IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 4, 2019, accepted August 15, 2019, date of publication August 21, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936630

Collaborative Filtering Based on Gaussian Mixture Model and Improved Jaccard Similarity

HANGYU YAN AND YAN TANG
School of Computer and Information Science, Southwest University, Chongqing 400715, China
Corresponding author: Yan Tang (ytang@swu.edu.com)

ABSTRACT The recommender systems play an important role in our lives, since it can quickly help users find what they are interested in. Collaborative filtering has become one of the most widely used algorithms in recommender systems due to its simplicity and efficiency. However, when the user's rating data is sparse, the accuracy of the collaborative filtering algorithm for predictive rating is badly reduced. In addition, the similarity calculation method is another important factor that affects the accuracy of the collaborative filtering algorithm recommendation. Faced with these problems, we propose a new collaborative filtering algorithm which based on Gaussian mixture model and improved Jaccard similarity. The proposed model uses Gaussian mixture model to cluster users and items respectively and extracts new features to build a new interaction matrix, which effectively solves the impact of rating data sparsity on collaborative filtering algorithms. Meanwhile, a new similarity calculation method is proposed, which is combined by triangle similarity and Jaccard similarity. Compare our proposed model with four models based on collaborative filtering algorithms on three public datasets. The experimental results show that the proposed model not only mitigates the sparseness of the data, but also improves the accuracy of the rating prediction.

INDEX TERMS Recommender systems, collaborative filtering, clustering, Gaussian mixture model, Jaccard similarity.

I. INTRODUCTION

The rapid development of the Internet, in recent years, has brought tremendous changes in people's lives. On the one hand, the Internet brings a huge amount of information to users, which satisfies the user's demand for information; On the other hand, makes it more difficult for users to obtain the information they need in front of a large amount of information, resulting in information overloads [1]–[3]. Recommender systems can be an effective way to solve this problem without requiring the user to provide explicit requirements [4], [5]. Instead, the user's behavior is modeled by analyzing the user's behavior, which actively recommend information on users that can meet their interests and needs [6], [7]. How to design an effective recommendation algorithm has become the focus of research.

Collaborative filtering algorithm is widely used because of its simplicity [4], [8], high efficiency in recommender systems. Collaborative filtering can be divided into user-based collaborative filtering and item-based collaborative filtering [9]–[12]. Collaborative filtering (CF) algorithm constructs similarity matrix to predict target ratings by finding user sets or item sets similar to target users or items. As we all know, with the increasing number of users and items, user-item rating matrix will become increasingly large [13]. However, the user's rating data only accounts for a small part of it, causing the rating data to be sparse [14]. In this case, the CF algorithm will face a series of problems such as sparse rating data, real-time performance, and scalability [15]. How to get the user's accurate rating of the item under the sparse rating data becomes a research hotspot. Clustering technology is an important data preprocessing method in the field of data mining [16]–[19], which try to find its distribution status or mode in unlabeled datasets. It is also widely used in many fields [20]–[22], such as machine learning [23], pattern recognition [24], image processing [25], information retrieval and so on [26]. Combining clustering techniques with CF to eliminate the impact of data sparse on CF has been used in many literatures [27]–[31]. For example, Deng et al. proposed a novel K-medoids clustering recommendation algorithm based on probability distribution to improve the accuracy of the recommendation by improving the Kullback Leibler (KL)

The associate editor coordinating the review of this article and approving it for publication was Le Hoang Son.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

divergence and maximizing the distance [27]. Moradi et al. proposed a collaborative filtering algorithm that uses novel graph clustering to recommend invisible items to users [28]. Nilashi et al. combine clustering algorithms with CF to reduce the sparsity and scalability of recommender systems using dimensionality reduction and ontology techniques [29]. Although the algorithm proposed by Nilashi *et al.* has a good effect, the method used by it is more complicated. In the CF, the similarity calculation method which is used to find similar users or similar items plays an important role in predicting the rating [32]–[37], so a lot of research on similarity is generated. For example, Qian *et al.* proposed an improved similarity CF model, which considers three similarity impact factors, makes full use of the rating data and minimizes the bias of the similarity calculation [32]. Uncertainty exists in recommend systems [38]–[40]. As a result, it is necessary to handle uncertainty in recommend systems [41]. Evidence theory [42]–[46] is widely used in recommendation systems [47], [48]. Based on the construction of users preference with basic probability assignment in evidence theory [49], [50], Yin et al. propose the concept of transfer similarity to measure potential high similarity users or items [51]. Sun *et al.* proposed triangle similarity is combined with Jaccard similarity to measure the similarity between users or items [36]. However, these improved similarity calculation methods are not combined with clustering algorithms to optimize CF.

In order to deal with the above two factors affecting the accuracy of collaborative filtering algorithm, we propose a collaborative filtering algorithm based on Gaussian mixture model and improved Jaccard similarity. The main contributions are as follows:

- Combine the Gaussian mixture model (GMM) with the collaborative filtering algorithm. The Gaussian mixture model is used to cluster the user item rating matrix, and then feature extraction is performed to construct a new user item interaction matrix. The new interaction matrix mitigates the sparsity of the original matrix rating data.
- Combining Jaccard similarity and triangle similarity, a new similarity calculation method is proposed to improve the accuracy of the rating.
- Combine the proposed similarity calculation method with GMM clustering to improve the accuracy of rating prediction.

The organization of this paper is as follows. Preliminaries were introduced in Section 2. In Section 3, the architecture and details of the proposed model are given. In Section 4, we give experimental results on three public datasets. In the concluding remarks, a summary of the contributions to this paper is given.

II. PRELIMINARIES

In this section, we introduce some preliminary knowledge that needs to be used. The following is a detailed introduction of them.

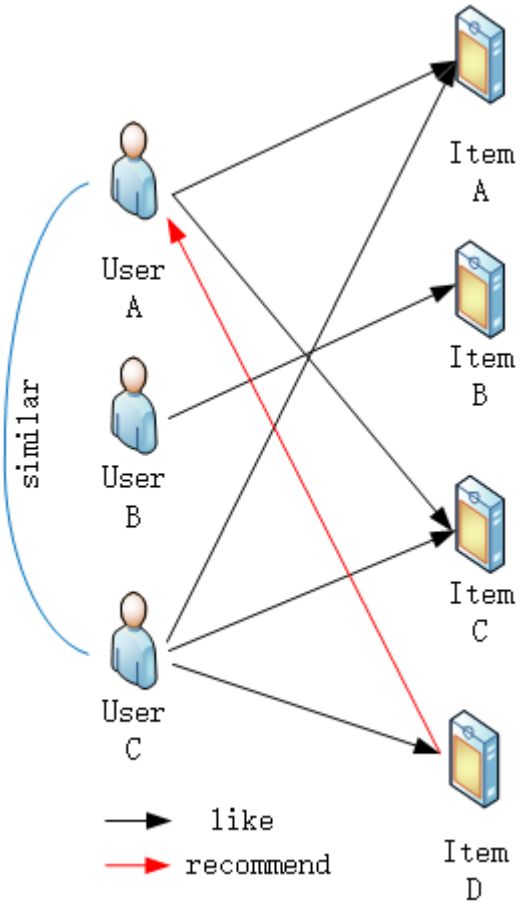


FIGURE 1. The illustration for user-based CF.

A. COLLABORATIVE FILTERING

The CF algorithm is the earliest and well-known recommendation algorithm [52], [53]. The algorithm searches for user preferences by mining user historical behavior data, which can be divided into user-based CF and item-based CF.

1) USER-BASED CF

Suppose that m users $U_M = \{u_1, u_2, u_3, u_4, \ldots, u_M\}$ rate on n items $I_N = \{i_1, i_2, i_3, i_4, \ldots, i_N\}$ so that they can be described using a $M \times N$ matrix. User-based CF is to search for items by rating similar users [9]. The idea is as shown in Figure 1. Both user A and user C like item A and item C. It can be seen that user A and user C share a common hobby. User C also likes item D, so that user C is used to predict user A's preference for item D and user A is recommended.

2) ITEM-BASED CF

The principle of item-based CF is similar to user-based CF, except that the item itself is used in the calculation of neighbors, rather than from the user's point of view, that is, based on the user's preference for the item to find similar items and then recommend similar based on the user's historical preferences [54]. That is, all users' preferences for an item are

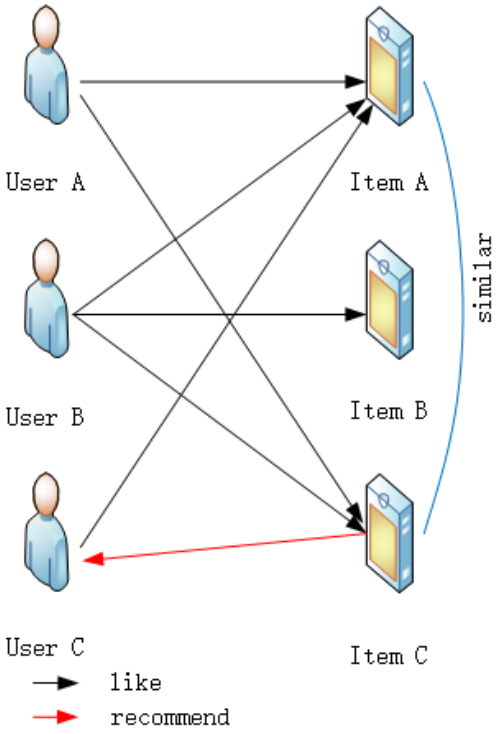


FIGURE 2. The illustration for Item-based CF.

used as a vector to calculate the similarity between the items and after obtaining the similar items of the items, the items of the items that have not been touched are recommended to the current user according to the preference of the user history. As shown in Figure2, users A and B like items A and C, when user C likes item A, it can be inferred that user C also likes item C.

B. POPULAR SIMILARITIES

In the CF algorithm, the similarity calculation method affects the accuracy of the rating. Some similarity calculation methods widely used in the algorithm are given below:

cosin: Cosine similarity which measures the two vectors by calculating the cosine of the angle between the two vectors [55], is defined as

$$cosin\left(\vec{i},\vec{j}\right)=\frac{\vec{i}\cdot\vec{j}}{\left|\vec{i}\right|\times\left|\vec{j}\right|} \tag{1}$$

whose value range is $[-1, 1]$, where i, j are rating vectors for different users or items.

Pearson: The Pearson correlation coefficient reflects the degree of linear correlation between two vectors [56]–[58], which is defined as

$$p\left(i,j\right)=\frac{\sum_{r\epsilon i,j}\left(R_{i,r}-\bar{R}_i\right)\left(R_{j,r}-\bar{R}_j\right)}{\sqrt{\sum_{r\epsilon i,j}\left(R_{i,r}-\bar{R}_i\right)^2}\sqrt{\sum_{r\epsilon i,j}\left(R_{j,r}-\bar{R}_j\right)^2}} \tag{2}$$

whose is between $[-1, 1]$, where r is the intersection of the non-zero parts of the vector i, j, $\bar{R}_i$ and $\bar{R}_j$ are average of the vector i, j.

Jaccard: The Jaccard similarity coefficient is used to compare similarities and differences between finite sample sets [59], which is defined as

$$J\left(A,B\right)=\frac{\left|A\bigcap B\right|}{\left|A\bigcup B\right|}=\frac{\left|A\bigcap B\right|}{|A|+|B|-\left|A\bigcap B\right|} \tag{3}$$

where A and B are two different set.

C. GMM CLUSTERING

Gaussian mixture model refers to the linear combination of multiple Gaussian distribution functions. The GMM can fit any type of distribution, which is usually used to solve the case where the data in the same set contains multiple different distributions [37], [60], [61]. Gaussian mixture distribution is defined as

$$p\left(x\right)=\sum_{i=1}^{k}\alpha_i\cdot N\left(x|\mu_i,\Sigma_i\right) \tag{4}$$

As can be seen from equation 4, $N\left(x|\mu_i,\Sigma_i\right)$ is called the *ith* component of the hybrid model, which is a probability density function of the n dimensional random vector x obeying Gaussian distribution and can be defined as

$$N(x)=\frac{1}{(2\pi)^{\frac{n}{2}}\left|\sum\right|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)} \tag{5}$$

where μ is the mean vector, Σ is the covariance matrix. In equation 4, α_i can be regarded as the *ith* selected probability, which satisfies the following condition

$$\sum\nolimits_{i=1}^{k}\alpha_i=1 \tag{6}$$

Assume that a sample set $D=\{x_1,x_2,x_3,\ldots,x_m\}$ generation process is given by Gaussian distribution mixture distribution, we use the random variable $z_j\epsilon\{1,2,3,\ldots,k\}$ to represent the mixed component of the generated sample x_j, whose value is unknown. It can be seen that the prior probability $P(z_j=i)$ of z_j corresponds to $\alpha_i(i=1,2,3,\ldots,k)$. According to Bayes' theorem [62], we can get the posterior probability of z_j which is defined as

$$\begin{aligned}p\left(z_j=i|x_j\right)&=\frac{P\left(z_j=i\right)\cdot p(x_j|z_j=i)}{p(x_j)}\\&=\frac{\alpha_i\cdot N\left(x_j|\mu_i,\Sigma_i\right)}{\sum_{l=1}^{k}\alpha_l\cdot N\left(x_j|\mu_l,\Sigma_l\right)}\end{aligned} \tag{7}$$

In equation 7, $p\left(z_j=i|x_j\right)$ represents the posterior probability of sample x_j generated by the *ith* Gaussian mixture. We use $\gamma_{ji}(i=1,2,3,\ldots,k)$ to represent $p\left(z_j=i|x_j\right)$. When the model parameters $\{(\alpha_i,\mu_i,\Sigma_i)\,|1\le i\le k\}$ in equation 4 are known, the GMM cluster divides the sample set D into k clusters $C=\{C_1,C_2,C_3,\ldots,C_k\}$ [60], and the cluster label λ_j of each sample x_j can be determined according to the equation 8

$$\lambda_j=\underset{i\epsilon(1,2,3,\ldots,k)}{\arg\max}\ \gamma_{ji} \tag{8}$$

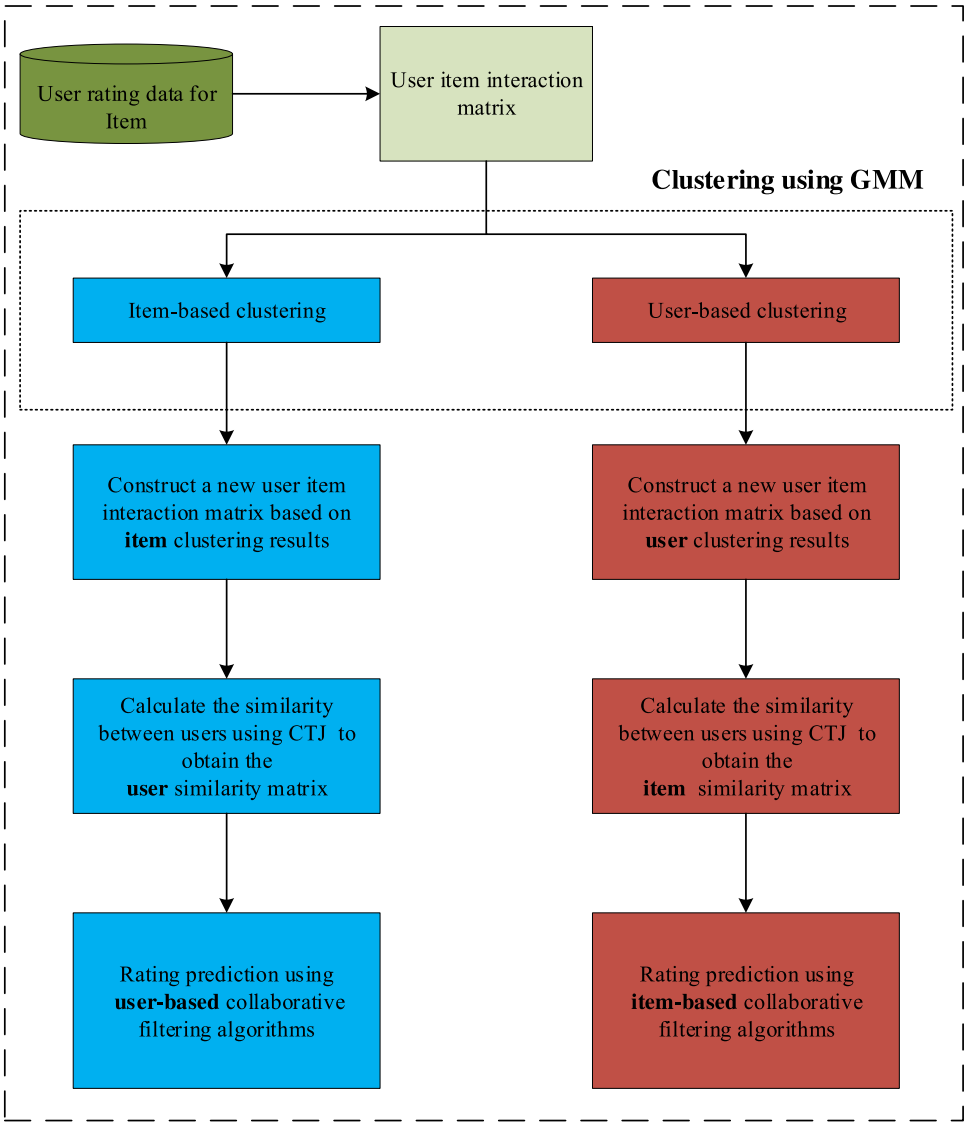


FIGURE 3. The proposed model framework.

we get the cluster label λ_j to which x_j belongs and divide x_j into cluster C_{λ_j}. The model parameters $\{(\alpha_i, \mu_i, \Sigma_i) \mid 1 \leq i \leq k\}$ is solved using the EM algorithm [63].

III. PROPOSED MODEL

In the rating matrix, the user's rating of items only accounts for a small part of the rating matrix, which results in the sparsity of the rating matrix. Our model uses GMM to cluster the rating matrix, and then constructs a new interaction matrix to alleviate the sparsity of the rating matrix. The improved similarity formula is used to calculate the similarity. Users and items are clustered separately during the clustering process. So the model is divided into two parts, one is to cluster the users, the other is to cluster the items, and then construct a new user-item interaction matrix. Finally, using CF algorithm for rating prediction. The proposed model is shown in Figure 3. The process of the model will be explained in detail in the following subsections.

A. CLUSTERING AND FEATURE EXTRACTION

There are two methods for highly sparse data processing, clustering and dimensionality reduction [8], [11], [64]–[66]. Our proposed model uses clustering methods. The goal of clustering using GMM is to construct an appropriate sparse user item interaction matrix based on the highly sparse user item rating matrix, which solves the impact of high sparsity of rating data on CF prediction accuracy. The process of user GMM clustering and the process of item GMM clustering are similar, so only the process of user GMM clustering is introduced in this subsection. An example is given below to illustrate clustering for users. In Figure 4(a), we use GMM to cluster all users in the user item rating matrix, divide the user into corresponding clusters, and we label each user to identify the cluster they belong to. It can be seen that the user is divided into 4 clusters. After the users are clustered, the corresponding features are extracted to construct the user item interaction matrix.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	cluster
U_1	3	5		1			4	C_1
U_2	5		4	2		1		C_2
U_3				5		3		C_1
U_4		2	2		4		1	C_4
U_5			3	3			1	C_4
U_7		3		4			5	C_2
U_8		3			4			C_3

(a)

	C_1	C_2	C_3	C_4
I_1	[U_1, U_3]	[U_2, U_7]	[NULL]	[NULL]
I_2	[U1]	[U7]	[U8]	[U4]
I_3	[NULL]	[U2]	[NULL]	[U4, U5]
I_4	[U1, U5]	[U2, U7]	[NULL]	[U5]
I_5	[NULL]	[NULL]	[U8]	[U4]
I_6	[U3]	[U2]	[NULL]	[NULL]
I_7	[U1]	[U7]	[NULL]	[U4, U5]

(b)

FIGURE 4. Using GMM to extract features.

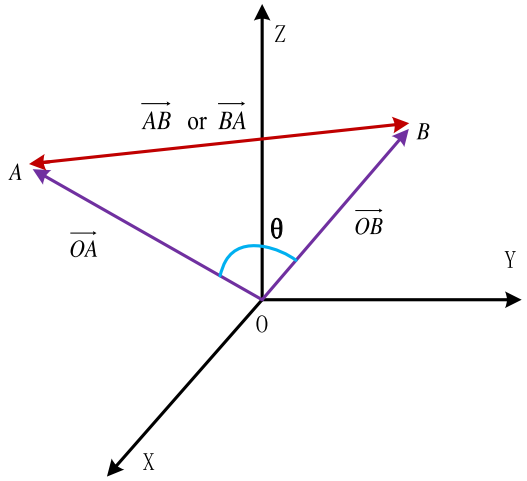


FIGURE 5. The illustration of triangle similarity.

The new matrix is constructed using the item as the row and the user's cluster as the column. The constructed user item interaction matrix is shown in Figure 4(6). In the matrix, each element represents a collection of users belonging to the same cluster and rating the same item. Compared with the original user item rating matrix of Figure 4(a), the new user item interaction matrix of Figure 4(b) solves the problem of too sparse data in the matrix.

B. COMBINE TRIANGLE AND JACCARD SIMILARITIES

As illustrated in Figure 5, assume that $\overrightarrow{OA}$ and $\overrightarrow{OB}$ are the rating vectors of two users or two items, forming a triangle in the space. The triangle similarity is one minus the third divided by the sum of two edges corresponding to the vectors [36], which is defined as

$$Triangle\left(\overrightarrow{OA}, \overrightarrow{OB}\right) = 1 - \frac{|AB|}{|OA| + |OB|} \tag{9}$$

whose value range is [0,1]. We can see that the bigger value of triangle, the more similar they are. Triangle similarity considers not only the angle between two vectors, but also the length of them, so it is more reasonable than the angle based cosine similarity. For instance, give two user rating vectors $\overrightarrow{u_1} = (5, 5, 5)$ and $\overrightarrow{u_2} = (1, 1, 1)$, the cosine similarity is 1, which is not true. However, the triangle similarity between them is 0.33, which eliminates this irrationality.

However, it only considers common rating users or items and does not work well when used alone. Fortunately, the Jaccard similarity performs well in the similarity calculations of no common rating users or items. Therefore, Sun et al. combined the triangle similarity with Jaccard similarity to construct a new similarity. In Sun's paper, it is proved that Jaccard is better than triangle similarity in rating prediction [36]. Based on this, we believe that the Jaccard similarity has a greater impact on the new similarity when the two similarities are combined. Therefore, an improved similarity calculation method is proposed as shown in equation 10:

$$CTJ\ (i, j) = 0.5 \cdot Jaccard\ (i, j) \cdot (triangle\ (i, j) + 1) \tag{10}$$

In equation 10, we combine Jaccard and triangle, not only to optimize the triangle by multiplying two similarities, but also to improve the *CTJ* by combining Jaccard. For example, when the Jaccard similarity of two rating vectors is 0.9 and the triangle similarity is 0.8, the similarity obtained by multiplying the two is 0.72, which is far less than 0.9, so the Jaccard similarity is introduced again in the equation to solve this problem. The proposed CTJ similarity is 0.81. In equation 10, 0.5 is to satisfy the obtained similarity value from 0 to 1, the closer the value of CTJ is to 1, the higher the similarity between the two vectors.

C. RATING PREDICTION

In the previous two subsections, we introduced the GMM clustering process for users or items in the rating matrix and the *CTJ* similarity calculation method. In this subsection, the two methods are combined. As shown in Figure 4(b), *CTJ* is used in the interaction matrix constructed based on user

clustering to calculate the similarity between different two items. The item similarity matrix is obtained by calculation, and then KNN-based CF is used to perform rating prediction on items that are not evaluated by the users [67]–[69]. The prediction value of user i on item j is computed as follows

$$P(u,i) = \bar{r}_i + \frac{\sum_{a\epsilon n}\left(r_{u,a} - \bar{r}_a\right)\cdot sim(a,i)}{\sum_{a\epsilon n}|sim(a,i)|} \tag{11}$$

where n is set of neighbors, a is an element in set, and $sim(a, i)$ is similarity of item a and item i.

Item-based clustering and similarity calculation methods are similar to user-based clustering and similarity calculation methods. After clustering the items, the similarity between different users is calculated in the new interaction matrix. Then use equation 12 to predict the rating [70], the equation is as follows

$$P(u,i) = \bar{r}_u + \frac{\sum_{b\epsilon n}\left(r_{b,i} - \bar{r}_b\right)\cdot sim(b,u)}{\sum_{b\epsilon n}|sim(b,u)|} \tag{12}$$

where n is set of neighbors, b is an element in set, and $sim(b, u)$ is similarity of user b and user u.

In the end, we give the proposed model its algorithm execution process, because the user-based clustering and the item-based clustering algorithms are similar, here only the user-based clustering algorithm process is given. At the same time, GMM and KNN-based CF are well known, so they are used directly in the description of algorithm 1.

Algorithm 1 Framework of Proposed Model

Input:
User-item rating matrix $R_{m\times n}$;
Number of clusters: k;

Output:
Predictive ratings for items that have not been reviewed by users;

GMM is used to cluster the rating vectors of m users, and the vectors are clustered into k clusters;
Label each user with the cluster they belong to;
Initialize the user clustering matrix $M_{n\times k}$;
for item i from $i = 1$ to $i = n$ **do**
  for cluster j from $j = 1$ to $j = k$ **do**
    Find all users who have rated item i;
    Save the corresponding user to the corresponding $M[i, j]$ according to the cluster label to which the user belongs.
  end for
end for
Initialize the item similarity matrix $S_{n\times n}$;
for item i from $i = 1$ to $i = \mathrm{n} - 1$ **do**
  for item j from $j = i + 1$ to $j = \mathrm{n}$ **do**
    Use CTJ to calculate the similarity of item i and item j of matrix $M_{n\times k}$;
    Store the similarity in $S[i, j]$ and $S[j, i]$.
  end for
end for
Obtain a item similarity matrix $S_{n\times n}$ and use $KNN - based\ CF$ for rating prediction.

IV. EXPERIMENTS

In this section, to demonstrate the validity of the proposed model, it will be compared to four known recommendation models on three public datasets. We also validate the impact of GMM clustering and CTJ similarity on the accuracy of rating prediction in three datasets.

A. DESCRIPTION OF THE DATASETS

The three public datasets are MovieLens-100k, MovieLens-1M and Yahoo! Webscope R4. The detailed description of the datasets is as follows

MovieLens-100k: It is a well-known datasets provided by grouplens, which usually used in recommender systems evaluation. The datasets contains nearly 1,000 user ratings for 1,700 movies, with integer ratings ranging from 1 to 5. The data set contains 100,000 anonymous ratings.

MovieLens-1M: This datasets is also provided by gruoplens, which contains 6040 users and 3952 items. users provided integer ratings from 1 to 5, where the higher of ratings, the more users like them. Each user rated at least 20 movies, with an anonymous rating of 1,000,209 in the datasets.

Yahoo! Webscope R4: It was provided by the Yahoo! Research Alliance Webscope program. Users provided integer ratings from 1 to 5 in this datasets. The data set provides two groups Data, training sets and test sets. The former contains 7,642 Users, 11,915 movies and 211,231 ratings.The latter consisted of 2,309 users, 2,380 movies and 10,136 ratings.

TABLE 1. Summaries of datasets.

Dataset	user	item	ratings	scale
MovieLens 100K	943	1682	$\{1, 2, 3, 4, 5\}$	10^5
MovieLens 1M	6040	3952	$\{1, 2, 3, 4, 5\}$	10^6
Yahoo! Webscope R4	7642	11915	{1,2,3,4,5}	10^5

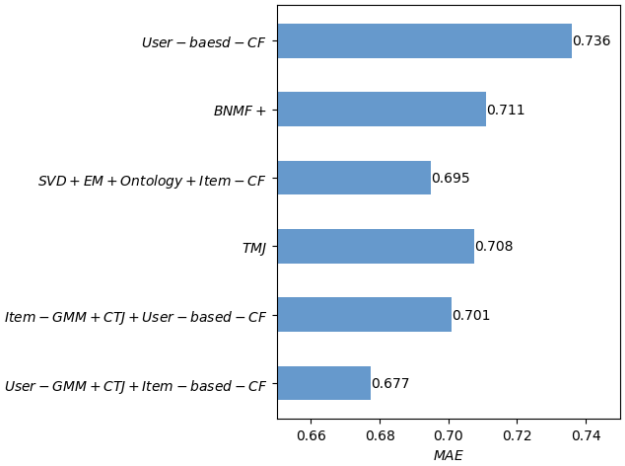
For the experimental verification, we randomly divide the MovieLens-100k and MovieLens-1M datasets into two parts, 80% training sets and 20% test sets. The comparison of the three datasets is shown in Table 1.

B. EVALUATION INDICATOR

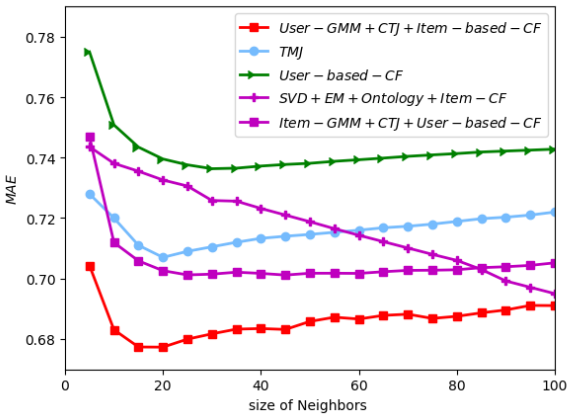
The evaluation indicator of the rating prediction accuracy commonly used, in the recommender systems, is Mean Absolute Error (MAE) that is the average of the absolute errors [71]–[73]. The smaller value of MAE, the closer predicted value to the actual value, which represent more accurate of the prediction. It is defined as

$$MAE = \frac{1}{N}\sum_{1}^{N}\left|r_i - \hat{r}_i\right| \tag{13}$$

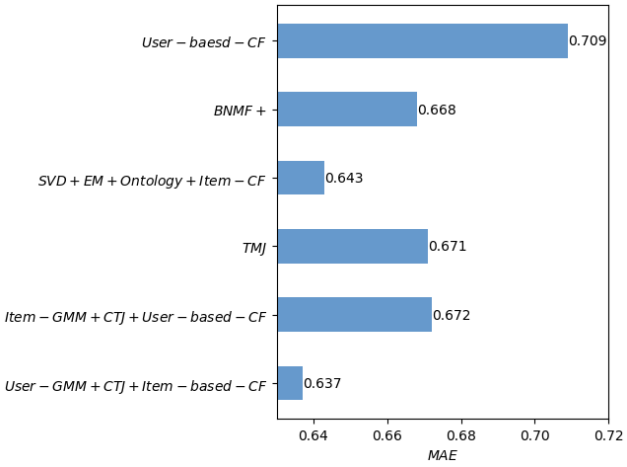
where r_i is the forecast rating of the item, $\hat{r}_i$ is the original rating of the item.



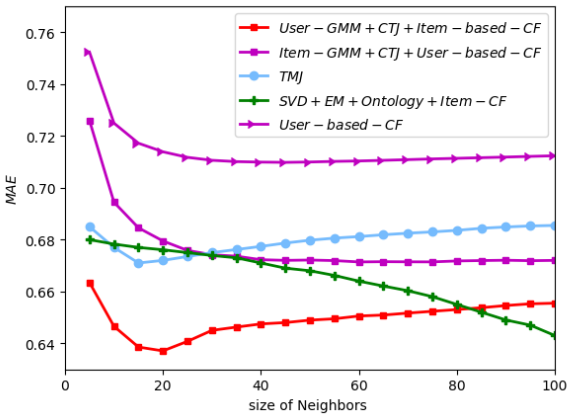
(a) MovieLens 100K



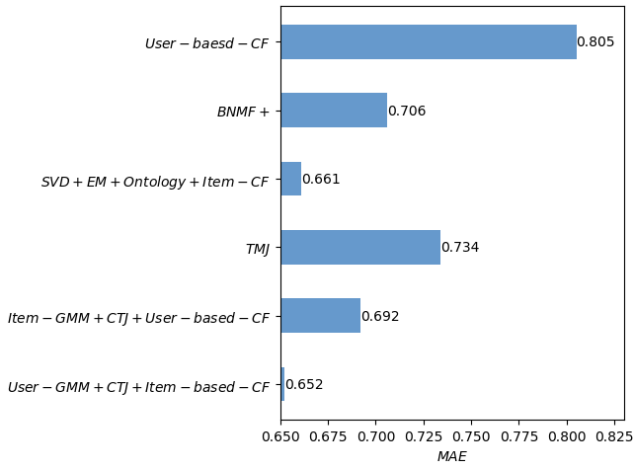
(b) MovieLens 100K



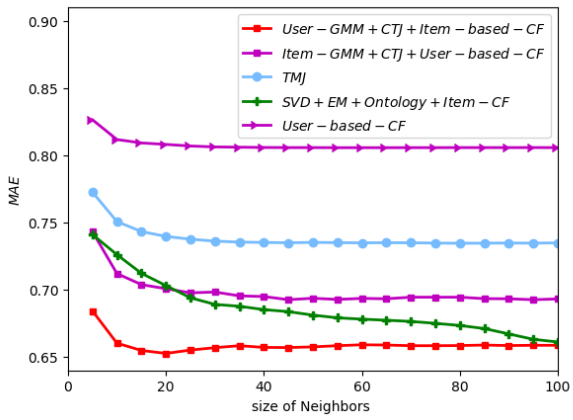
(c) MovieLens 1M



(d) MovieLens 1M



(e) Yahoo! Webscope R4



(f) Yahoo! Webscope R4

FIGURE 6. MAE of several algorithms on three public datasets.

C. EXPERIMENTAL RESULT AND ANALYSIS

In this subsection, experiments will be conducted to verify the proposed Collaborative filtering based on Gaussian mixture model and improved Jaccard similarity. We use $GMM + CTJ + CF$ to refer to the proposed model, and $+$ represents the combination in $GMM + CTJ + CF$. The experiment is divided into three parts. Experiment 1 gives the comparison between $GMM + CTJ + CF$ and the four

TABLE 2. The results of MAE being affected by the number of clusters.

number of clusters	$Item-GMM+CTJ+User-based-CF$			$User-GMM+CTJ+Item-based-CF$		
	MovieLens 100K	MovieLens 1M	Yahoo! Webscope R4	MovieLens 100K	MovieLens 1M	Yahoo! Webscope R4
2	0.701	0.675	0.692	0.677	0.637	0.652
3	0.707	0.671	0.693	0.694	0.640	0.687
4	0.723	0.685	0.761	0.748	0.646	0.702
5	0.765	0.759	0.697	0.737	0.647	0.664
6	0.750	0.674	0.709	0.697	0.647	0.707

recommended algorithms. Experiment 2 is divided into two parts. The first part gives the influence of the number of clusters on the MAE of the $GMM + CTJ + CF$ model, the second part gives the MAE affected by three similarities CTJ, triangle and Jaccard in CF and the influence of the proposed CTJ on the MAE of the $GMM + CTJ + CF$ model. In experiment 3, the results of the GMM clustering of the proposed model without CTJ are given.

1) EXPERIMENT 1

In this experiment, $GMM + CTJ + CF$, $SVD + EM + Ontology + Item - CF$ [29], TMJ [36], $User - based - CF$ and $BNMF+$ were compared on three datasets [30], [74]. $SVD + EM + Ontology + Item - CF$ is a CF algorithm using ontology and dimensionality reduction techniques, TMJ is a CF algorithm based on an improved Jaccard similarity and $BNMF+$ is Bayesian non-negative matrix factorization method to improve the current clustering results in the CF area. $GMM + CTJ + CF$ can be divided into $User - GMM + CTJ + Item - based - CF$ and $Item - GMM + CTJ + User - based - CF$ based on clustering users and clustering items. The result is shown in Figure 6. The MAE of $GMM + CTJ + CF$ and the other four methods on the three datasets is given in (a), (c), (e) of Figure 6. It can be seen that the model we proposed has a good performance on all datasets. Compared with $Uer - based - CF$, $Item - GMM + CTJ + User - based - CF$ has a big improvement on all three datasets, especially with an 11% improvement on Yahoo! Webscope R4. $User - GMM + CTJ + Item - based - CF$ performed best on all three datasets, with 3%-4% improvements over $Item - GMM + CTJ + User - based - CF$, which correspondingly verified that clustering of users to predict ratings is better than clustering items.

In (e), (d), (f) of Figure 6, we give the influence of different size of neighbors on the MAE of various algorithms. Referring to (a), (c), (e) of Figure 6, it can be observed that although the MAE difference between $SVD + EM + Ontology + Item - CF$ and $User - GMM + CTJ + Item - based - CF$ is small, $User - GMM + CTJ + Item - based - CF$ requires only a few neighbors to achieve the best MAE.

2) EXPERIMENT 2

The number of clusters is one of the important factors affecting the accuracy of the ratings of $User - GMM + CTJ + Item - based - CF$ and $Item - GMM + CTJ + User - based - CF$. The first part of the experiment gives MAE values for $User - GMM + CTJ + Item - based - CF$ and $Item - GMM + CTJ + User - based - CF$ in different numbers of clusters on three public datasets. As shown in Table 2, we can see the influence of the number of clusters on the MAE of the two algorithms. When the number of clusters is two, the performance of the two algorithms is the best. With the increasing number of clusters, the performance of the two algorithms decreases in varying degrees. The reason is that as the number of clusters increases, the columns in the user item interaction matrix also increase. At the same time, the number of users or items contained in each element in the matrix is also reduced. The increase in two factors will lead to the gradual sparseness of the matrix and the sparseness of each element in the matrix, which eventually lead to a decrease in the accuracy of the rating prediction.

Similarity is another important factor affecting the rating prediction of this model. In the second part of the experiment, in the sparse rating matrix, we compared the effects of three similarities of CTJ, Jaccard and triangle on MAE in CF at the beginning. Experiments were compared on three public datasets and the results are shown as Figure 7. In (a) of Figure 7, we compare MAE of three similarities in user-based CF. In Figure 7(b), we compare MAE of three similarities in item-based CF. From the figure we can see that the CTJ similarity has only a slight improvement compared to the Jaccard similarity. At the same time, the experiment also verified that the MAE of CTJ and Jaccard similarity is better than the triangle similarity. Secondly, the performance of CTJ and Jaccard in our proposed model will be compared. When using Jaccard as the similarity, we use $User - GMM + Item - based - CF$ stands for GMM clustering of users and $Item - GMM + User - based - CF$ refers to GMM clustering for items. Experiments were also performed on three public datasets and the number of clusters in the experiment is 2. The results of the experiment are shown in Figure 8. It can be observed from the figure that the proposed CTJ is improved compared

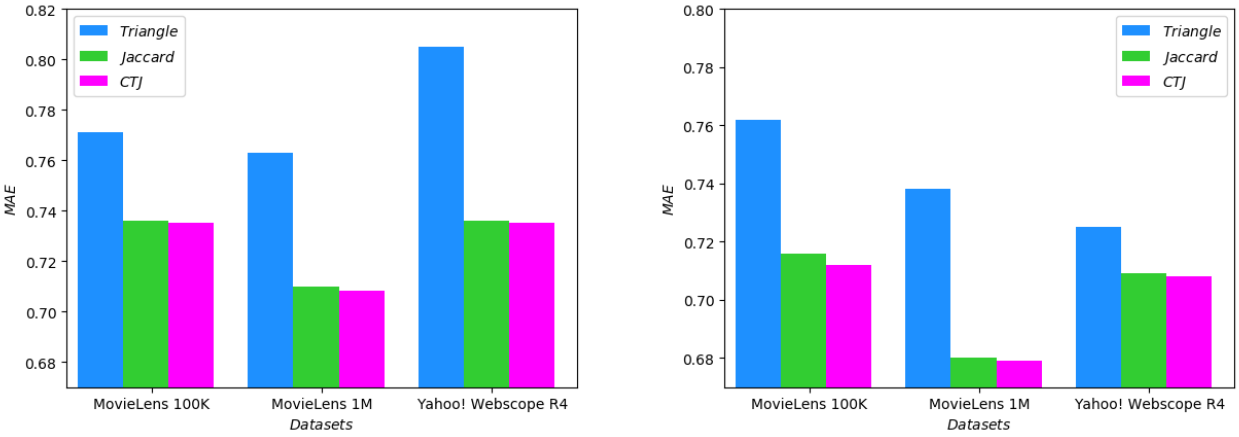


FIGURE 7. Comparison of MAE of three similarities.

TABLE 3. The MAE comparison.

Methods/Datasets	MovieLens 100K	MovieLens 1M	Yahoo! Webscope R4
$User-based\,CF$	0.736	0.709	0.805
$Item-based\,CF$	0.720	0.676	0.794
$Item-GMM+User-based-CF$	0.703	0.670	0.703
$User-GMM+Item-based-CF$	0.685	0.647	0.683

TABLE 4. Sparsity comparison before and after GMM clustering.

Datasets	Sparsity	Methods	Sparsity after clustering
MovieLens 100K	95.2%	$Item-GMM+User-based-CF$	2%
		$User-GMM+Item-based-CF$	2%
MovieLens 1M	95.8%	$Item-GMM+User-based-CF$	1%
		$User-GMM+Item-based-CF$	1%
Yahoo! Webscope R4	99.8%	$Item-GMM+User-based-CF$	5%
		$User-GMM+Item-based-CF$	5%

to the Jaccard on the three public datasets. At the same time, compared with $Item-GMM+CTJ+User-based-CF$, the improvement of $User-GMM+CTJ+Item-based-CF$ is particularly obvious. This also means that the proposed combination of CTJ and $User-GMM+Item-based-CF$ is better than combination of CTJ and $Item-GMM+CTJ+User-based-CF$.

By analyzing Figure 7 and Figure 8, we can conclude that CTJ similarity and Jaccard similarity have almost the same effect on MAE when the rating matrix is particularly sparse. However, when the user item interaction matrix is constructed by GMM clustering, and the data in the matrix is no longer sparse, our proposed CTJ similarity is better than the Jaccard similarity.

3) EXPERIMENT 3

In this experiment, the process of clustering the Gaussian mixture model is verified. The Jaccard similarity is also used

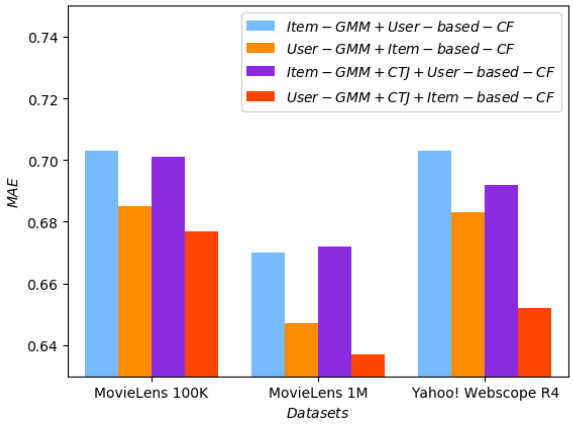


FIGURE 8. MAE is affected by Jaccard or CTJ.

in the algorithms as the calculation method of similarity. Compare them with $User-based\ CF$ and $Item-based\ CF$. The results are shown in Table 3.

As can be seen from Table 3, compared with $user-based\ CF$ and $Item-based\ CF$, our proposed model has greatly improved on MAE. Especially in Yahoo! Webscope R4 datasets, compared with $Item-based\ CF$, $User-GMM+Item-based-CF$ has increased by approximately 11%. This proves the effectiveness of using GMM to cluster the rating matrix.

In Table 4, the changes in the sparsity of the three datasets were verified using $User-GMM+Item-based-CF$ and $Item-GMM+User-based-CF$. The sparsity of the user item rating matrix constructed by the three public datasets and the sparsity of the new user item interaction matrix constructed after clustering using GMM are given in the table. It can be seen that the new matrix constructed by GMM clustering and feature extraction solves the problem of data sparsity.

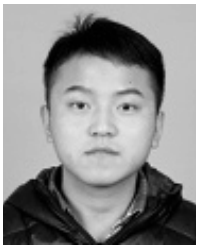
V. CONCLUSION

In this paper, a recommendation model is proposed to eliminate the data sparsity problem in CF, which is called Collaborative filtering based on Gaussian mixture model and improved Jaccard similarity. The proposed model uses GMM to cluster the user item rating matrix to extract corresponding features to construct a new user item interaction matrix to eliminate the impact of data sparseness on rating prediction. Based on Jaccard, a new similarity calculation method is proposed, which combines Jaccard and triangle similarity. In order to verify the prediction effect of the proposed model, the algorithm is compared with four cluster-based collaborative filtering algorithms on the three public datasets of MovieLens 100K, Movielens 1M and Yahoo! Webscope R4. The experiment used MAE as the evaluation standard, which confirmed that the proposed model improved the accuracy of the rating prediction.

REFERENCES

[1] I. Guy, "Social recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 511–543.
[2] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: Issues, challenges, and research opportunities," in *Information Science and Applications*. Singapore: Springer, 2016, pp. 1179–1189.
[3] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus, "Understanding choice overload in recommender systems," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 63–70.
[4] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 77–118.
[5] J. Bobadilla and A. Gutiérrez, F. Ortega, and B. Zhu, "Reliability quality measures for recommender systems," *Inf. Sci.*, vols. 442–443, pp. 145–157, May 2018.
[6] P. Melville and V. Sindhwani, *Recommender Systems*. Boston, MA, USA: Springer, 2017, pp. 1056–1066.
[7] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, Oct. 2012.
[8] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 81–173, Feb. 2010.
[9] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 478–481.
[10] Y. Cai, H.-F. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-based collaborative filtering recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 766–779, Mar. 2014.
[11] L. Fabisiak, "Web service usability analysis based on user preferences," *J. Org. End User Comput. (JOEUC)*, vol. 30, no. 4, pp. 1–13, 2018.
[12] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile Internet applications," *IEEE Access*, vol. 4, pp. 3273–3287, 2016.
[13] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, p. 3, Jul. 2014.
[14] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.-Based Syst.*, vol. 26, pp. 225–238, Feb. 2012.
[15] J. Chen, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering," *Swarm Evol. Comput.*, vol. 38, pp. 35–41, Feb. 2018.
[16] A. Dutt, M. A. Ismail, and T. Herawan, "A systematic review on educational data mining," *IEEE Access*, vol. 5, pp. 15991–16005, 2017.
[17] A. R. Condrobimo, B. S. Abbas, A. Trisetyarso, W. Suparta, and C.-H. Kang, "Data mining technique with cluster anaysis use K-means algorithm for LQ45 index on Indonesia stock exchange," in *Proc. Int. Conf. Inf. Commun. Technol.*, Mar. 2018, pp. 885–888.
[18] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 235–244, Mar. 2015.
[19] G. Manogaran and D. Lopez, "A Gaussian process based big data processing framework in cluster computing environment," *Cluster Comput.*, vol. 21, no. 1, pp. 189–204, 2017.
[20] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[21] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.
[22] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, and A. R. Green, "SC3: Consensus clustering of single-cell RNA-seq data," *Nature Methods*, vol. 14, no. 5, pp. 483–486, 2017.
[23] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3275–3285, Nov. 2016.
[24] Y. Yan, E. Ricci, G. Liu, and N. Sebe, "Egocentric daily activity recognition via multitask clustering," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 2984–2995, Oct. 2015.
[25] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.

[26] B. Mitra and N. Craswell, "An introduction to neural information retrieval," *Found. Trends Inf. Retr.*, vol. 13, no. 1, pp. 1–126, 2018.
[27] J. Deng, J. Guo, and Y. Wang, "A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering," *Knowl.-Based Syst.*, vol. 175, pp. 96–106, Jul. 2019.
[28] P. Moradi, S. Ahmadian, and F. Akhlaghian, "An effective trust-based recommendation method using a novel graph clustering algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 436, pp. 462–481, Oct. 2015.
[29] M. Nilashi, O. Ibrahi, and K. Bagherifard, "A recommender system based on collaborative filtering using Ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.
[30] J. Bobadilla, R. Bojorque, A. H. Esteban, and R. Hurtado, "Recommender systems clustering using Bayesian non negative matrix factorization," *IEEE Access*, vol. 6, pp. 3549–3564, 2018.
[31] A. Salah, N. Rogovschi, and M. Nadif, "A dynamic collaborative filtering system via a weighted clustering approach," *Neurocomputing*, vol. 175, pp. 206–215, Jan. 2016.
[32] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.
[33] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: Algorithms and evaluation," in *Social Information Access: Systems and Technologies*. Cham, Switzerland: Springer, 2018, pp. 344–390.
[34] F. Ortega, D. Rojo, and P. Valdiviezo-Díaz, and L. Raya, "Hybrid collaborative filtering based on users rating behavior," *IEEE Access*, vol. 6, pp. 69582–69591, 2018.
[35] H. Koohi and K. Kiani, "User based collaborative filtering using fuzzy C-means," *Measurement*, vol. 91, pp. 134–139, Sep. 2016.
[36] S.-B. Sun, Z.-H. Zhang, X.-L. Dong, H.-R. Zhang, T.-J. Li, L. Zhang, and F. Min, "Integrating Triangle and Jaccard similarities for recommendation," *PLoS ONE*, vol. 12, no. 8, Aug. 2017, Art. no. e0183570.
[37] Y. Zhang, X. Liu, W. Liu, and C. Zhu, "Hybrid recommender system using semi-supervised clustering based on Gaussian mixture model," in *Proc. Int. Conf. Cyberworlds (CW)*, Sep. 2016, pp. 155–158.
[38] B. Saravanan, V. Mohanraj, and J. Senthilkumar, "A fuzzy entropy technique for dimensionality reduction in recommender systems using deep learning," *Soft Comput.*, vol. 23, no. 8, pp. 2575–2583, Apr. 2019.
[39] P. Singh and R. Agrawal, "A customer centric best connected channel model for heterogeneous and IoT networks," *J. Org. End User Comput. (JOEUC)*, vol. 30, no. 4, pp. 32–50, 2018.
[40] G. Khatwani and P. R. Srivastava, "Impact of information technology on information search channel selection for consumers," *J. Org. End User Comput. (JOEUC)*, vol. 30, no. 3, pp. 63–80, 2018.
[41] V.-D. Nguyen and V.-N. Huynh, "Two-probabilities focused combination in recommender systems," *Int. J. Approx. Reasoning*, vol. 80, pp. 225–238, Jan. 2017.
[42] Y. Dong, J. Zhang, Z. Li, Y. Hu, and Y. Deng, "Combination of evidential sensor reports with distance function and belief entropy in fault diagnosis," *Int. J. Comput. Commun. Control*, vol. 14, no. 3, pp. 329–343, 2019.
[43] H. Cui, Q. Liu, J. Zhang, and B. Kang, "An improved Deng entropy and its application in pattern recognition," *IEEE Access*, vol. 7, pp. 18284–18292, 2019.
[44] Y. Song and Y. Deng, "A new method to measure the divergence in evidential sensor data fusion," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, 2019, Art. no. 1550147719841295.
[45] H. Seiti and A. Hafezalkotob, "Developing pessimistic–optimistic risk-based methods for multi-sensor fusion: An interval-valued evidence theory approach," *Appl. Soft Comput.*, vol. 72, pp. 609–623, Nov. 2018.
[46] Y. Li and Y. Deng, "Generalized ordered propositions fusion based on belief entropy," *Int. J. Comput. Commun. Control*, vol. 13, no. 5, pp. 792–807, 2018.
[47] L. Troiano and L. J. Rodríguez-Muñiz, and I. Díaz, "Discovering user preferences using Dempster–Shafer theory," *Fuzzy Sets Syst.*, vol. 278, pp. 98–117, Nov. 2015.
[48] V.-D. Nguyen and V.-N. Huynh, "Integrating with social network to enhance recommender system based-on Dempster-Shafer theory," in *Proc. Int. Conf. Comput. Social Netw.* Cham, Switzerland: Springer, 2016, pp. 170–181.
[49] R. Sun and Y. Deng, "A new method to identify incomplete frame of discernment in evidence theory," *IEEE Access*, vol. 7, pp. 15547–15555, 2019.
[50] Y. Li and Y. Deng, "TDBF: Two-dimensional belief function," *Int. J. Intell. Syst.*, vol. 34, no. 8, pp. 1968–1982, Aug. 2019.
[51] L. Yin and Y. Deng, "Measuring transferring similarity via local information," *Phys. A, Stat. Mech. Appl.*, vol. 498, pp. 102–115, May 2018.
[52] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, 2009.
[53] F. Ortega, B. Zhu, J. Bobadilla, and A. Hernando, "CF4J: Collaborative filtering for Java," *Knowl.-Based Syst.*, vol. 152, pp. 94–99, Jul. 2018.
[54] B. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, May 2001, pp. 285–295.
[55] L. Muflikhah and B. Baharudin, "Document clustering using concept space and cosine similarity measurement," in *Proc. Int. Conf. Comput. Technol. Develop.*, Nov. 2009, pp. 58–62.
[56] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[57] H. Xu and Y. Deng, "Dependent evidence combination based on Shearman coefficient and pearson coefficient," *IEEE Access*, vol. 6, pp. 11634–11640, 2018.
[58] H. Xu and Y. Deng, "Dependent evidence combination based on DEMATEL method," *Int. J. Intell. Syst.*, vol. 34, no. 7, pp. 1555–1571, 2019.
[59] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. Multiconf. Eng. Comput. Sci.*, 2013, vol. 1, no. 6, pp. 380–384.
[60] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 554–560.
[61] D. Görür and C. E. Rasmussen, "Dirichlet process Gaussian mixture models: Choice of the base distribution," *J. Comput. Sci. Technol.*, vol. 25, no. 4, pp. 653–664, 2010.
[62] D.-S. Lee, J. J. Hull, and B. Erol, "A Bayesian framework for Gaussian mixture background modeling," in *Proc. Int. Conf. Image Process.*, Sep. 2003, p. III-973.
[63] Y. Lu, X. Bai, and F. Wang, "Music recommendation system design based on Gaussian mixture model," in *Proc. 4th Int. Conf. Mechatronics, Mater., Chem. Comput. Eng.* Paris, France: Atlantis Press, 2015.
[64] M. Nilashi, O. B. Ibrahim, N. Ithnin, and R. Zakaria, "A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques," *Soft Comput.*, vol. 19, no. 11, pp. 3173–3207, Nov. 2015.
[65] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, "A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques," in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2018, pp. 102–106.
[66] C. Wangwatcharakul and S. Wongthanavasu, "Improving dynamic recommender system based on item clustering for preference drifts," in *Proc. 15th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2018, pp. 1–6.
[67] V. Subramaniyaswamy and R. Logesh, "Adaptive KNN based recommender system through mining of user preferences," *Wireless Pers. Commun.*, vol. 97, no. 2, pp. 2229–2247, 2017.
[68] B. Wang, Q. Liao, and C. Zhang, "Weight based KNN recommender system," in *Proc. 5th Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, vol. 2, Aug. 2013, pp. 449–452.
[69] S. Vargas and P. Castells, "Improving sales diversity by recommending users to items," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 145–152.
[70] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.
[71] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, no. 1, pp. 79–82, Dec. 2005.
[72] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?" *Geosci. Model Develop. Discuss.*, vol. 7, pp. 1525–1534, Feb. 2014.
[73] A. Gunawardana and G. Shani, "Evaluating recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 265–308.
[74] J. Zhang, Y. Lin, M. Lin, and J. Liu, "An effective collaborative filtering algorithm based on user preference clustering," *Appl. Intell.*, vol. 45, no. 2, pp. 230–240, Sep. 2016.



HANGYU YAN was born in Shannxi, China, in 1994. He is currently pursuing the master's degree in computer technology with Southwest University, Chongqing, China. His current research interests include recommender systems and deep learning.

YAN TANG received the M.Sc. and Ph.D. degrees in computer science from Southwest University, Chongqing, China, where she is currently a Professor with the School of Computer and Information Science. Her current research interests include recommender systems, web application technology, and image processing.