Received December 13, 2018, accepted January 3, 2019, date of publication January 14, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891544

Cold Start Recommendation Based on Attribute-Fused Singular Value Decomposition

XING GUO[1], SHI-CHAO YIN[1], YI-WEN ZHANG[1], WEI LI[1], AND QIANG HE[2]
[1]School of Computer Science and Technology, Anhui University, Hefei 230601, China
[2]School of Software and Electrical Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

Corresponding author: Yi-Wen Zhang (zhangyiwen@ahu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872002, in part by the National Key Technology Research and Development Program under Grant 2015BAK24B01, and in part by the Natural Science Foundation of Anhui Province of China under Grant 1808085MF197.

ABSTRACT Collaborative filtering plays an important role in promoting the service recommendation ecosystem, and the matrix decomposition technology has been proven to be one of the most effective recommendation methods. However, the traditional collaborative filtering algorithm has great shortcomings in the recommendation of cold start items, especially the emergence of new items will be largely ignored. This not only has a very bad impact on the development of the item, but also greatly reduces the diversity of the recommendation system. The rise of mobile devices has also brought a large number of mobile applications, and these emerging applications need to be promoted in order to maintain the robustness of the application system. In order to solve this problem, we propose a method of combining the attribute information of the item with the historical rating matrix to predict the potential preferences of the user. It combines the attribute and time information into a matrix decomposition model. By testing our method on the movielens and the climbed JD dataset, the experimental results show that, compared with the baseline method, the proposed method achieves a significant improvement in recommendation accuracy. Therefore, this method is an effective way to solve the cold start problem of new items.

INDEX TERMS Cold start, matrix factorization, recommendation systems, mobile recommendation.

I. INTRODUCTION

With the development of information technology and Internet technology, people have entered the era of information overload from the era of information scarcity. In the information overload era, it is increasingly difficult for people to find information of their own interest from a large amount of information [1]. The task of the recommendation system [2]–[4] is to connect users and information to create value.

Service recommendation technology plays an important role in helping users access information. Today, due to the development of the Internet, the network generates a large amount of data, and it is difficult for users to obtain information of interest from massive amounts of data. Such as movies, books, news, music, etc. In fact, the recommendation system has a particularly significant impact on the content business. For example, two-thirds of movie watch records are caused by suggestions in the Netflix case [5], [6]. More than 38% of Google News clicks are generated by recommendations, while 35% of Amazon's [7] sales are attributed to recommendations.

As the rapid spread and development of mobile smart devices, traditional web services began to migrate to mobile devices [8]. In recent years, mobile applications have become more and more popular, such as food, takeaways, online shopping and so on. According to official data from Jingdong, during the double eleventh period, Jingdong's turnover was 159.8 billion yuan, and most of the goods were sold on mobile devices. With the development of mobile service, the information overload has also been brought to mobile applications. Therefore, the service recommendation technology urgently needs to migrate from the web side to the mobile side.

However, as shown in Fig.1, most of the recommended technologies used in recommendation systems are based on collaborative filtering [9]–[11]. The main idea of collaborative filtering is to analyze historical feedback information of users and items [12]–[14]. Users who have used more similar items tend to have more similar preferences, and multiple items that users have used may have similar attributes. But there are still certain problems with the collaborative filtering method. For example, using traditional collaborative filtering techniques is impossible to recommend some new items such

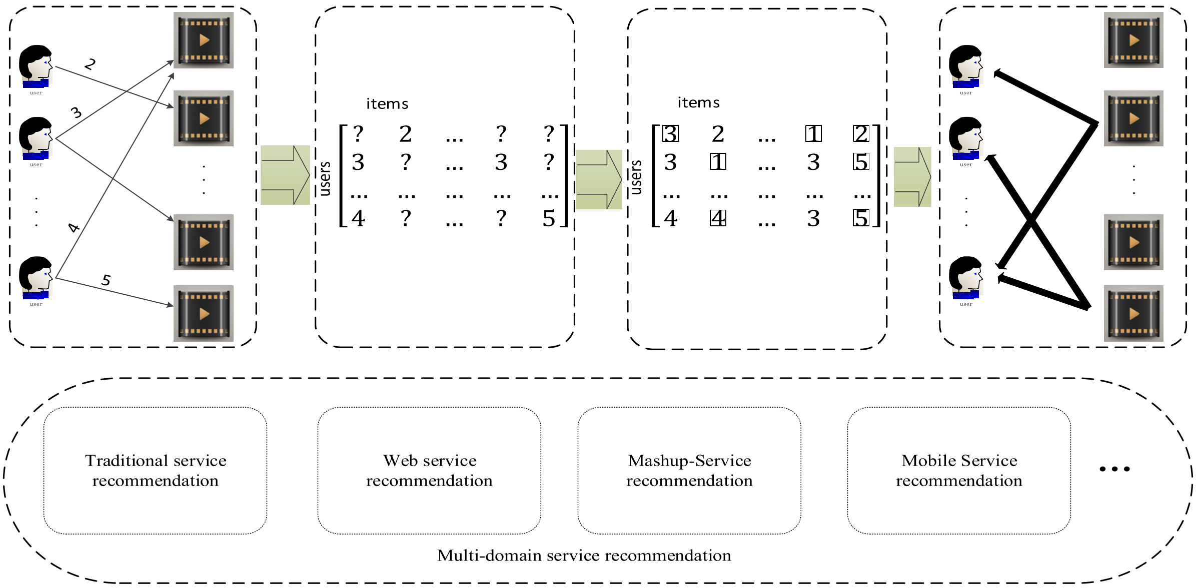

FIGURE 1. Illustration of service recommendation.

as the release of a new movie, the launch of a new product, or the launch of a new item. On the past day of Nov.11th, many merchants have launched their own new products which have just been listed for consumers and no one knows whether the quality of the products is good or bad or meets the needs of individuals. In this case, these products will be difficult to get recommendations by using the traditional collaborative filtering recommendation method, which will bring a great break to the merchants because they invested a lot of resources on these newly listed products. For the users, their satisfaction with the mobile application will be greatly reduced if the recommendation is not pleased by them, which is not beneficial to users and application providers. Therefore, the recommendation for cold start in the recommendation system is urgently needed to be solved.

In this work, we focus on using item attributes to handle cold start recommendations. The item's attribute information is usually provided at the time of producer release, including the category, publisher, and publication year. Based on the item information, we propose a new model which is called attribute-fused singular value decomposition (FASVD) to solve the cold start recommendation problem of the item. The method extracts the user's interest through the user's historical behavior to establish associations between the user's interest and the attributes of the new item. And then it can predict user preferences for new items through established associations and ultimately make recommendations.

The main contributions of this paper are as follows:

(1) We use the attributes of the item to calculate the similarity instead of the traditional rating, and convert the user item rating matrix information into user attribute rating information. Then we predict the user's rating of the cold start item based on the relationship between the attribute and the item.

(2) We believe that the user's interest may change over time. Therefore, we add a time penalty factor when calculating the similarity between items to reduce the huge impact of long-term items on current item impact.

(3) We validate the effectiveness of our method in the traditional film recommendation field and the dataset crawled on the mobile Jingdong. The experimental results show that the proposed method is significantly higher than other well-known methods in MAE and RMSE. And our method has some portability.

The structure of this paper is as follows. Section 2 introduces the background and related work. Section 3 discusses the problem statement. Section 4 describes the cold start problem and the present methods. Section 5 reports the evaluation results. Finally, we conclude the paper in Section 6.

II. RELATED WORK

In this section, we review the related work including two representative collaborative filtering approaches. Then, we discuss the cold start issue and the key techniques for cold start recommendations.

Collaborative filtering (CF), since it was introduced in the Tapestry recommender system [1], has been considered as one of the most effective techniques in various mobile service recommender systems [15]–[17]. It is a method for making automatic predictions about the interests of a user by collecting preferences or taste information from many users [18]. Generally, CF techniques can be decomposed into two categories: memory-based methods and model-based methods.

The memory-based CF is also named the neighborhood-based CF. It includes the user-based CF [9], [10], [18], the item-based CF [13], [14] and the hybrid CF [19]. In the user-based CF, a subset of appropriate users is chosen as neighbors based on their similarities to the active user. Then, a weighted aggregate of their ratings is used to generate predictions for the target user. In the item-based CF, a subset of appropriate items is chosen as neighbors based on their similarities to the target item. The idea of the hybrid-based CF is to combine the user-based CF predicting result and the item-based CF predicting result with weights for the target user. For measuring the similarity between users or items, the Pearson Correlation and Cosine Similarity are widely used in various CF methods. Both of these formulas calculate the similarity based on the ratings of two users on the same items.

Although the memory-based CF has been widely used, it still has many disadvantages. For example, it cannot scale to large-scale data and it performs poorly in the case of sparse data. What's more, it can't predict ratings by finding similar users or similar services because of the lack of history ratings, which indicates memory-based CF cannot handle cold-start problems. Model-based CF methods use statistical and machine learning techniques to train models using historical ratings, and they include clustering models [20]–[22], latent factor models [23], matrix factorization [24], locality-sensitive hashing [25], intelligence algorithm [26], [27] and so on. Such algorithms can be quickly recommended for the user and have good performance. As one of the most classical algorithms of the model-based CF, matrix factorization (MF) has been widely used. MF aims to learn the latent factors with the assumption that the ratings are based on the interactions between users' latent factors and items' latent factors [28], [29]. By minimizing the sum of squared distances using the gradient descent method (SGD) [30] or the alternating least squares (ALS) [5], the efficiency of the model training is effectively improved. Mnih and Salakhutdinov [31] gave their explanation on matrix factorization in a probabilistic way. Balakrishnan and Chopra [32] proposed a new ranking metric for latent factor models. In 2010, Agarwal and Chen [33] first assembled latent Dirichlet allocation (LDA) [23] and matrix factorization as the fLDA. Chen *et al.* [34] used content from social networks as the input of the CTR to improve the quality of recommendation for users.

The traditional CF only takes into account ratings information, and the performance is limited in terms of recommendations for data sparsity and cold start issues. To alleviate the problem of data sparsity, several methods have been proposed to apply some auxiliary information to the recommendation process. In addition to considering the ratings information [35]–[37], some studies added information about social networks to address the issue of data sparsity. For example, Shi *et al.* [36] proposes a flexible regularization framework that integrates different types of users and items relationship information into the referral process. Xu *et al.* [37] uses the trust relationship between users to improve the performance of the recommendation system.

Since collaborative filtering relies on the ratings completely, a problem occurs when there are either no known ratings for a specific item/user. In this case, it is difficult to make reliable recommendations [38]. The academic community divides the cold start problem into three categories: new systems, new items and new users. The cold start problem of a system refers to the "start-up of" a new recommender system. New item and new user problems occur when a new item/user enters an already existing system. In this paper, we focus on the new item cold start problem.

Some approaches for the cold start problem's solution have been proposed. Recently, Li *et al.* [39] proposed a trust-based recommendation model based on relevant information of similar stores, recommending new products to new shop openings in social networks and mitigating the issue of "cold start" on certain issues. Sarwar *et al.* [40] introduce an incremental singular value decomposition (iSVD) algorithm for the cold-start users. Tackcs *et al.* [41] and Rendle and Schmidt-Thieme [42] also provide incremental algorithms to update the latent factor vectors for cold-start users when they give new ratings.

TABLE 1. Symbols and descriptions.

Symbols	Definition and Description
u,v	users
i, j	items
α	the learning rate of matrix factorization
λ	the regularization parameters
r_{ui}	real rating from user u to item i
$\hat{r}_{ui}$	the predicted rating from user u to item i
$\bar{r}_u$	the average rating of user u
x_u	the latent profiles for user u
y_i	the latent profiles for item i
w_{ij}	time penalty factor
b_u	the biases of user u
b_i	the biases of item i
U	the set of users
I	the set of items
R	the matrix of ratings
$A(i)$	the attribute set of item i
T	the time set of items
$I(u)$	the set of items that have been rated by user u
$U(i)$	the set of users who have rated item i
$N(i)$	the similar set of neighbors for user i
D	the dimension of matrix factorization
$Top\text{-}K$	the most similar k items

III. PRELIMINARY

In this section, we explain the symbols and define the problem statement of recommending cold start items to users. However, before the illustration, we summarize the main notations used in this paper in Table 1.

Based on these notations, we define the problem of the item cold start as follows.

Definition 1 (Item Cold Start Problem):

Given: (1) the existing user-item rating matrix, (2) the existing item attribute information, and (3) new item attribute information.

Find: Predict user ratings for new items.

As the definition shows, the input of our problem includes the existing ratings from users to existing items, old items' attribute information and new items' attribute information. In the existing recommendation system, it is difficult to recommend cold start items because of the lack of a sufficient historical rating. How to solve the cold start of new items is the problem to be solved in this paper.

In recommendation systems, due to the lack of historical ratings, it is difficult to model the relationship between users and cold start items. When a cold start item enters the recommendation system, for enriching the system's diversity, the first thing to do is to recommend the new item efficiently. However, many existing methods treat the cold start items as regular items. Although some methods claim that they can solve the cold start problem, none of them can effectively recommend new items when they enter the system for the first time.

A. K NEAREST NEIGHBORS ALGOTIYHM

K nearest neighbor algorithm (KNN) is one of the popular approaches in neighborhood-based collaborative filtering. There are two types of KNNs in the recommendation system (item-based KNN and user-based KNN). The principle of KNN is replacing the target user with the K nearest neighbors. The most important thing is the calculation of similarity.

User-based KNN: The thinking of the user-based KNN algorithm is that users with similar tastes tend to give similar ratings for the same items. It is important how to calculate the similarity between each pair of the given users. The Pearson Correlation Coefficient (PCC) is a widely used similarity computation approach in the KNN algorithm. Then, the user-based KNN using the PCC for the similarity calculation is respectively defined as follows:

$$sim(u,v) = \frac{\sum_{i\in U(u)\cap U(v)}(r_{ui}-\bar{r}_i)(r_{uj}-\bar{r}_j)}{\sqrt{\sum_{i\in i\in U(u)\cap U(v)}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in i\in U(u)\cap U(v)}(r_{vi}-\bar{r}_v)^2}} \tag{1}$$

Then, the user ratings calculated by the User-based KNN are defined as follows:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{j\in N(u)} sim(u,v)\times(r_{vi}-\bar{r}_v)}{\sum_{j\in N(u)} sim(u,v)} \tag{2}$$

Item-based KNN: The idea of the Item-based KNN is to find similar items based on users' preferences for items, and then recommend similar items to users according to the user's historical preference. Similar to the UKNN, the key of the item-based KNN is also the similarity calculation. The user-based KNN using the PCC for the similarity calculation is respectively defined as follows:

$$sim(i,j) = \frac{\sum_{u\in U(i)\cap U(j)}(r_{ui}-\bar{r}_i)(r_{uj}-\bar{r}_j)}{\sqrt{\sum_{u\in U(i)\cap U(j)}(r_{ui}-\bar{r}_i)^2}\sqrt{\sum_{u\in U(i)\cap U(j)}(r_{uj}-\bar{r}_j)^2}} \tag{3}$$

Then, the user ratings calculated by the Item-based KNN are defined as follows:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j\in N(i)} sim(i,j)\times(r_{uj}-\bar{r}_j)}{\sum_{j\in N(i)} sim(i,j)} \tag{4}$$

B. MATRIX DECOMPOSITION

As one of the most classical algorithms of model-based CF, matrix factorization (MF) has been widely used. The main idea of matrix factorization is to reduce the ratings matrix to two low-dimensional matrices. The basic principle of matrix factorization is mapping the ratings matrix to user-based and item-based latent profiles. Specifically, each user u and each item i correspond to the vectors x_u and y_i. For each item i, y_i is used to measure the extent to which items belong to these factors. Similarly, for each user u, x_u is used to measure the interest that a user has in items that meet certain factors. We denote the rating of user u rated item i as r_{ui}, which represents the user's preference for item i. A higher r_{ui} implies a stronger preference and the goal of MF is to predict an unknown rating. To predict the rating $\hat{r}_{ui}$ from user u to item i, we just need to compute $\hat{r}_{ui}$ as in Eq. (1).

$$\hat{r}_{ui} = x_u^T y_i \tag{5}$$

To train the factor vectors $\mathbf{x_u}x_u$ and $y_i\mathbf{y_i}$, the following optimization formulation is usually constructed.

$$\min_{x_*,y_*} \sum_{(u,i)\in D}(r_{ui}-\hat{r}_{ui})^2 + \gamma(||x_u||^2+||y_i||^2) \tag{6}$$

$$e_{ui} = r_{ui} - x_u^T y_i \tag{7}$$

where set D is the users and items set that has been rated. γ is the parameter that regulates the degree of regularization to prevent overfitting, and e_{ui} represents the error between the predicted value and the real value.

To this end, we are committed to exploring the potential relationships between new items and existing items and then quantifying these relationships. We will describe how to model the relationships between new items and existing items in the next section.

IV. METHODS

In this chapter, we first describe how we establish the connection between new items and existing items and then incorporate the results into the model of matrix factorization to further reduce the prediction error. As shown in Fig. 2, our cold start recommendation method consists of the following main stages.

Stage 1: Based on the attribute information of the items, this paper proposes an Attribute-based KNN method. The method uses the attribute information to establish the association between the existing items and the cold start items. Furthermore, this method also adds time information to reduce the weights of items that are too distant for cold start items.

Stage 2: Use the SVD model to predict the missing ratings of existing items.

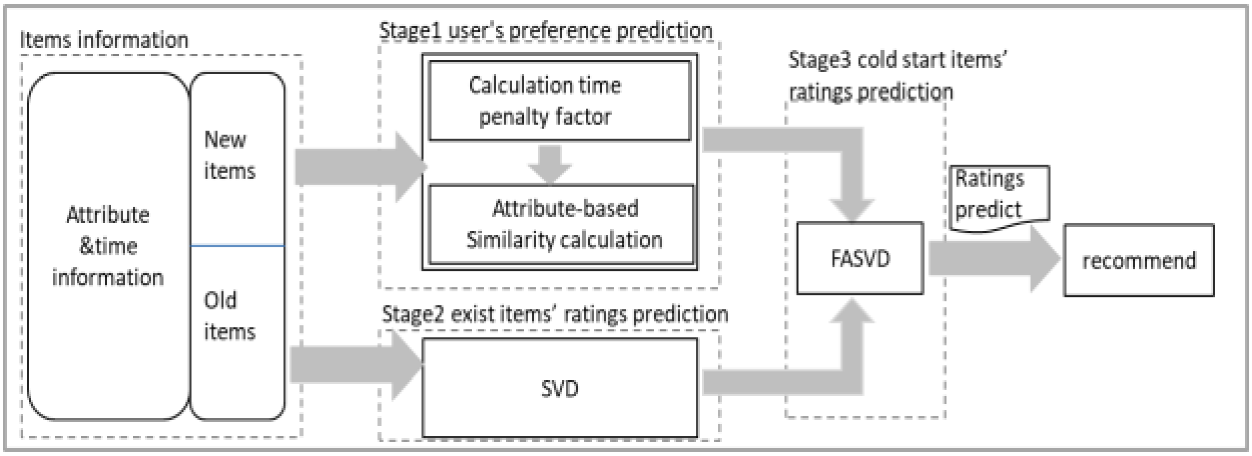

FIGURE 2. Framework of attribute-fused singular value decomposition.

Algorithm 1 Item Attribute Matrix Generation Algorithm

Input: I_{old} (old items set) and I_{new} (new items set)
Output: *IAmatrix* (items attribute matrix)

```
Initialize Aset, IAmatrix
for i ∈ I_old do
   Aset ← Aset ∪ A_i
   update(IAmatrix)
end for
for i ∈ I_new do
      if A_i ∪ Aset do
         update(IAmatrix)
      end if
end for
return IAmatrix
```

Stage 3: The user preference prediction part is fused with the SVD model and obtains the new prediction model – FASVD. This method combines the properties of the items to predict the cold start items.

A. STAGE 1: USERS' PREFERENCE PREDICTION

As we discussed in Section 2, those methods use historical ratings to model the existing items and new items. However, those methods may not adapt to the "cold start" problem discussed in this paper. Because there are no useful ratings for cold start items, it makes no sense to use Eq. (1) to calculate the similarity between the cold start items and old items. Unlike the traditional method of calculating the similarity between each pair of items from the user's historical rating information, we use the item attribute information to calculate the similarity. Calculating the similarity between each pair of items is a key part of our method. Before calculating, we need to generate the item attribute matrix. The items attribute matrix generation algorithm is as follows.

Combined with the item attribute matrix, we adjusted the traditional similarity calculation formula. The adjusted similarity is calculated as follows:

$$sim'(i,j) = \frac{\sum_{k\in A}(a_{ik}-\bar{a}_i)(a_{jk}-\bar{a}_j)}{\sqrt{\sum_{k\in A}(a_{ik}-\bar{a}_i)^2}\sqrt{\sum_{k\in A}(a_{ik}-\bar{a}_j)^2}} \tag{8}$$

In this paper, we take into account the potential features based on user attributes. For a cold start item, we use the following formula to make a preliminary prediction of the new item's rating.

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{j\in N(j)} sim'(i,j)\times(r_{uj}-\bar{r}_u)}{\sum_{j\in N(j)} sim(i,j)} \tag{9}$$

Through experimental demonstration, we found that the experimental results did not have any good effect after the data increased. With the passage of time, it is possible that the user's tastes and preferences have changed. By adding the time penalty factor in the above formula, the prediction accuracy can be improved. The time penalty factor is calculated as follows:

$$w_{ij} = 1 - \frac{t_i - t_j}{t_i - t_{min}} \tag{10}$$

where t_i is the new item release time, t_j is item j release time, and t_{min} is the earliest release of the item. We add the time penalty factor to the similarity measure. The updated similarity prediction formula is as follows:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{j\in N(i)} w_{ij} sim'(i,j)\times(r_{uj}-\bar{r}_u)}{\sum_{j\in N(i)} sim(i,j)} \tag{11}$$

Set the following:

$$r' = \frac{\sum_{j\in N(i)} w_{ij} sim'(i,j)\times(r_{uj}-\bar{r}_u)}{\sum_{j\in N(i)} sim(i,j)} \tag{12}$$

It represents the user's preference forecast for the item. The attribute-based KNN prediction algorithm is shown in Algorithm 2.

Algorithm 2 Attribute-Based KNN

Input: *IAmatrix*, *UImatrix*(ratings matrix), K (number of neighbors)
Output: *UImatrix′* (predicted matrix)

```
Initialize UImatrix′, SimilarList, UserPreference
for i ∈ I_new do
    use Eq. (8) get SimilarList
    Neighbor ← sort(SimilarSet)[0:K]
    for n ∈ Neighbor do
        use Eq.(12) get UserPreference
    end for
    use Eq. (11) get r̂_ui
    update(UImatrix′)
end for
return UImatrix′
```

B. STAGE 2: EXISTING ITEMS' RATINGS PREDICTION

As we can see, the main idea of matrix factorization is to reduce the ratings matrix to two low-dimensional matrices. MF is commonly used to capture the interaction between users and items of different rating values. However, most of the changes in the ratings are not only about rating but are more about the user or the item itself. For example, some users tend to give higher ratings than others, while others tend to get higher ratings. Given the effects of these biases, SVD adjusts the predictive formula for ratings and updates the objective function and the error calculation formula:

$$\hat{r}_{ui} = \mu + b_u + b_i + x_u^T y_i \tag{13}$$

$$\min_{x_*, y_*, b_*} \sum\nolimits_{(u,i)\in D} (r_{ui} - \hat{r}_{ui})^2 + \gamma(||x_u||^2 + ||y_i||^2 + b_u^2 + b_i^2) \tag{14}$$

$$e_{ui} = r_{ui} - \mu - b_u - b_i - x_u^T y_i \tag{15}$$

where μ is the training set average rating, and b_u and b_i represent the biases of user u and item i. Currently, the most successful methods to solve the matrix factorization optimization problem are the least square method and the stochastic gradient descent method. A large number of experiments show that when the rating matrix is sparse, the stochastic gradient descent method will get a better effect. Based on the method of the stochastic gradient descent, the update of the related parameters is as follows:

$$\begin{aligned} b_u &\leftarrow b_u + \alpha(e_{ui} - \gamma b_u) \\ b_i &\leftarrow b_i + \alpha(e_{ui} - \gamma b_i) \\ x_u &\leftarrow x_u + \alpha(y_i - \gamma x_u) \\ y_i &\leftarrow y_i + \alpha(x_u - \gamma y_i) \end{aligned} \tag{16}$$

where α represents the learning rate.

C. STAGE 3: EXISTING ITEMS' RATINGS PREDICTION

Since the matrix factorization method ignores the connections between the cold start items and existing items, we added

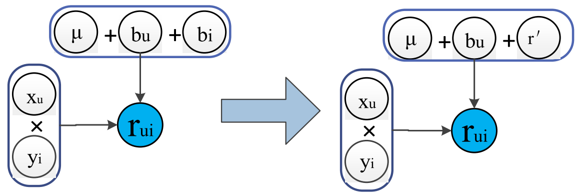

FIGURE 3. Improved matrix factorization mode.

Algorithm 3 Attribute-Fused Matrix Factorization Algorithm

Input: *TrainMatrix*, α (learning rate)
Output: *UImatrix′* (predicted matrix)

```
Initialize b_u, x_u, y_i
repeat
    for r_ui ∈ TrainMatrix do
        use Eq.(14) get r̂_ui
        use Eq.(16) get e_ui
        use Eq.(17) update(b_u, x_u, y_i, α)
    end for
    update(UImatrix′)
until convergence
return UImatrix′
```

the items' attribute information to solve this problem. The modified matrix factorization model is shown in Fig.3.

As seen from the improved model, we combine Eq. (6) with the matrix factorization model. Using the global average and the user bias b_u instead of $\bar{r}_u$ in the KNN calculation formula, the item bias term b_i in the matrix factorization is replaced by the KNN similarity calculation part. After these two operations, we get the final matrix factorization prediction model, the objective function and the error calculation formula.

$$\hat{r}_{ui} = \mu + b_u + r' + x_u^T y_i \tag{17}$$

$$\min_{x_*, y_*, b_*} \sum\nolimits_{(u,i)\in D} (r_{ui} - \hat{r}_{ui})^2 + \gamma(||x_u||^2 + ||y_i||^2 + b_u^2 + r'^2) \tag{18}$$

$$e_{ui} = r_{ui} - \mu - b_u - r' - x_u^T y_i \tag{19}$$

The parameters are updated as follows:

$$\begin{aligned} b_u &\leftarrow b_u + \alpha(e_{ui} - \gamma b_u) \\ x_u &\leftarrow x_u + \alpha(y_i - \gamma x_u) \\ y_i &\leftarrow y_i + \alpha(x_u - \gamma y_i) \\ \alpha &\leftarrow \alpha \times \eta \end{aligned} \tag{20}$$

where η represents the updated coefficient of α, which is a constant.

The parameter updating steps are as shown in Algorithm 3:

Based on the predicted results, this function recommends items with optimal ratings to the active user.

V. TEST DATA AND ANALYSIS

In this section, we verify the availability of our method through a large number of experiments on a real dataset and compare the experience results with the current relatively newer methods and classical methods. Moreover, we also study the influence of different parameters in the method on the accuracy of the experimental prediction.

A. DATA PREPROCESSING

To evaluate the effect of our method on different data scales, we conducted experiments on two datasets, MovieLens 100K, MovieLens 20M and Jingdong Dataset. Table 2 shows information about these datasets.

TABLE 2. Datasets information.

Characteristic	User	Item	Rating
MovieLens 100K	671	9066	100004
MovieLens 20M.	270896	45843	26024289
JingDong	26252	241	61632

MovieLens provides datasets of different sizes with user ratings ranging from 0.5-5.0. The rating time period is from 1995 to 2016. For the MovieLens 100K dataset (dataset 1), it provides discrete ratings on films ranging from 0.5 to 5.0 for 671 users on 9066 items. Just 1.6% of the user/item pairs have a rating value, and the rest of the ratings are unknown. Similarly, MovieLens 20M (dataset 2) provides 270826 users' ratings to 45863 items. For the experiment for solving the cold start problem, we divide the dataset experimentally. For the movies released before 2016, we use it as the training set. After this time, we use the data as a testing set. To solve the data sparsity problem, we vary the training set density from 30 to 80 percent with a step of 10 percent.

The Jingdong dataset is derived from the hot-selling products during the double eleventh period. It contains 26,252 users' ratings on 241 items, with ratings ranging from 1 to 5. Our experiments were conducted on new products released after the double eleven.

B. PREDICTION ACCURACY EVALUATION

The Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are often used in collaborative filtering methods to measure the prediction accuracy. They are defined as formula (21) and formula (22):

$$MAE = \frac{\sum_{ui} |r_{ui} - \hat{r}_{ui}|}{N} \tag{21}$$

$$RMSE = \sqrt{\frac{\sum_{u,i} (r_{ui} - \hat{r}_{ui})^2}{N}} \tag{22}$$

where r_{ui} represents the true rating of user u for item i, $\hat{r}_{ui}$ represents the predicted rating of user u for item i, and N represents the number of predicted values.

The value of MAE is calculated from the average difference between the true rating of the item in the testing set and the predicted rating. The smaller the value is, the higher the prediction accuracy will be. The RMSE value is the square root of the ratio of the square of the deviation of the predicted value from the true value to the number of observations. It is very sensitive to the large or special effects of the prediction and has high requirements on the stability of the experimental method.

C. COMPARATIVE EXPERIMENT

To prove the superiority of our method, we choose the following four methods to test the performance of our experiment on the MAE and RMSE.

(1) User-based mean prediction (UMEANS): This algorithm recommends a cold start item based on historical ratings and predicts by taking the average of historical items. (This is a baseline approach that employs the average rating value observed by a user (*i*.e., the mean of row R) to predict the unknown rating of this user by invoking other unused items.)

(2) Item-based K nearest neighbor algorithm (IKNN): Unlike the traditional method of calculating item similarity from users' historical rating information, we use the item description information to begin the calculation of the similarity. Finally, it is based on the adjusted cosine similarity formula to predict the user's rating of the new item.

(3) Funk-Singular Value Decomposition (Funk-SVD): Funk-SVD is an algorithm published by Simon Funk on his blog. The algorithm is improved from the traditional SVD, and it can effectively solve the large storage space and the high complexity that exist in the traditional SVD.

(4) Seed-based Matrix Factorization (SMF) [24]: This method is proposed by Aleksandrova *et al.*. We choose one of the ways to select seed users in this paper, who provide the largest number of ratings in the system. In addition, then use the singular value decomposition model to predict other users' ratings of new items.

D. EXPERIMENTAL RESULTS AND ANALYSIS

In this part, we present the experimental evaluations of the proposed approach. All the experiments are designed to answer the following questions.

(1) How accurate is the proposed approach for the item cold start problem?

(2) What is the impact of changes in different parameter values on the experimental results?

(3) Compared with other experiments, is our method more advantageous in terms of scalability?

We compare the accuracy of our method with several classical CF methods using experiments. These methods include UMEANS, IKNN, Funk-SVD and SMF. Random initialization is used for all the MF models. The parameters are tuned through standard cross validation. We search through a set of values on the training set to find the optimal one for each method and apply the optimal parameters to the test set.

1) ACCURACY ANALYSIS

In this subsection, we will test the performance of our proposed cold start recommendation algorithm for new items

TABLE 3. Dataset 1 data sparseness experimental results.

Methods	MAE/ *Density*						RMSE/ *Density*					
	0.3	0.4	0.5	0.6	0.7	0.8	0.3	0.4	0.5	0.6	0.7	0.8
UMEANS	1.311	1.181	1.199	1.151	1.105	1.069	1.474	1.452	1.421	1.432	1.405	1.375
IKNN	1.030	1.015	1.010	1.005	0.996	0.994	1.273	1.270	1.269	1.257	1.254	1.250
Funk-SVD	0.971	0.935	0.934	0.930	0.927	0.927	1.101	1.066	1.064	1.068	1.063	1.051
SMF	0.864	0.859	0.842	0.850	0.839	0.825	0.945	0.940	0.924	0.932	0.921	0.911
FASVD	**0.806**	**0.802**	**0.785**	**0.773**	**0.762**	**0.751**	**0.933**	**0.913**	**0.884**	**0.876**	**0.865**	**0.857**

TABLE 4. Dataset 2 data sparseness experimental results.

Methods	MAE/ *Density*						RMSE/ *Density*					
	0.3	0.4	0.5	0.6	0.7	0.8	0.3	0.4	0.5	0.6	0.7	0.8
UMEANS	1.260	1.255	1.209	1.229	1.207	1.195	1.487	1.483	1.429	1.447	1.419	1.414
IKNN	1.013	1.005	0.992	0.985	0.984	0.971	1.291	1.262	1.161	1.159	1.154	1.140
Funk-SVD	1.107	1.104	1.102	1.091	1.091	1.084	1.391	1.387	1.383	1.372	1.366	1.372
SMF	0.955	0.929	0.905	0.867	0.855	0.842	1.180	1.162	1.150	1.114	1.118	1.113
FASVD	**0.874**	**0.866**	**0.845**	**0.841**	**0.835**	**0.822**	**1.103**	**1.104**	**1.076**	**1.067**	**1.053**	**1.040**

through multiple sets of experiments. Verify the accuracy of our experiments by comparing them with several common recommendation algorithms. As shown in Table 3 and Table 4, we compare FASVD with four other well-known approaches, which include UMEANS, IKNN, Funk- SVD and SMF. Table 3 and Table 4 show the MAE and RMSE values of five recommendation approaches on the ratings matrix by using the densities of the training set that range from 0.3 to 0.8 with a step value of 0.1.

Experimental results of Table 2 show the following.

(1) Our method significantly outperforms the other four approaches under both the MAE and RMSE metrics consistently, thereby indicating that the prediction accuracy can be improved by employing items' attributions instead of ratings for cold start recommendations.

(2) The MAE and RMSE values of the five methods decrease with the increase of the *density* from 0.3 to 0.8, which shows that a larger training set provides more information for the prediction.

(3) It can be found that the larger dataset 2 prediction results are worse than dataset 1. This is because dataset 2 contains more users and items. Although the dataset has more ratings, actually, the rating matrix of dataset 2 is sparser than dataset 1.

2) IMPACT OF TOP-K

In our method, the parameter *Top-K* is employed to control the number of similar items. The number of neighbors is an important factor influencing the prediction performance.

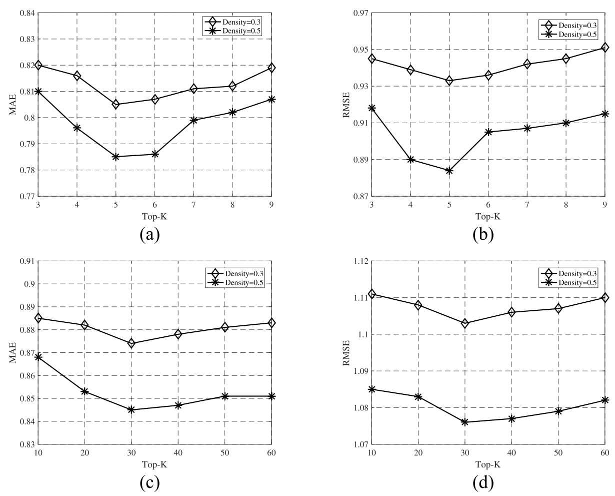

FIGURE 4. Impact of *Top-K* on the prediction accuracy. (a) The MAE of dataset 1. (b) The RMSE of dataset1. (c) The MAE of dataset 2. (d) The RMSE of dataset1.

Fig. 4 illustrates the impact of *Top-K* on the prediction accuracy with *density*=0.3 and *density*=0.5. We set $\alpha = 0.01$ and $\gamma = 1000$. For dataset 1, we vary the*Top-K* from 3 to 9 with a step increase of 1. For dataset 2, we vary the *Top-K* from 10 to 60 with a step increase of 10.

Figs. 4 (a) and (b) show the results on dataset 1 and (c) and (d) shows the experimental results on dataset 2. When the *density*=0.3, it can be seen from Fig. 4 that with the increase of *Top-K*, the MAE and RMSE values become increasingly smaller. Moreover, for dataset 1, our method obtains the optimal performance when *Top-K*=5. For dataset 2, our method obtains the optimal performance when *Top-K*=30. Additionally, when the *density*=0.5, the trends of the MAE and RMSE values are consistent with the result of *density*=0.3. The experimental results show that no matter whether the value of *Top-K* is too large or too small, the accuracy of the prediction will be reduced. If the value of *Top-K* is too small, then we cannot fully consider the relationship between the items. If the *Top-K* value is too large, it will cause too many irrelevant items to join the similarity calculation.

3) IMPACT OF TRAINING SET DENSITY

To study the impact of the training set *density*, we set $\alpha = 0.01$, $\gamma = 1000$, *Top-K*=5 in dataset 1 and *Top-K*=30 in dataset 2. We vary the *density* from 30% to 80% with a step of 10%. Fig. 5 shows the experimental results with the MAE and RMSE in different datasets. Figs. 4 (a) and (b) show the results on dataset 1 and (c) and (d) shows the experimental results on dataset 2.

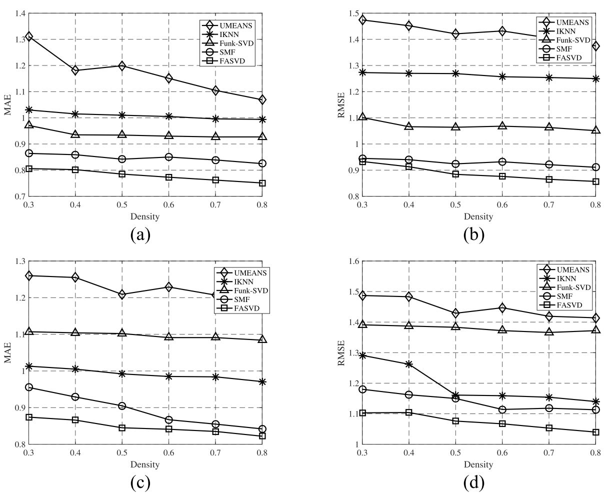

FIGURE 5. **Impact of data *density* on the prediction accuracy. (a) The MAE of dataset 1. (b) The RMSE of dataset1. (c) The MAE of dataset 2. (d) The RMSE of dataset1.**

As we can see from Fig. 5, the five methods mentioned above experience a downward trend with the increase of the *density* from 0.3 to 0.8, thus indicating that more ratings information can enhance the prediction accuracy. This observation shows that larger training set *density* can provide more information to enhance the prediction accuracy.

4) IMPACT OF α

The parameter α is the learning rate of our model, which determines our method's convergence speed. To evaluate the impact of α on the prediction performance, we conducted a set of experiments using dataset 1 and dataset 2 with the densities 0.3 and 0.5. We vary α from 0.0001 to 10 and set $\gamma = 1000$, *Top-K*=5 in dataset 1 and *Top-K*=30 in dataset 2. The experimental results on dataset 1 and dataset 2 are shown in Fig. 6.

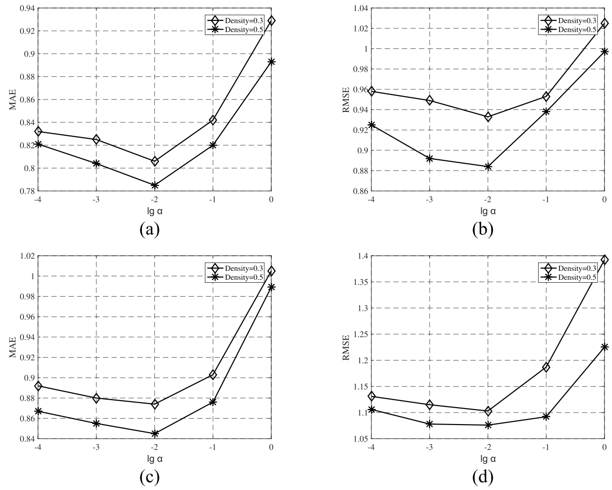

FIGURE 6. **Impact of α on the prediction accuracy. (a) The MAE of dataset 1. (b) The RMSE of dataset1. (c) The MAE of dataset 2. (d) The RMSE of dataset1.**

Fig. 6 shows the impacts of α on the MAE and RMSE in the FASVD model. We can find that with the same training ratio as 0.3 or 0.5, the performances of our method on the MAE and RMSE have very similar trends. As we can see, when the parameter α increases from 0.0001 to 10, the MAE and RMSE values decreased at first and then increased. When $\alpha = 0.01$, the MAE and RMSE values are the smallest, which means the best prediction performance. If the values of α is too small, the model has not completely converged at the end of the iterations. If the values of α is too large, the experiment may jump out of the optimal results.

5) IMPACT OF γ

The parameter γ controls the how much our proposed approach relies on the items' attributes-based regularization term. We conducted a set of experiments to evaluate the impact of γ on the prediction performance with the densities 0.3 and 0.5. We vary γ from 0.1 to 10000 and set $\alpha = 0.01$, *Top-K*=5 in dataset 1 and *Top-K*=30 in dataset 2. The experimental results are shown in Fig. 7.

As we can see from Fig. 7, if γ is too large or too small, the prediction accuracy will be unsatisfactory. When γ is too small, the influence of the attribute information in the model is too low, thus leading to a decrease in the prediction accuracy. When γ is too large, the attribute information is excessively fitted with the training set, which has a bad effect on the predictive performance. When $\gamma = 1000$, our method obtains the optimal performance.

6) SCALABILITY VERIFICATION

To determine the portability of our approach, we also tested our method on the JD dataset. We divided the items released before November 11 and the purchase records of these items into training sets, and divided the score records of the released products into test sets. Through experimental verification,

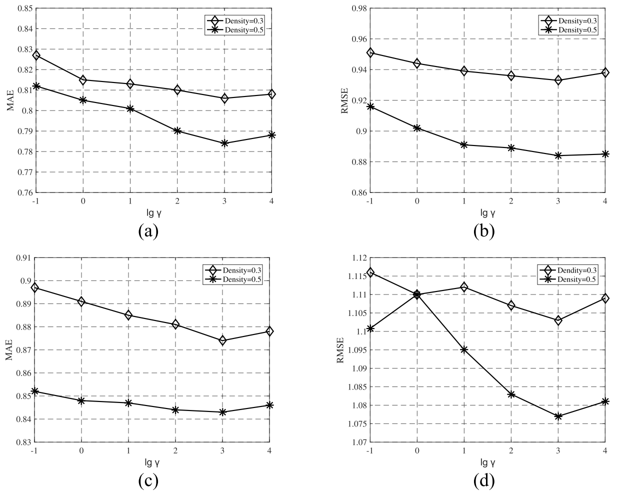

FIGURE 7. Impact of γ on the prediction accuracy. (a) The MAE of dataset 1. (b) The RMSE of dataset1. (c) The MAE of dataset 2. (d) The RMSE of dataset1.

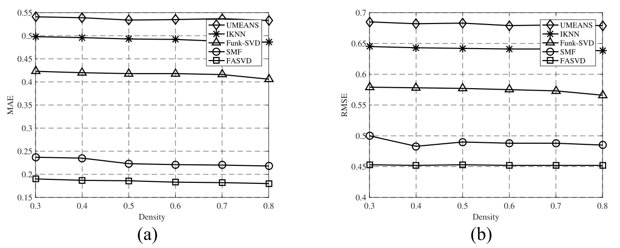

FIGURE 8. Scalability verification. (a) The MAE of JD dataset. (b) The RMSE of JD dataset.

we obtained the prediction results under different density test sets. The experimental results are shown in Figure 8.

The experimental results in Fig.8 are analyzed as follows.

(1) Since users on JD are more inclined to rate high scores, the results obtained by the five methods are significantly better than the first two data sets. At the same time, on the dataset, the predictions of our method have been further improved, indicating that our method is more scalable.

(2) Compared with other methods, our method has more accurate prediction results on JD dataset and the prediction results are more stable than other methods. There is no doubt that our approach can maintain good performance on cold start recommendations in various application scenarios.

VI. CONCLUSION

Currently, the recommendation of a cold start item is still an open question, and the recommendation of a cold start item is still a major challenge for the recommendation system. In this paper, we propose a singular value decomposition method of blending attributes that combines item attributes and historical rating information for cold start recommendations. Our method calculates the similarity between items through attribute information, and then combines the calculation result with the matrix decomposition model to improve the prediction accuracy. During the experimental verification process, we found that the similar weights between the long-term item and the new item should be reduced, thus increasing the time penalty factor. Experimental results of MovieLens and JD datasets show that the proposed method has better performance in prediction accuracy and stability, and our method has good scalability.

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[2] Y. Wang, Q. He, and Y. Yang, "QoS-aware service recommendation for multi-tenant SaaS on the cloud," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2015, pp. 178–185.

[3] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for Web service recommendation and selection," *Expert Syst. Appl.*, vol. 53, pp. 75–86, Jul. 2016.

[4] K. Su, B. Xiao, B. Liu, H. Zhang, and Z. Zhang, "TAP: A personalized trust-aware QoS prediction approach for Web service recommendation," *Knowl.-Based Syst.*, vol. 115, pp. 55–65, Jul. 2017.

[5] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *Proc. Int. Conf. Algorithmic Appl. Manage.*, 2008, pp. 337–348.

[6] A. Töscher, M. Jahrer, and R. M. Bell, "The bigchaos solution to the Netflix grand prize," *Netflix Prize Document*, pp. 1–52, Jan. 2009. [Online]. Available: http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf

[7] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[8] S. Deng, L. Huang, D. Hu, J. L. Zhao, and Z. Wu, "Mobility-enabled service selection for composite services," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 394–407, May/Jun. 2016.

[9] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.

[10] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 230–237.

[11] A. M. Acilar and A. Arslan, "A collaborative filtering method based on artificial immune network," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8324–8332, 2009.

[12] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2009, pp. 437–444.

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

[14] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[15] J. Xu, Y. Yao, H. Tong, X. Tao, and J. Lu, "RaPare: A generic strategy for cold-start rating prediction problem," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1296–1309, Jun. 2017.

[16] T. Liang, L. Chen, X. Ying, S. Y. Philip, J. Wu, and Z. Zheng, "Mobile application rating prediction via feature-oriented matrix factorization," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 261–268.

[17] N. Chen, S. C. Hoi, S. Li, and X. Xiao, "SimApp: A framework for detecting similar mobile applications by online kernel learning," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 305–314.

[18] J. Liu, M. Tang, Z. Zheng, X. F. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for Web service recommendation," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 686–699, Sep. 2016.

[19] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 501–508.

[20] Z. Zhou, Z. Cheng, L. Zhang, W. Gaaloul, and K. Ning, "Scientific workflow clustering and recommendation leveraging layer hierarchical analysis," *IEEE Trans. Services Comput.*, vol. 11, no. 1, pp. 169–183, Jan. 2018.

[21] Y. W. Zhang, Y. Y. Zhou, F. T. Wang, Z. Sun, and Q. He, "Service recommendation based on quotient space granularity analysis and covering algorithm on Spark," *Knowl.-Based Syst.*, vol. 147, pp. 25–35, May 2018.
[22] G. Xue, C. Lin, Q. Yang, and W. Xi, "Scalable collaborative filtering using cluster-based smoothing," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2005, pp. 114–121.
[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[24] M. Aleksandrova, A. Brun, A. Boyer, and O. Chertov, "Identifying representative users in matrix factorization-based recommender systems: Application to solving the content-less new item cold-start problem," *J. Intell. Inf. Syst.*, vol. 48, pp. 365–397, Apr. 2017.
[25] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2616–2624, Nov. 2017.
[26] S. Deng, H. Wu, J. Taheri, A. Y. Zomaya, and Z. Wu, "Cost performance driven service mashup: A developer perspective," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2234–2247, Aug. 2016.
[27] S. Deng *et al.*, "Toward risk reduction for mobile service composition," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1807–1816, Aug. 2016.
[28] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017.
[29] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*, 2nd ed. Berlin, Germany: Springer, 2015, pp. 1–34, doi: 10.1007/978-1-4899-7637-6_1.
[30] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, 2009.
[31] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
[32] S. Balakrishnan and S. Chopra, "Collaborative ranking," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 143–152.
[33] D. Agarwal and B. C. Chen, "fLDA: Matrix factorization through latent Dirichlet allocation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 91–100.
[34] C. Chen, X. Zheng, Y. Wang, F. Hong, and Z. Lin, "Context-aware collaborative topic regression with social matrix factorization for recommender systems," in *Proc. AAAI*, 2014, pp. 9–15.
[35] J. Wu, L. Chen, Q. Yu, P. Han, and Z. Wu, "Trust-aware media recommendation in heterogeneous social networks," *World Wide Web*, vol. 18, no. 1, pp. 139–157, 2015.
[36] C. Shi, J. Liu, F. Zhuang, S. Y. Philip, and B. Wu, "Integrating heterogeneous information via flexible regularization framework for recommendation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 835–859, 2016.
[37] Z. Xu, F. Zhang, W. Wang, H. Liu, and X. Kong, "Exploiting trust and usage context for cross-domain recommendation," *IEEE Access*, vol. 4, pp. 2398–2407, 2016.
[38] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[39] T. Li, A. Liu, and C. Huang, "A similarity scenario-based recommendation model with small disturbances for unknown items in social networks," *IEEE Access*, vol. 4, pp. 9251–9272, 2016.
[40] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," in *Proc. 5th Int. Conf. Comput. Inf. Sci.*, 2002, pp. 27–28.
[41] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Investigation of various matrix factorization methods for large recommender systems," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2008, pp. 553–562.
[42] S. Rendle and L. Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proc. ACM Conf. Recommender Syst.*, 2008, pp. 251–258.

XING GUO received the master's degree in computer science, and the Ph.D. degree from Anhui University, China, in 2009 and 2013, respectively, where he is currently a Lecturer with the School of Computer Science and Technology. His research interests include computer vision, image processing, service computing, and cloud computing.

SHI-CHAO YIN received the bachelor's degree from Anhui University, in 2016, where he is currently pursuing the master's degree with the School of Computer Science and Technology. His current research interests include service computing and cloud computing.

YI-WEN ZHANG received the Ph.D. degree in management science and engineering from the Hefei University of Technology, in 2013. He is currently a Professor with the School of Computer Science and Technology, Anhui University. His research interests include service computing, cloud computing, and e-commerce.

WEI LI received the Ph.D. degree in computer science from Anhui University, in 2006, where she is currently a Professor with the School of Computer Science and Technology. Her research interests include computer vision and embedding system.

QIANG HE received the first Ph.D. degree from the Swinburne University of Technology, Australia, in 2009, and the second Ph.D. degree in computer science and engineering from the Huazhong University of Science and Technology, China, in 2010. He is currently a Senior Lecturer with the Swinburne University of Technology.