the numbers of items rated by user i and user j, respectively. $|K_i \cap K_j|$ indicates the number of co-rated items of user i and user j. If the co-rated items are very few, the basic similarity will be reduced greatly by the decay factor. Likewise, the enhanced similarity between item k and item l is defined as,

$$\omega'^{i}_{kl} = \frac{2|U_k \cap U_l|}{|U_k| + |U_l|}\omega^{i}_{kl}. \tag{15}$$

– Time-aware Similarity:
As the users' interests may change over time, a growing number of recommender systems have realized the influence of time on the predictive accuracy. [50] proposed a time-aware CF algorithm which takes the effect of time into consideration on similarity computation. On the one hand, ratings with similar timestamps contribute more to the user similarity and the contribution is shown as,

$$f_1(t_{ik}, t_{jk}) = e^{-\alpha|t_{ik}-t_{jk}|}, \tag{16}$$

where t_{ik} and t_{jk} are the timestamps when user i and user j rated item k. α is a non-negative constant, which decides how fast f_1 decreases with the increase of $|t_{ik} - t_{jk}|$. On the other hand, more recent ratings contribute more to the user similarity. Namely, if two users rated the same item a long time ago, their ratings are less important on the prediction at the current time. The contribution is defined as,

$$f_2(t_{ik}, t_{jk}) = e^{-\beta|t_{current}-(t_{ik}+t_{jk})/2}, \tag{17}$$

where $t_{current}$ is the time when the recommendation is performed. Based on the above analysis, the time-aware similarity between user i and user j is defined by Eq. (18).

$$\omega_{ij} = \frac{\sum_{k\in K}(r_{i,k}-\bar{r_i})(r_{j,k}-\bar{r_j})f_1(t_{ik},t_{jk})f_2(t_{ik},t_{jk})}{\sqrt{\sum_{k\in K}(r_{i,k}-\bar{r_i})^2}\sqrt{\sum_{k\in K}(r_{j,k}-\bar{r_j})^2}}. \tag{18}$$

- **Step 2**: Neighbor Selection
The predictive accuracy will be reduced once some dissimilar users are involved into the neighborhood. Therefore, neighbors of the active user should be chosen carefully by certain methods. Traditional Top-N algorithm [29] selects N most similar neighbors to make predictions which ignores the phenomenon that some users may have a limited number of neighbors less than N. [30] proposed that a candidate neighbor whose similarity is smaller than 0 should be removed from the Top-N set which is defined as follows,

$$S(i) = \{i_a | i_a \in N(i), \omega^{u}_{ii_a} > 0, i_a \neq i\}, \tag{19}$$

where $N(i)$ is the set of Top-N similar users. Correspondingly, the neighbors of item k is defined as,

$$S(k) = \{k_a | k_a \in N(k), \omega^{i}_{kk_a} > 0, k_a \neq k\}, \tag{20}$$

where $N(k)$ is the Top-N similar items.
Furthermore, people in different countries or regions are more likely to have different preferences. Therefore user locations are needed to be considered while selecting neighbors for the active user. Thanks to the development of mobile Internet, location information can be obtained through either mobile client or IP address [31], and be transmitted to the server for further analysis. Generally, users can be first divided into several partitions depending on their locations and the spatially closed users have priority in neighbor selection. For instance, [32] proposed a three-tier model of spatial relationships, i.e., the same Autonomous System (AS), same country, and others. Users reaching the similarity threshold with closer spatial relationship are prior to be selected as neighbors of the active user.

- **Step 3**: Prediction
With the neighbors' similarities and ratings, the recommender system calculates the weighted average as the prediction. For user-based CF, predicted rating of user i on item k is calculated as follows,

$$P(i,k) = \bar{r}_i + \frac{\sum_{i_a\in S(i)} \omega^{u}_{ii_a}(r_{i_a,k} - \bar{r}_{i_a})}{\sum_{i_a\in S(i)} \omega^{u}_{ii_a}}, \tag{21}$$

where $S(i)$ is the set of neighbors for user i. $\omega^{u}_{ii_a}$ is the similarity between user i and i_a. For item-based CF, predicted rating of user i on item k is illustrated in Eq. (22).

$$P(i,k) = \bar{r}_k + \frac{\sum_{k_a\in S(k)} \omega^{i}_{kk_a}(r_{i,k_a} - \bar{r}_{k_a})}{\sum_{k_a\in S(k)} \omega^{i}_{kk_a}}, \tag{22}$$

where $S(k)$ is the set of neighbors for item k. $\omega^{u}_{kk_a}$ is the similarity between item k and k_a.

- **Step 4**: Items Ranking
Once the predictions are obtained, the recommender system needs to rank all the items according to their predicted ratings. In order to improve the diversity of

recommendations, some recommender systems also take the popularity into consideration [26]. An item with larger predicated value and lower popularity is supposed to rank higher.

- **Step 5**: Selection of Top-N Items
  After ranking all the candidate items, Top-N of them are provided to the user where N is a parameter needed to be preset before the recommendation task.

B. Model-based CF

Although the memory-based CF is useful in effectively predicting missing ratings and presenting recommendations, it still has a few limitations. For instance, whenever a recommendation task is conducted, the system has to load all the ratings into the memory and implement specific algorithm based on the complete dataset. Limited by the storage and computing resources, memory-based CF may often become quite time-consuming. Therefore, recommender system which can provide proper items with acceptable time consumption is highly desired. Model-based CF algorithms are designed to mitigate these problems whose general principle is to use machine learning or data mining approaches to establish prediction models offline. Based on these models, missing ratings can be predicted efficiently. Typical model-based algorithms include matrix factorization-based algorithms, clustering-based algorithms, etc.

1) Matrix Factorization-based Algorithms: The sparsity of rating matrix is always the major challenge which restricts the performance of collaborative filtering [37]. The cause of this problem is that the vector dimension of users or items is always very large. Matrix factorization (MF) algorithm, one of the unsupervised learning methods, can play a role in reducing dimensionality and eventually alleviating the data sparsity [38]. The main procedures of matrix factorization-based CF algorithms are shown in Fig. 8.

a) Procedure 1 - Latent Feature Modeling: The rating matrix usually contains some latent features which can be used to describe the profiles of users and items more specifically. Take videos for example, the latent features may be the styles of videos, such as comedy, tragedy, etc. As shown in Fig. 7, the user feature vector p_u indicates how much user u is interested in each feature, and the item feature vector q_i measures the degree of each feature for item i. Based on these two vectors, the rating of user u on item i can be calculated by Eq. (23).

$$r_{u,i} = p_u^T q_i. \tag{23}$$

b) Procedure 2 - Determination of Optimization Objective: It is easy to understand that once the feature matrices of users and items are established, all the missing ratings can be obtained conveniently by calculating the dot product of specific feature vectors. However, traditional matrix factorization methods, such as SVD and PCA [40], fail to decompose the rating matrix due to the large numbers of missing values. In fact, the two desired matrices need to meet the requirement that the dot product between p_u and q_i is close to the known value $r_{u,i}$ in the rating matrix [39]. As a result, this can be modeled as an optimization problem with the objective defined as,

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} \left(r_{u,i} - q_i^T p_u\right)^2 + \lambda\left(\|q_i\|^2 + \|p_u\|^2\right), \tag{24}$$

where κ is the set of user-item pairs and $r_{u,i}$ can be obtained from the training set. An additional parameter λ is introduced to mitigate the over-fitting problem.

The above formula sometimes has to face the problems of biases from users or items. For example, ratings from a critical user may be lower than others. Hence, additional variables are needed to take biases into account and the new objective function is shown as follows,

$$\begin{aligned}\min_{q^*,p^*} \sum_{(u,i)\in\kappa} & \left(r_{u,i} - \mu - b_i - b_u - q_i^T p_u\right)^2 \\ & +\lambda\left(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2\right),\end{aligned} \tag{25}$$

where μ is the average of all the ratings. b_u is the difference between μ and the average rating of user u. b_i is the difference between μ and the average rating on item i.

Ratings usually change over time due to the alternation of user interests or the decline of item popularities [35]. In this case, temporal factors are taken into consideration and the objective function is reconstructed as,

$$\begin{aligned}\min_{q^*,p^*} \sum_{(u,i)\in\kappa} & \left(r_{u,i}(t) - \mu(t) - b_i(t) - b_u(t) - q_i^T p_u(t)\right)^2 \\ & +\lambda\left(\|q_i(t)\|^2 + \|p_u(t)\|^2 + b_u(t)^2 + b_i(t)^2\right),\end{aligned} \tag{26}$$

where $r_{u,i}(t)$, $\mu(t)$, $b_u(t)$, $b_i(t)$, $q_i(t)$ and $p_u(t)$ are treated as functions of time.

c) Procedure 3 - Solving the Optimization Problem: Many algorithms are proposed to solve the above optimization problems and the most commonly used ones are stochastic gradient descent [33] and alternating least squares [34].

Stochastic gradient descent (SGD) is an iterative algorithm whose general principle is to update the unknown parameters according to the gradient descent direction of the objective function. For instance, in order to solve Eq. (24), p_u and q_i are initialized randomly at first. And then the prediction error is calculated as follows,

$$e_{u,i} = r_{u,i} - q_i^T p_u. \tag{27}$$

Then p_u and q_i are modified in the opposite direction of the gradient:

$$\begin{aligned} q_i &\leftarrow q_i + \alpha(e_{u,i}p_u - \beta q_i), \\ p_u &\leftarrow p_u + \alpha(e_{u,i}q_i - \beta p_u), \end{aligned} \tag{28}$$

where α and β are two constants which can affect the rate of convergence.

Apart from SGD, alternating least squares is also an effective method to settle these problems. In order to solve Eq. (24),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

	V1	V2	V3
U1	6	2	2
U2	8	3	5
U3	4	2	6

=

	Comedy	Tragedy
U1	2	0
U2	3	-1
U3	2	-2

User Feature Vector

×

	V1	V2	V3
Comedy	3	1	1
Tragedy	1	0	-2

Item Feature Vector

Fig. 7. An example of matrix factorization in video recommendation.

this algorithm first fixes one of the variables and calculates the other by dealing with a least squares problem. Then the two variables rotate, the latter becomes fixed and the former is to be calculated. This procedure continues until the prediction error converges to a stable value [36]. By this means, SVD can be implemented on an incomplete matrix.

2) Clustering-based Algorithms: With the increasing amount of data, it has become a pretty complicated work to calculate the similarities between the active user and all the other users in the dataset. As one of the most widely used data mining methods, clustering can greatly reduce the time and computing resources during recommendation. After a series of specific operations, the input data are divided into several partitions. Objects in the same partition are of higher similarities with each other than those between partitions. Based on clustering, recommender systems can provide proper items with higher reliability and lower computational complexity. Fig. 9 plots the main procedures of the clustering-based algorithms in CF recommender systems.

a) Procedure 1 - Clustering Object Modeling: The first priority of clustering is to answer the questions of what to cluster and how to represent them. Generally, both users and items can be regarded as the objects of clustering [41]. Once the object is determined, various mathematical models can be used to represent it. Rating vectors can be directly utilized to represent users and items. Specifically, an item can be modeled as a multi-dimensional vectors whose features are the ratings from users. Similarly, a user can be represented by all the ratings he has rated on each item.

Although the above method is easy to understand, it is still time-consuming since the dimension of vectors is high. Some additional information can play a vital role in this procedure. For instance, an item can be represented by a number of key words which describe its functions or features and this information can be modeled as a set [42]. Items with similar set elements are likely to be regarded as neighbors. In the same way, users can be represented by the demographic information, such as gender, age, etc. As a result, the dimension has been reduced dramatically compared with the rating vector.

Descriptive information can represent the users or items effectively, however, these data are not always accessible. Usually, only rating data are available for the recommender system. Dimensionality reduction algorithms can be helpful under this circumstance. For example, with the help of Principal Component Analysis (PCA) on the rating matrix, main features can be acquired to represent the previous vector approximately. After that, clustering can be implemented on these vectors with a much lower dimension.

b) Procedure 2 - Similarity Calculation: Once the users or items are represented by specific mathematical models, the selection of similarity measures becomes critical. Basic similarity measures such as Euclidean distance or PCC are still valid for vectors. Moreover, if the clustering objects are a group of sets, specific statistics used for measuring the similarity between two sets come into play. For example, Jaccard similarity coefficient (JSC), which is defined as the size of the intersection divided by the size of the union, is commonly used in model-based recommender systems.

$$\mathrm{Sim}(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|}, \tag{29}$$

where s_i and s_j are the two sets and $|\ |$ represents the cardinality of the set.

c) Procedure 3 - Clustering: After the similarity measure is determined, the next step is to apply specific clustering algorithms on the dataset. Clustering algorithms can be roughly divided into two parts: Partitional Clustering and Hierarchical Clustering [46]. A typical representative of Partitional Clustering algorithms is the well-known K-means, which can divide the complete dataset into K partitions quickly and efficiently [44]. However, the preset parameter K has a significant impact on the results which is difficult to estimate before clustering. In order to mitigate this problem, Hierarchical clustering is proposed which can generate a dendrogram that illustrates the hierarchy of clustering [45]. Based on this dendrogram, various clustering results with different numbers of partitions can be easily obtained.

d) Procedure 4 - Operations after Clustering: Clustering is often regarded as an intermediate step of CF and further operations on the results are needed. For example, after identifying the popularity of each item in the cluster of the active user, some attractive items can be directly selected as recommendations [43]. In addition, once the cluster of the active user is determined, the task of neighbor selection can be implemented just on the users within cluster, instead of all users in the dataset. Consequently, the computational complexity can be reduced dramatically.

3) Other Model-based Algorithms: Apart from the matrix factorization-based and clustering-based algorithms, a plenty of mathematical models have been applied in model-based CF as well, e.g., the Bayesian network, random walk, deep learning, etc. [20] [52] [53]. These approaches have been paid an increasing attention due to the high accuracy and efficiency. For instance, Naive Bayesian classification model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

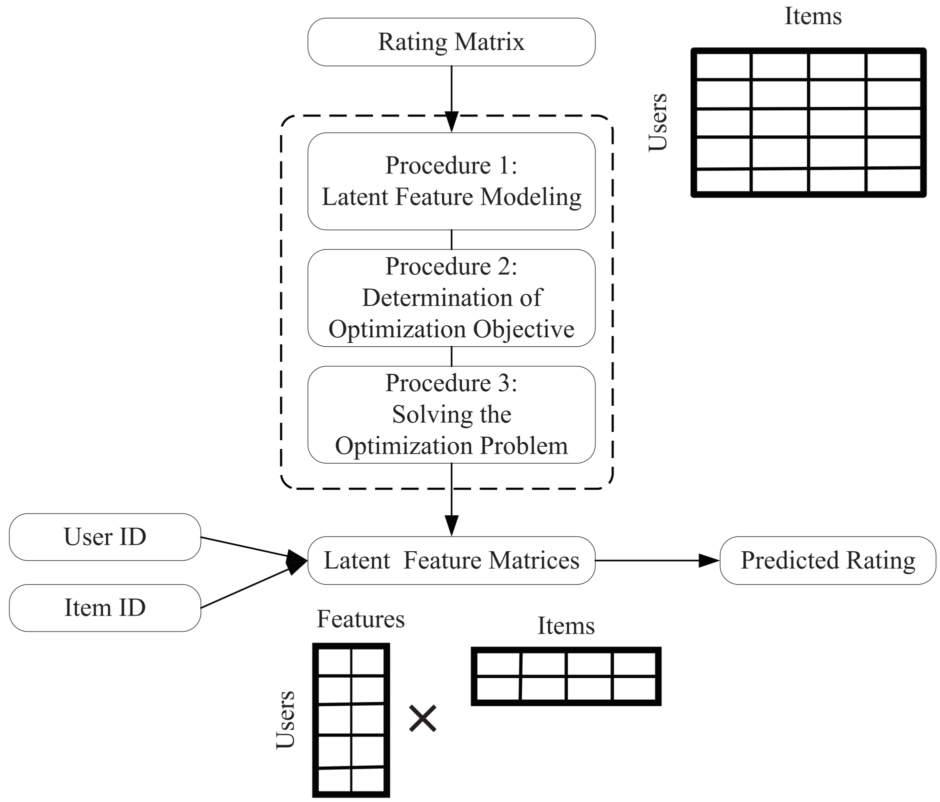


Fig. 8. Main procedures of matrix factorization-based algorithms.

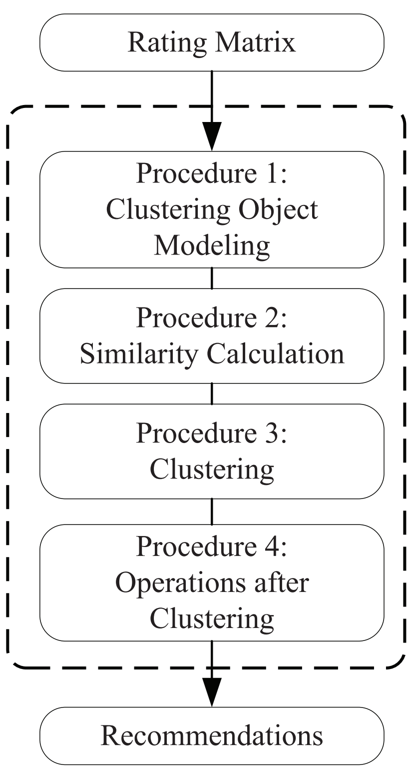


Fig. 9. Main procedures of clustering-based algorithms.

can be applied to predict the rating of user u on item i based on the probability distribution of the known ratings. With the help of this model, probabilities of all the possible ratings can be obtained and the rating with the highest probability is chosen as the prediction, which is defined as

$$\mathrm{R}_p = \underset{r \in RatingSet}{arg\max} \; P(R_r) \prod_n P(X_n = x_n | Y = R_r), \tag{30}$$

where R_p is the predicted rating. X_n indicates the ratings on item i from other users. R_r represents the ratings of user u on other items.

IV. Case Studies Of Recommender Systems Based on CF Algorithms

In recent years, an increasing number of individuals prefer to watch movies or TV plays using mobile applications [51]. Therefore, video websites have attracted growing attention, such as YouTube and Netflix. Collaborative filtering helps these websites to provide desirable recommendations for the consumers lest they should spend a lot of time looking for their favorite videos. In order to explain the CF recommender system more specifically, two case studies are presented in this section based on either user behaviors or user ratings. Experiments carried out in these case studies aim to compare the performances among several typical CF algorithms, and illustrate the impact of key parameters on MAE. Experimental results have shown that both user ratings and behaviors can be employed in CF through specific pre-porcessing operations. Furthermore, CF algorithms usually have a great improvement

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

User ID	Video ID	Download	Play	Share	Like	Ratings
U1	A	√	√		√	3
U1	B	√				1
U2	A	√		√		2
U3	C	√				1

	A	B	C
U1	3	1	
U2		2	
U3			1

Fig. 10. Generation of implicit ratings based on operation records.

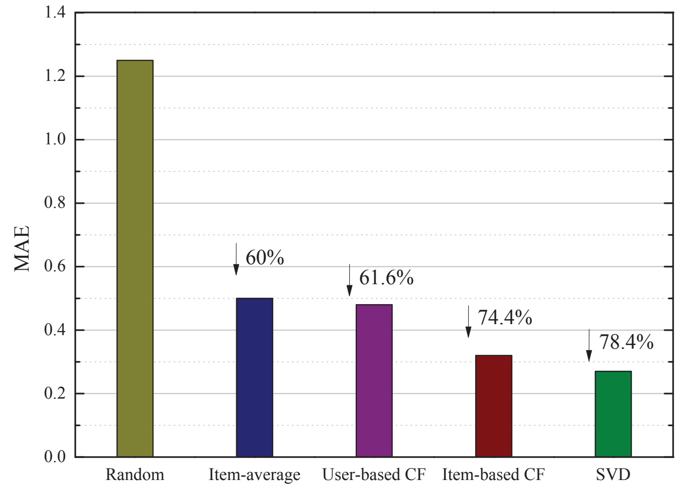


Fig. 11. Comparison of MAE among typical recommendation algorithms.

on the predictive accuracy compared with the baseline, and the model-based CF, such as SVD, is superior to the memory-based CF in some cases. In addition, some parameters of CF may have significant impacts on the predictive accuracy, e.g., the similarity measure, neighbor size and ratio of training set.

A. Case 1: CF Based On User Behaviors

This case study aims to explain how to implement the CF algorithms on the basis of user behaviors. Real-world data are collected from the mobile applications of a fast-growing video platform at China. Records of certain user behaviors are collected and sent to the database through this application. After getting rid of the dirty data and removing the users whose operation records are less than 20, 1131053 records are reserved in this case study which involve 16082 consumers and 1982 videos. Among all kinds of user behaviors, 4 of them are chosen to represent the user preferences for videos, which are “Download”, “Play”, “Like” and “Share”. Specific algorithms need to be used to transform these operation records into ratings. In this case study, for the sake of simplicity, the number of the above operations that one user has implemented on a certain video is regarded as implicit ratings, which is illustrated in Fig. 10. It is evident that the implicit rating based on user behaviors varies from 1 (bad) to 4 (excellent), which reflects the user interest on a certain item. The complete dataset is divided into two parts: 80% of the ratings are regarded as the training set and the others as test set. MAE is selected as the metric of predictive accuracy and the similarity between users or items is measured by Pearson correlation coefficient (PCC).

Random recommendation which predicts missing rating randomly is regarded as the baseline algorithm. Recommendation algorithms considered in this case study are listed as follows.

- Random
- User-based
- Item-average
- Item-based
- SVD

As shown in Fig. 11, the predictive accuracy of random algorithm is the lowest since it has not made use of any data. The user-based CF is slightly better than the item-average algorithm on account of the more accurate neighbor selection. In this case study, item-based CF is superior to user-based CF. One probable cause is that the number of users is far more than that of items, making it more difficult to find neighbors properly. SVD, the representative of mode-based CF, has the highest predictive accuracy in this experiment because two feature matrices are obtained through matrix factorization in order to describe the features of users and items more specifically. There will be some changes on the results along with the variations of some important parameters. The effects of these parameters are analyzed detailedly in the second case study.

B. Case 2: CF Based On User Ratings

This case study aims to discuss the effects of key parameters on the predictive accuracy, including similarity measure, neighbor size and ratio of training set. Dataset from Movielens [54] is used to conduct this experiment. Movielens is a research website run by GroupLens Research at the University of Minnesota. Hundreds of users visit this website and rate movies each week [47]. GroupLens Research collected data during a period of seven months from September 19th 1997 to April 22nd 1998 and filtered the dataset by removing the users who rated less than 20 movies. The dataset used in this case study consists of 100000 ratings from 943 users on 1683 movies and the ratings vary from 1 (bad) to 5 (excellent). Also, the dataset is divided into two parts as training set and test set. The variable α represents the ratio of training set. For example, $\alpha = 0.8$ indicates that 80% of the data are selected as training set and the rest 20% as test set. Both the user-based and item-based algorithms are implemented and MAE is selected to measure the predictive accuracy of the above algorithms.

The performances of three similarity measures are analyzed at first, including pure cosine, adjusted cosine and PCC. Other parameters are set to constant: the neighbor size is 24 and the ratio of training set is 80%. Experimental results are shown in Fig. 12(a). It can be observed that for both user-based and item-based algorithms, PCC has an obvious advantage. Besides, for user-based CF, adjusted cosine is the same as PCC, which has been illustrated in Section III.

Neighbor size also has a significant impact on the quality of prediction [48]. Therefore, an experiment is carried out to calculate the MAE as the neighbor size varies from 5 to 80 with the value of α fixed at 0.8. Experimental results are shown in Fig. 12(b) and Fig. 12(c). It can be seen clearly that all these curves have a similar trend and the optimal neighbor size of each scenario can be easily obtained. Taking the scenario of user-based CF with PCC (green line in Fig. 12(b)) as an example, the MAE decreases rapidly as neighbor size increases from 5 to 32 and then the curve begins to rise. Therefore, the optimal neighbor size is 32. This is because at the beginning, the growth of neighbor size involves more high-similarity neighbors to predict the rating of the target item. However, once the neighbor size exceeds the optimal value, low-similarity neighbors are involved which could degrade the neighborhood and lead to the rise of MAE. For user-based CF, the curve of adjusted cosine overlaps on the curve of PCC, which is explained in Section III.

According to the above experiments, item-based CF with PCC performs better. Therefore, it is selected to analyze the impact of α on the results. An experiment is conducted as α varies from 48% to 80% with the step interval being 4%. Neighbor size is also considered in this experiment which varies from 12 to 32 with the step interval being 4. The experimental results are shown in Fig. 12(d). It can be observed that with the increase of α, the predictive accuracy improves owing to the enrichment of training data.

V. Conclusion

This article discusses CF algorithms employed in mobile Internet applications. A framework is firstly proposed in order to demonstrate the main procedures of a typical CF recommender system, i.e., data collection, data pre-processing, and collaborative filtering. Features of two kinds of user data, i.e., user behaviors and user ratings, are analyzed and compared in detail. After transforming the user behaviors into implicit ratings through specific methods, the sparsity problem of the rating matrix can be mitigated to some extent. Typical CF algorithms including memory-based and model-based are introduced, and their general procedures are summarized for the sake of revealing the common features of these methods. Finally, in order to validate this framework, two case studies were carried out based on the user behaviors and user ratings respectively. Each case study implemented certain typical CF algorithms and then compared their performances in predictive accuracy with the metric of MAE. Although significant progress has been made in CF, further studies are still needed in certain aspects. For example, some recommender systems fail to process massive data in time limited by the storage and computing capabilities. Therefore, in order to improve processing capacity and reduce time consumption, developing algorithms for distributed computing systems may become a significant research direction in the future. It is believed that the continuously improved CF recommender systems can greatly help mobile Internet users find proper items without excessive time and energy consumption in the era of Big Data.

VI. Acknowledgment

This work was supported by the National High-Tech R&D Program (863 Program 2015AA01A705), the National Key Technology R&D Program of China under grant 2014ZX03003011-004, the China Natural Science Funding under the grant 61271183, and the Fundamental Research Funds for the Central Universities under grant 2014ZD03-02.

References

[1] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Network*, vol. 30, no. 1, pp. 44-51, Jan. 2016.

[2] C. Zhao and J. Wang, "Network education video recommendation algorithm based on context and trust relationship," in *Proc. IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, May 2013, pp. 537-540.

[3] Z. Huang, D. Zeng, and H. Chen, "A comparison of collaborative-filtering recommendation algorithms for e-commerce," *IEEE Intelligent Systems*, vol. 22, no. 5, pp. 68-78, Sep. 2007.

[4] Z. Su, Q. Xu, H. Zhu, and Y. Wang, "A novel design for content delivery over software defined mobile social networks," *IEEE Network*, vol. 29, no. 4, pp. 62-67, Aug. 2015.

[5] Z. Su, Q. Xu, and Q. Qi, "Big data in mobile social networks: a QoE-oriented framework," *IEEE Network*, vol. 30, no. 1, pp. 52-57, Jan. 2016.

[6] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, Z. Ma, and J. Guo, "A cross-domain recommendation model for cyber-physical systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 3, pp. 384-393, July 2013.

[7] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453-466, Sep. 2014.

[8] Y. Doen, M. Murata, R. Otake, M. Tokuhisa, and Q. Ma, "Which feature is better? TF*IDF feature or topic feature in text clustering," in *Proc. IEEE International Conference on Multimedia Information Networking and Security (MINES)*, Nanjing, Nov. 2012, pp. 425-428.

[9] H. Yang, G. Ling, Y. Su, M. R. Lyu, and I. King, "Boosting response aware model-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2064-2077, Feb. 2015.

[10] W. Nadee, Y. Li, and Y. Xu, "Acquiring user information needs for recommender systems," in *Proc. IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Atlanta, GA, Nov. 2013, pp. 5-8.

[11] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34-44, Jun. 2013.

[12] K. Wei, J. Huang, and S. Fu, "A survey of e-commerce recommender systems," in *Proc. IEEE International Conference on Service Systems and Service Management*, Chengdu, June 2007, pp. 1-5.

[13] M. Nilashi, O. Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3879-3900, Jun. 2014.

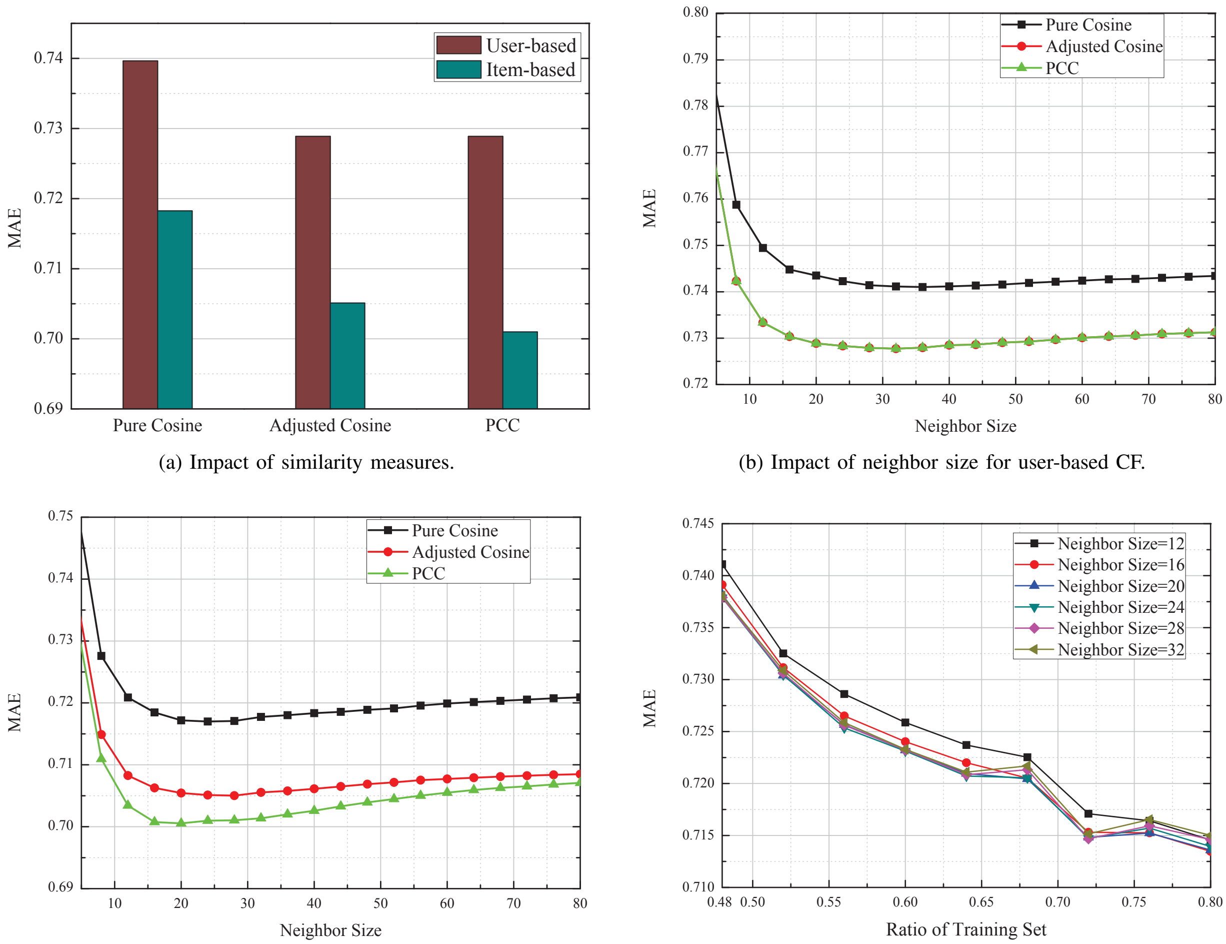
[14] D. H. Choi and B. S. Ahn, "Eliciting customer preferences for products from navigation behavior on the web: a multicriteria decision approach with implicit feedback," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 39, no. 4, pp. 880-889, May. 2009.

[15] H. Y. Hsieh, V. Klyuev, Q. Zhao, and S. H. Wu, "SVR-based outlier detection and its application to hotel ranking," in *Proc. 6th IEEE International Conference on Awareness Science and Technology (iCAST)*, Paris, Oct. 2014, pp. 29-31.

[16] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei "Soft-defined heterogeneous vehicular network : architecture and challenges," *IEEE Network Magazine*, accepted, arXiv preprint arXiv:1510.06579.

[17] S. Chen, Y. Shi, B. Hu, and M. Ai, "Mobility-driven networks (MDN): From Evolutions to Visions of Mobility Management," *IEEE Network magazine*, vol. 28, no. 4, pp. 66-73, Aug. 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access



(a) Impact of similarity measures.

(b) Impact of neighbor size for user-based CF.

(c) Impact of neighbor size for item-based CF.

(d) Impact of the ratio of training set.

Fig. 12. Impact of key parameters on MAE.

[18] X. Zhu, S. Chen, H. Hu, Y. Shi, X. Su, "TDD-based Mobile Communication Solutions for High Speed Railway Scenario," *IEEE Wireless Communications*, vol. 20, no. 12, pp. 22-29, Dec. 2013.

[19] G. Go, J. Yang, H. Park, and S. Han, "Using online media sharing behavior as implicit feedback for collaborative filtering," in *Proc. 2nd IEEE International Conference on Social Computing (SocialCom)*, Minneapolis, Aug. 2010, pp. 20-22.

[20] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, no. 4, Jan. 2009.

[21] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk, "A unified approach of factor models and neighbor based methods for large recommender systems," in *Proc. 1st IEEE International Conference on the on the Applications of Digital Information and Web Technologies (ICADIWT)*, Ostrava, Aug. 2008, pp. 186-191.

[22] Y. Mo, J. Chen, X. Xie, C. Luo, and L. T. Yang, "Cloud-based mobile multimedia recommendation system with user behavior information," *IEEE Systems Journal*, vol. 8, no. 1, pp. 184-193, Jan. 2014.

[23] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[24] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-based collaborative filtering for citation recommendation," *IEEE Access*, vol. 3, pp. 1695-1703, Oct. 2015.

[25] T. Zhou, Z. Kuscsik, J. Liu, M. Medo, J. Wakeling, and Y. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 10, pp. 4511-4515, Mar. 2010.

[26] C. Yang, C. Ai, and R. Li, "Neighbor diversification-based collaborative filtering for improving recommendation lists," in *Proc. 10th IEEE International Conference on High Performance Computing and Communications*, Zhangjiajie, Nov. 2013, pp. 13-15.

[27] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 896-911, Apr. 2012.

[28] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative web service quality prediction via exploiting matrix factorization and network map," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126-137, Jan. 2016.

[29] I. Bartolini, Z. Zhang, and D. Papadias, "Collaborative filtering with personalized skylines," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 2, pp. 190-203, May 2010.

[30] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR '99)*, Berkeley, 1999, pp. 230-237.

[31] C. Priya and S. Rani, "Location-aware and personalized collaborative filtering for web service recommendation : a review," *International Journal of Computer Applications*, vol. 133, no. 14, pp. 1-3, Jan. 2016.

[32] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for web service recommendation," *IEEE Transactions on Services Computing*, May. 2015, DOI: 10.1109/TSC.2015.2433251.

[33] H. F. Yu, C. J. Hsieh, S. Si, and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems," in *Proc. 12th IEEE International Conference on Data Mining (ICDM)*, Brussels, Dec. 2012, pp. 765-774.

[34] M. Gates, H. Anzt, J. Kurzak, and J. Dongarra, "Accelerating collaborative filtering using concepts from high performance computing," in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2573314, IEEE Access

Proc. IEEE International Conference on Big Data, Santa Clara, CA, Nov. 2015, pp. 667-676.

[35] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30-37, Aug. 2009.

[36] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, “Large-scale parallel collaborative filtering for the Netflix Prize,” *Algorithmic Aspects in Information and Management*, vol. 5034, pp. 337-348, 2008.

[37] A. Gogna and A. Majumdar, “A comprehensive recommender system model: improving accuracy for both warm and cold start users,” *IEEE Access*, vol. 3, pp. 2803-2813, Dec. 2015.

[38] D. Bokde, S. Girase, and D. Mukhopadhyay, “Matrix factorization model in collaborative filtering algorithms: a survey,” *Procedia Computer Science*, vol. 49, pp. 136-146, Apr. 2015.

[39] D. Bokde, S. Girase, and D. Mukhopadhyay, “Role of matrix factorization model in collaborative filtering algorithm: a survey,” *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, vol. 1, no. 6, May 2015.

[40] A. Morshed, R. Dutta, and J. Aryal, “Recommending environmental knowledge as linked open data cloud using semantic machine learning,” in *Proc. 29th IEEE International Conference on Data Engineering Workshops (ICDEW)*, Brisbane, QLD, April 2013, pp. 27-28.

[41] D. Zhang, C. H. Hsu, M. Chen, Q. Chen, N. Xiong, and J. Lloret, “Cold-start recommendation using Bi-clustering and fusion for large-scale social recommender systems,” *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 239-250, July 2014.

[42] R. Hu, W. Dou, and J. Liu, “ClubCF: a clustering-based collaborative filtering approach for big data application,” *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 302-313, Oct. 2014.

[43] G. M. Dakhel and M. Mahdavi, “A new collaborative filtering algorithm using K-means clustering and neighbors’ voting,” in *Proc. IEEE International Conference on Hybrid Intelligent Systems (HIS)*, Melacca, Dec. 2011, pp. 179-184.

[44] K. Lee and K. Lee, “Using dynamically promoted experts for music recommendation,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1201-1210, July 2014.

[45] S. Renaud-Deputter, T. Xiong, and S. Wang, “Combining collaborative filtering and clustering for implicit recommender system,” in *Proc. 27th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Barcelona, Mar. 2013, pp. 748-755.

[46] U. Gupta and N. Patil, “Recommender system based on hierarchical clustering algorithm Chameleon,” in *Proc. IEEE International Advance Computing Conference (IACC)*, Banglore, June 2015, pp. 1006-1010.

[47] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, 2001, pp. 285-295.

[48] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Proc. Proceedings of ACM SIGIR’99. ACM press*, Berkeley, 1999, pp. 230-237.

[49] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “WSRec: a collaborative filtering based web service recommender system,” in *Proc. IEEE International Conference on Web Services*, Los Angeles, CA, July 2009, pp. 437-444.

[50] Y. Hu, Q. Peng, X. Hu, and R. Yang, “Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782-794, Dec. 2014.

[51] F. Xia, N. Y. Asabere, A. M. Ahmed, J. Li, and X. Kong, “Mobile multimedia recommendation in smart communities: a survey,” *IEEE Access*, vol. 1, pp. 606-624, Sep. 2013.

[52] G. Xu, B. Fu, and Y. Gu, “Point-of-interest recommendations via a supervised random walk algorithm,” *IEEE Intelligent Systems*, vol. 31, no. 1, pp. 15-23, Jan. 2016.

[53] H. Wang, N. Wang, and D. Yeung, “Collaborative deep learning for recommender systems,” in *Proc. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, Aug. 2015, pp. 1235-1244.

[54] Movielens.org, “MovieLens”, 2016. [Online]. Available: http://movielens.org. [Accessed: 13- May- 2016].