Hindawi
Journal of Sensors
Volume 2023, Article ID 9758709, 1 page
https://doi.org/10.1155/2023/9758709

Retraction

Retracted: PCA and Binary *K*-Means Clustering Based Collaborative Filtering Recommendation

Journal of Sensors

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Li, H. Peng, H. Wang, Q. Huang, and Z. Xu, "PCA and Binary *K*-Means Clustering Based Collaborative Filtering Recommendation," *Journal of Sensors*, vol. 2023, Article ID 2724418, 13 pages, 2023.

Hindawi
Journal of Sensors
Volume 2023, Article ID 2724418, 13 pages
https://doi.org/10.1155/2023/2724418

Research Article

PCA and Binary *K*-Means Clustering Based Collaborative Filtering Recommendation

Xiao Li, Heping Peng, Hongbin Wang, Qingdan Huang, and Zhong Xu

Guangzhou Power Supply Bureau of Guangdong Power Grid Co., Ltd., Guangzhou 510013, China

Correspondence should be addressed to Heping Peng; papercrane@263.net

Received 3 August 2022; Accepted 26 August 2022; Published 11 April 2023

Academic Editor: Yuan Li

Aiming at the problem of similarity calculation error caused by the extremely sparse data in collaborative filtering recommendation algorithm, a collaborative filtering recommendation algorithm based on slope one matrix prefilling model, principal component dimension reduction, and binary *K*-means clustering is proposed in this paper. Firstly, the algorithm uses the slope one model based on item similarity to prefill the original scoring matrix. Secondly, principal component analysis is used to reduce the dimension of the filled matrix, retain the most representative dimension of user characteristics, and remove the dimension with less information. Finally, in order to solve the time-consuming problem of similarity calculation of collaborative filtering algorithm in the case of large-scale system, binary *K*-means clustering is carried out in the reduced dimension vector space to reduce the search range of the nearest neighbour of the target user. The algorithm ensures the efficiency and accuracy of recommendation while the scale of users is expanded. The experimental results on movielens dataset show that the algorithm proposed in this paper is superior to the traditional collaborative filtering algorithm and the collaborative filtering recommendation algorithm based on PCA (principal component analysis) and binary *K*-means clustering in recall rate, accuracy rate, average error, and running time.

1. Introduction

With the development of science and technology and society, people gradually enter the era of information overload from the era of information scarcity. In this era, how to find the information that you are interested in from the mass information and how to make the information that you release stand out have become an urgent problem to be solved. Because people often cannot make clear the information they need, so the past classification catalog and search engine are difficult to meet their needs, so personalized recommendation system as a supplement of search engine becomes a representative solution to the problem of information overload. The recommendation system actively provides users with information they are interested in by analyzing users' historical behaviour information, so it does not need users to provide clear information. Among a large number of existing recommendation technologies, collaborative filtering recommendation technology is the most successful and widely used, which mainly includes user-based and item-based collaborative filtering recommendation algorithm. User-based collaborative filtering is to recommend to target user the items that are liked by users with similar interests and that are not scored by target user. Item-based collaborative filtering is to recommend the target user with items similar to those they used to like. Both of them are realized by calculating user similarity or item similarity through user-item scoring matrix. However, with the increase of users and items, there are some problems such as sparse scoring matrix, cold start, slow response, and poor scalability, which lead to the decline of recommendation quality of traditional collaborative filtering algorithm. In order to solve the problem of data sparsity, Bidyut Kr. Patra et al. [1] fused the similarity of Bhattacharyya and proposed a collaborative filtering algorithm based on neighbourhood similarity measurement method. The results show that the algorithm is superior to the existing collaborative filtering algorithm based on other similarity measurement methods. Suryakant [2] proposed a new similarity method based on average divergence, which considered users' scoring habits.

The results show that the proposed similarity measure is better than the existing one in prediction accuracy.

Han Yanan et al. [3] calculated the user's preference for the item by combining the user's preference for the item's attribute and the popularity of the item and filled in the unsettled item by the sum of the user's preference value and the user's average score. Finally, the time function was used as the weight factor in the similarity calculation and recommendation process, which effectively alleviated the sparse matrix problem and improved the recommendation accuracy. Mahdi Nasiri et al. [4] regarded time as a three-dimensional space and applied an architecture to block users and items in similar groups at the same time and increased time for each block, then inputted appropriate values for the lost data according to the similar users and item scores in each block, finally modelled the relationship between users, items, and time through tensor decomposition. The algorithm reduced the sparsity problem and error rate and achieved good results in practical application.

Truyen Tran et al. [5] combined with Markov random field (MRF) and proposed a sparse induction algorithm to automatically estimate the interaction structure between users and items. Finally, they had proved the effectiveness of this method in large-scale experiments on movie recommendation and data matching datasets. Panpan Wang et al. [6] proposed a recommendation algorithm based on weighted slope one. The experimental results show that the improved algorithm can improve the accuracy and recommendation performance of grade prediction effectively. Maryam Khanian Najafabadi et al. [7] proposed a collaborative filtering algorithm based on association rules and clustering. The experimental results show that the performance of algorithm in precision, recall rate, and other aspects is better than the basic CF and other extended CF technologies even when the data is very sparse.

Mahdi Nasiri and Behrouz Minaei [8] proposed a matrix decomposition method which integrated the initial potential factors of users and items. The results show that the method can improve the accuracy of matrix decomposition technology based on optimization and improve the convergence speed of matrix decomposition. Bo Yang et al. [9] combined the sparse rating data given by users with sparse social trust network and compared with the social collaborative filtering recommendation algorithm based on trust, this algorithm has better performance, especially for cold-start users.

Liu Xiaojun [10] proposed an improved collaborative filtering recommendation algorithm based on clustering. This algorithm used time decay function to preprocess user's score and used clustering algorithm to cluster users and items, respectively. Then, it used the improved similarity measure to generate recommendations. It can effectively solve the problem of data sparsity and new items and the recommendation accuracy has been improved significantly. Faris Alqadah et al. [11] proposed a new collaborative filtering method of top-n recommendation tasks based on double clustering neighbourhood. Experiments show that better recommendations are produced in the algorithm, especially in the case of sparse data. Haipeng You et al. [12] combined item clustering with slope one and the results show that the algorithm can improve the accuracy of collaborative filtering recommendation system effectively. Qlong Ba et al. [13] proposed a collaborative filtering algorithm which combined clustering algorithm with SVD algorithm, which is used in the field of image processing widely. It improves the "cold start" and "data sparsity" of the system and improves the efficiency and scalability of the system. Zhang Shichang [14] proposes an improved collaborative filtering recommendation algorithm based on user-item hybrid model, and designs and implements a personalized news recommendation system. The experimental data proves that this system has a good personalized recommendation function, and the personalized news recommendation system based on this algorithm is more effective. Jing Chen et al. [15] proposed an improved merchant recommendation algorithm based on user reviews TWMR (Timing factors and user Weights Merchant Recommendation algorithm), which is verified that the algorithm TWMR effectively improves the stability of the implicit recommendation effect by the experimental comparison on the Yelp dataset, and a better effect on matrix matching recommendation is made. Alessandro B. Melchiorre et al. [16] investigate to which extent state-of-the-art recommendation algorithms yield different accuracy scores depending on the users' personality traits. Their paper shows several significant differences in performance between user groups scoring high vs. groups scoring low on several personality traits. Vito Walter Anelli et al. [17] establish a common understanding of the state-of-the-art for top-n recommendation tasks. The results of the research show that there is no consistent winner across datasets and metrics for the examined top-n recommendation task. Matteo Montanari et al. [18] researched a problem of the impact of data sampling on a hyper-parameter optimization (HPO) recommendation algorithm in order to achieve the highest accuracy performance.

Donghyun Kim et al. [19] combined convolutional neural network (CNN) with probability matrix decomposition (PMF) and proposed a new context aware recommendation model, convolutional matrix decomposition (convmf). Experimental results show that the algorithm is significantly better than the latest recommendation model even when the rating data is extremely rare. Sheng Li et al. [20] proposed a general CF depth structure combining matrix decomposition and depth feature learning and gave an example of CF depth structure combining probability matrix decomposition and edge denoising stack automatic encoder. Compared with the existing four large dataset movie/book recommendation and response prediction models, the performance of the combined framework is improved. Faisal M. Almutairi et al. [21] showcase the effectiveness of XPL-CF on real data from various application domains and evaluate the explainability of the user-item relationship obtained from XPL-CF through numeric evaluation and case study examples. Dong-Kyu Chae et al. [22] proposed AR-CF, which stands for Augmented Reality CF, a novel framework for addressing the cold-start problems by generating virtual, but plausible neighbours for cold-start users or items and augmenting them to the rating matrix as additional information for CF models. Lianghao Xia et al. [23] propose a new selfsupervised recommendation framework Hypergraph Contrastive Collaborative Filtering (HCCF) to jointly capture local and global collaborative relations with a hypergraph-enhanced cross-view contrastive learning architecture. Yiding Zhang et al. [24] study the novel problem of Geometric Disentangled Collaborative Filtering (GDCF), which aims to reveal and disentangle the latent intent factors across multiple geometric spaces. Oren Barkan et al. [25] break away from the paradigm which is common to a large body of collaborative filtering models that repeatedly demonstrated superior results, and present Anchor-based Collaborative Filtering (ACF). Baptiste Barreau and Laurent Carlier [26] propose a novel collaborative filtering algorithm that captures the temporal context of a user-item interaction through the users' and items' recent interaction histories to provide dynamic recommendations. Ren Jing-xia and Wu Zhi-feng [27] proposed a collaborative filtering algorithm based on dynamic trust attenuation (DTA-CF). Based on the traditional collaborative filtering recommendation algorithm, it examines the common score and time factor to adjust the neighbour selection mechanism and introduces the concept of trust attenuation to redefine the effect of neighbours. Wei Zhang et al. [28] proposed a method of neuro-symbolic interpretable collaborative filtering (NS-ICF), which learns interpretable recommendation rules (consisting of user and item attributes) based on neural networks. Dongsheng Li et al. [29] proposed a neural snapshot ensemble method for collaborative filtering, which can extensively and significantly improve the accuracy (up to 15.9% relatively) when applied to a variety of existing collaborative filtering methods. Hongzhi Liu et al. [30] propose a compiler pass selection and phase ordering approach, called Iterative Compilation based on Metric learning and Collaborative filtering (ICMC). Based on the learned similarity metric, a neighbourhood-based collaborative filtering method is employed to iteratively recommend a few superior compiler passes for each target program. Athanasios N. Nikolakopoulos and George Karypis [31] proposed item-based models are among the most popular collaborative filtering approaches for building recommender systems. To sum up, in the research of collaborative filtering recommendation algorithm, people try to improve the recommendation quality of collaborative filtering algorithm by improving various rules, mechanisms, and algorithms. In the aspect of similarity calculation, the nearest neighbour similarity, average divergence, personal preference, and time dimension are introduced to improve the accuracy of similarity calculation; Aiming at the problem of sparse data, we use sparse induction, weighted slope, matrix decomposition, and sparse rating to improve the performance of collaborative filtering algorithm; in the aspect of recommendation algorithm, there are many methods such as clustering based, double clustering neighbourhood based, user-item hybrid model based, user comments based, and hyper-parameter optimization based; in the aspect of collaborative filtering, convolutional neural network (CNN), probability matrix decomposition, self-monitoring recommendation framework hypergraph, and dynamic trust decay (DTA-CF) are applied to improve the performance and recommendation efficiency of collaborative filtering algorithm. However, further research is needed to solve the problem of similarity calculation error caused by sparse data in collaborative filtering recommendation algorithm. Therefore, a collaborative filtering algorithm (named SOPK-CF) based on slope one matrix prefilling model, principal component analysis, and binary K-means clustering is presented in this paper. Firstly, adopt slope one matrix filling model fills in the original user-item scoring matrix then uses principal component analysis (PCA) to reduce the dimension of the filled matrix. Finally, the binary K-means clustering algorithm is used to cluster the dimension reduced data. The nearest neighbour of the target user can be quickly obtained by finding the category of the target user. Finally, through the nearest neighbour similarity, the prediction value of the current user to the non-evaluated items is calculated by weighting.

2. The Problem of Similarity Calculation Error Caused by the Extremely Sparse Data in This Algorithm

The traditional collaborative filtering algorithms often use calculation formulas such as Jaccard, Euclid, cosine similarity, and modified cosine similarity to calculate user or item similarity. These methods are all calculated on the original scoring matrix, so the calculation accuracy depends on the accuracy of the original scoring matrix. Therefore, the original score matrix is too sparse, which will directly lead to the inaccuracy of similarity calculation. For example, when both users have comments on a few popular goods or necessities, it does not mean they are similar. Therefore, we can consider filling the original scoring matrix, but the filling method should be accurate; otherwise, the original scoring matrix will be wrongly filled, which will lead to the lower accuracy of the original scoring matrix and eventually lead to the decline of recommendation quality.

2.1. The Basic Principle of SOPK-CF. Aiming at the problem of data sparsity and scalability of traditional collaborative filtering recommendation algorithm, a collaborative filtering algorithm based on the slope one matrix prefilling model, principal component analysis, and binary K-means clustering (SOPK-CF) is proposed in this paper. The symbol in this paper is shown (see Symbols).

2.1.1. Data Sparsity Problem. Firstly, slope one matrix filling model is used to fill in the original scoring matrix, which is more accurate than mean filling, zero filling, and mode filling. Then, PCA is used to reduce the dimension of the filled matrix, retain the important information, and remove the noise information.

The slope one matrix filling model refers to the weighted slope one algorithm integrating project similarity. Its steps are as follows:

Input: original scoring matrix R.

Step 1: calculate the modified cosine similarity $sim\alpha(i, j)$ and category similarity $sim\beta(i, j)$ of the item, respectively;

the calculation formulas are (1) and (2), respectively:

$$sim_{\alpha}(i,j)=\frac{\sum_{u\in U_{i,j}}\left(R_{u,i}-\overline{R_u}\right)\left(R_{u,j}-\overline{R_u}\right)}{\sqrt{\sum_{u\in U_{i,j}}\left(R_{\mathrm{u},i}-\overline{R_u}\right)^2}\sqrt{\sum_{u\in U_{i,j}}\left(R_{u,j}-\overline{R_u}\right)^2}}, \tag{1}$$

$$sim_{\beta}(i,j)=\frac{\left|U_i\cap U_j\right|}{\left|U_i\cup U_j\right|}, \tag{2}$$

where $U_{i,j}$ is the common scoring user set of item i and item j, $R_{u,i}$ is the rating of item i by user u, $\overline{R_u}$ is the average rating of user u, and U_i is the user set of comment item i.

Step 2: combine the modified cosine similarity $sim\alpha(i,j)$ and category similarity $sim\beta(i,j)$ to synthesize the final item similarity sim (i,j) and the calculation formula of sim (i,j) is as follows:

$$sim(i,j)=(1-\lambda)sim_{\alpha}(i,j)+\lambda sim_{\beta}(i,j). \tag{3}$$

Step 3: fill in the original scoring matrix. If it is the default, use formula (4) to predict and fill in the scoring matrix.

$$P_{u,i}=\frac{\sum_{j\int \mathrm{sim}_i}\mathrm{num}_{i,j}\left(R_{u,j}-\mathrm{Dis}_{i,j}\right)}{\sum_{j\int \mathrm{sim}_i}\mathrm{num}_{i,j}}, \tag{4}$$

where $P_{u,i}$ represents the user u's prediction score for item i, sim_i represents the set of nearest neighbours to item i, $num_{i,j}$ represents the number of users scoring jointly for item i and item j, $R_{u,j}$ represents the user u's score for item i, and $Dis_{i,j}$ represents the average difference between the users scoring jointly for item i and item j.

Output: filled matrix FR.

Dimension reduction technology makes data easier to use and they can often remove the noise in the data, so it makes other machine learning tasks more accurate. Although dimension reduction also causes certain information loss, in practical applications, we usually only need to retain the most important features of data and information loss within a certain range is allowed. Dimension reduction is often used as a preprocessing step to clean data before it is applied to other algorithms. There are many techniques for data dimension reduction; in these techniques, independent principal component analysis, factor analysis, and principal component analysis are popular, then the principal component analysis is used widely. PCA is one of the most commonly used linear dimension reduction methods. PCA transforms the data from the original coordinate system to the new coordinate system, and the choice of the new coordinate system is determined by the data itself. The direction with the largest variance in the original data is selected in the first new coordinate axis and the direction with the largest variance which is orthogonal to the first coordinate axis is selected in the second new coordinate axis. This process is repeated all the time; the number of repetitions is the number of features in the original data. We will find that most of the variance is contained in the first few new axes. Therefore, in order to reduce the dimension of the data, the remaining coordinate axis is ignored. The main steps of PCA algorithm are as follows:

Input: filled matrix FR.

Step 1: average every dimension of matrix FR^T, that is, subtract the mean value of this dimension from the data of this dimension.

Step 2: calculate the covariance matrix of the sample matrix with equation (5).

$$C=\frac{1}{m}\mathrm{FR}^{T*}\mathrm{FR}. \tag{5}$$

Step 3: find out the eigenvalues and eigenvectors corresponding to the covariance matrix.

Step 4: arrange the eigenvectors into a matrix from top to bottom according to the size of the corresponding eigenvalues and form the matrix P from the first s rows.

Output: PR = FR * P, PR is the data that the filled matrix FR is reduced to s dimension. In order to reduce the projection error, it is essential to select the appropriate s value, which can be determined experimentally by equation (6):

$$\mathrm{Error}=\frac{1/m\sum_{i=1}^{m}\left|\left|\mathrm{FR}^{\mathrm{T}(i)}-\mathrm{FR}^{\mathrm{T}}{}_{\mathrm{approx}}{}^{(i)}\right|\right|^2}{1/m\sum_{i=1}^{m}\left|\left|\mathrm{FR}^{\mathrm{T}(i)}\right|\right|^2}\le \mathrm{errval}, \tag{6}$$

where m is the number of features, the numerator is the sum of the distance between the original data point and the projected point. Error indicates the error. The smaller the error indicates, more principal components are retained, the better the effect of dimensionality reduction. errval is the upper limit of error. Generally, errval is 0.01, i.e., 99% of the original data is retained.

2.1.2. Time Consuming of Similarity Calculation. Calculating the similarity between all users is needed in the traditional collaborative filtering algorithm. With the increase of the number of users, the calculation of similarity becomes very large and the scalability of the traditional algorithm also highlights. Therefore, the binary K-means clustering algorithm for the data clustering after dimension reduction is introduced in this paper, finally calculating the similarity between users in the same cluster is only needed, so it greatly reduces the calculation of similarity between users and improves the scalability of the algorithm.

Clustering is the process of dividing a set of physical or abstract objects into multiple classes. K-means clustering is a classical clustering algorithm. Firstly, the algorithm randomly selects k clustering centres according to the dataset, and then calculates the distance between each data point and each cluster centre, places the data point in the cluster corresponding to the nearest cluster centre, then calculates the average value of each cluster as the new cluster centre of the cluster. Repeat the above steps until the cluster centre no longer changes. However, our algorithm is very susceptible to the selection of the initial clustering centre. Improper

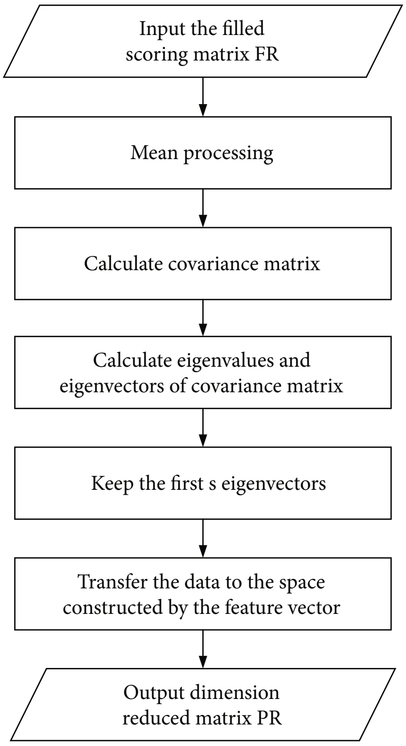
Input the filled scoring matrix FR
Mean processing
Calculate covariance matrix
Calculate eigenvalues and eigenvectors of covariance matrix
Keep the first s eigenvectors
Transfer the data to the space constructed by the feature vector
Output dimension reduced matrix PR

Figure 1: The process of dimensionality reduction of user-item scoring matrix filled by slope one using PCA.

selection will lead to the aggregation of the centre of mass, so that the clustering effect may be locally optimal. SSE is an index used to measure the clustering effect. Its value is the sum of the squares of the distance between the data points in each cluster and the centre of the cluster. The calculation formula is shown in (7):

$$\mathrm{SSE}=\sum_{i=1}^{k}\sum_{x\in c_i}\mathrm{dist}(c_i,x)^2, \tag{7}$$

where k is the number of selected clusters, c_i is the cluster centre of the i-th cluster, x is the data in the cluster of c_i, and dist is Euclidean distance. The more small SSE is, the better clustering effect is. Binary K-means algorithm is an improved algorithm of K-means algorithm. In this algorithm, all data points are regarded as a cluster first, and a cluster is selected for k-means (k=2) division. The criteria of selection are that the divided cluster can reduce the value of SSE to the maximum extent, so as to continue until the number of clusters is equal to the number k given by users. Compared with k-means algorithm, this algorithm has faster clustering speed, less influence by initial clustering centre, and better clustering effect.

2.2. The Algorithm Flow of SOPK-CF. Run SOPK-CF algorithm on movielens dataset and the specific process is as follows:

Step1: fill in the default value of the original scoring matrix R with slope one matrix filling model (weighted slope one algorithm incorporating project similarity), and the filled matrix is FR.

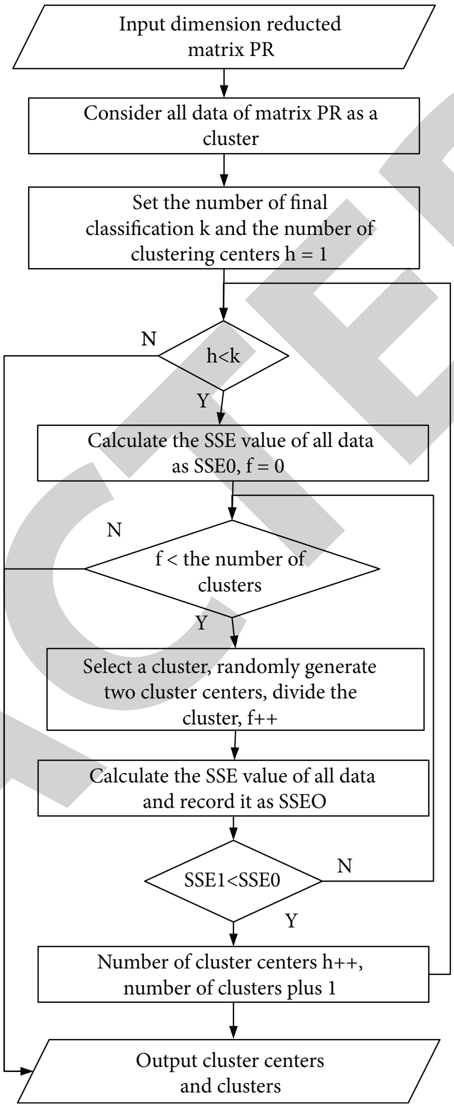
Input dimension reducted matrix PR
Consider all data of matrix PR as a cluster
Set the number of final classification k and the number of clustering centers h = 1
N
h<k
Y
Calculate the SSE value of all data as SSE0, f = 0
N
f < the number of clusters
Y
Select a cluster, randomly generate two cluster centers, divide the cluster, f++
Calculate the SSE value of all data and record it as SSEO
SSE1<SSE0
N
Y
Number of cluster centers h++, number of clusters plus 1
Output cluster centers and clusters

Figure 2: The process of clustering PCA dimensionality reduced matrix using binary K-means clustering.

Step 2: according to the process (see Figure 1), use PCA algorithm to extract the principal components of the filled matrix FR, and the reduced dimension matrix is PR.

Step 3: according to the process shown in Figure 2, binary K-means clustering is carried out for the reduced dimension matrix PR to obtain multiple clusters and cluster centres of each cluster.

Step 4: use formula (8) to calculate the similarity between the target user u and other user v in the target user u's cluster then sort the similarity from large to small.

$$\mathrm{sim}(u,v)=\frac{\mathrm{dist}(u,v)}{\sum_{v\in U_u}\mathrm{dist}(u,v)}, \tag{8}$$

where sim (u,v) represents the similarity of user u and user v, dist (u,v) represents the distance between target user u

Table 1: Performance comparison of collaborative filtering recommendation methods.

Algorithm\index	Recall	Precision	MAE	Time cost	Space cost
Traditional collaborative filtering algorithm	Low	Low	Low	Big	Small
Collaborative filtering recommendation algorithm based on PCA dimension reduction and binary K-means clustering	Hight	Hight	Hight	Small	Big
Collaborative filtering algorithm based on slope one matrix prefilling model, principal component analysis, and binary K-means clustering	Hight	Hight	Hight	More small	More big

Table 2: Format of u.data table.

UserID	MovieID	Rating	Timestamp
User number	Movie number	Score	Time stamp

Table 3: Format of u.item table.

MovieID	Title	Genres
Movie number	Movie name	Movie genre

and other user v, and U_u represents the set of other users in the cluster of target user u.

Step 5: use the prediction method based on nearest neighbour to predict the user u's score on the unseated item i. The specific formula is as follows:

$$P_{u,i} = \overline{R_u} + \frac{\sum_{v\in \mathrm{cuc}} \mathrm{sim}(u,v)\left(R_{vi} - \overline{R_v}\right)}{\sum_{v\in \mathrm{cuc}} \left|\mathrm{sim}(u,v)\right|}, \qquad (9)$$

where $\overline{R_u}$ is the average score of user u, cuc is the nearest neighbour of user u, $\mathrm{sim}\,(u,v)$ represents the similarity between target user u and other user v, R_{vi} represents the score of user v on item i, and $\overline{R_v}$ represents the average score of user v.

Step 6: make top-n recommendation and form the recommendation list according to the prediction score.

2.3. Theoretical Analysis and Comparison of Methods. Due to the sparsity of the original scoring matrix, the traditional user-based collaborative filtering algorithm has errors in calculating user similarity, which results in low recommendation quality (accuracy, recall, and average error); in calculating user similarity, it will calculate the similarity among all users, so the time consumption is large. The collaborative filtering recommendation algorithm based on PCA dimension reduction and binary K-means clustering firstly uses the mean to fill the original scoring matrix, which alleviates the data sparsity to some extent, so improves the recommendation quality. Then, PCA is used to reduce dimension and remove a small amount of information. Finally, binary K-means is used to cluster. When calculating user similarity, only the similarity between the target user and other users in the cluster needs to be calculated, so the time consumption is small. The collaborative filtering algorithm based on slope one matrix prefilling model, principal component analysis, and binary K-means clustering uses slope one matrix pre filling model in matrix prefilling, which makes the filling data more accurate and further improves the recommendation quality. The performance comparison of each algorithm is shown (see Table 1).

3. Experimental Results and Analysis

3.1. Dataset. The experiment uses the movieslen dataset [32] provided by the GroupLens project group of the University of Minnesota, which includes 943 users' 100000 scoring records for 1682 movies. Among them, each user has scored at least 20 movies with a rating range of 1, 2, 3, 4, and 5, 1 means "very bad" and 5 means "very good." By calculating the proportion of the unsettled items in the whole dataset, the data sparsity is 93.6953%, and it is suitable to test the alleviating effect of SOPK-CF algorithm on data sparsity. This paper uses the data in the u.data table and u.item table and mainly calculates the four fields of userid, movieid, rating, and genres. The genres field contains 18 types and a movie can belong to multiple types. u.data table and u.item table formats are shown (see Tables 2 and 3).

We also conduct experiments on dataset *MoCap* (comes from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php)) with dimension =36 and cardinality n =65, 536, which has 5 types of hand postures from 12 users.

3.2. Evaluation Index. In the experiment, MAE (mean absolute deviation), precision, and recall were used as evaluation indexes. MAE reflects the deviation between the predicted score and the actual score. The smaller the deviation, the higher the recommendation quality. MAE is calculated as follows:

$$\mathrm{MAE} = \frac{\sum_{i\in I}\left|P_i - T_i\right|}{|I|}, \qquad (10)$$

where I is the intersection set of the items in the recommendation list and the items in the test set, P_i is the predicted score of item i, and T_i is the score of item i in the test set.

The accuracy rate describes the ratio between the number of items recommended to the user accurately and the number of items recommended to the user. Therefore, the larger it is, the better. The calculation formula (11) is as

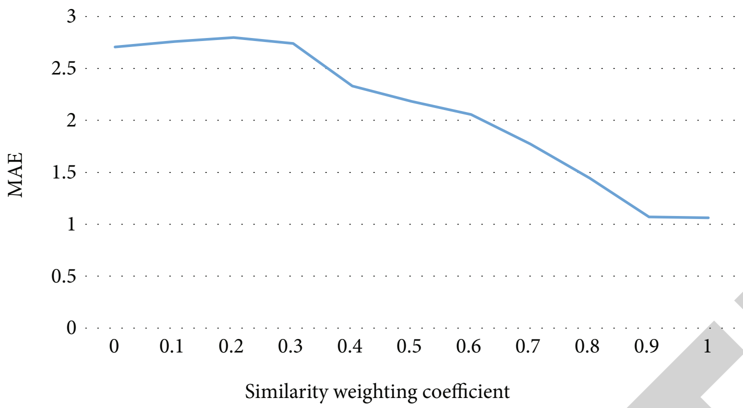

Figure 3: Influence of similarity weighting coefficient λ on MAE.

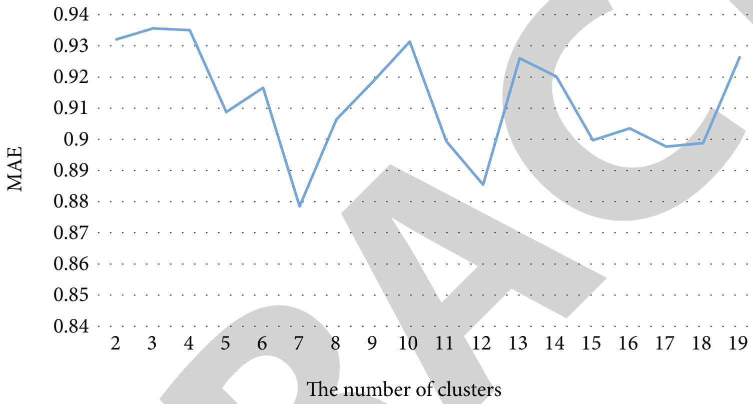

Figure 4: The influence of the number of clusters k on MAE.

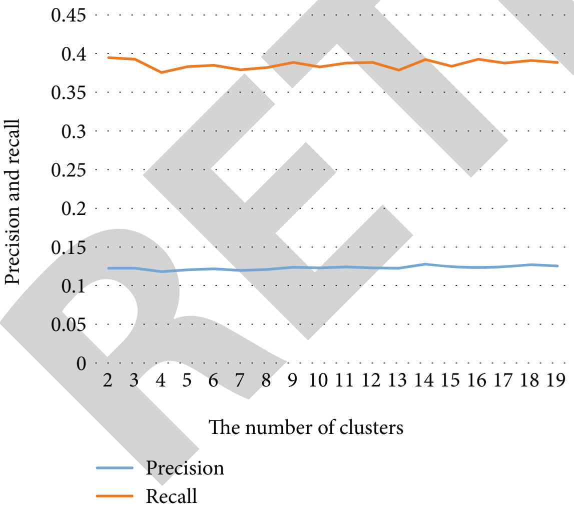

Figure 5: The influence of cluster number k on Precision and Recall.

follows:

$$\text{Precision} = \frac{|R(u) \cap T(u)|}{|R(u)|}, \tag{11}$$

where $R(u)$ is the list of items recommended to the target user u, and $T(u)$ is the set of items evaluated by target user u.

The recall rate describes the ratio between the number of items recommended to user accurately and the number of items commented by user in the test set. Therefore, the larger it is, the better. The formula (12) is as follows:

$$\text{Recall} = \frac{|R(u) \cap T(u)|}{|T(u)|}, \tag{12}$$

where $P(u)$ is the list of items recommended to the target user u, and $T(u)$ is the set of items evaluated by the target user u in the set.

3.3. Experimental Environment and Parameter Setting

3.3.1. Experimental Environment. Processor: Intel(R) Core(TM)i5-7400 CPU @3.00GHz.

Install memory (RAM): 8.00GB.

Running environment: Win10 (64 bit) operating system.

Development language: Python.

Programming tools: Pychar.

3.3.2. Parameter Setting

(1) The Training of Similarity Weighting Coefficient λ. It can be seen (see Figure 3) that when the similarity weighting

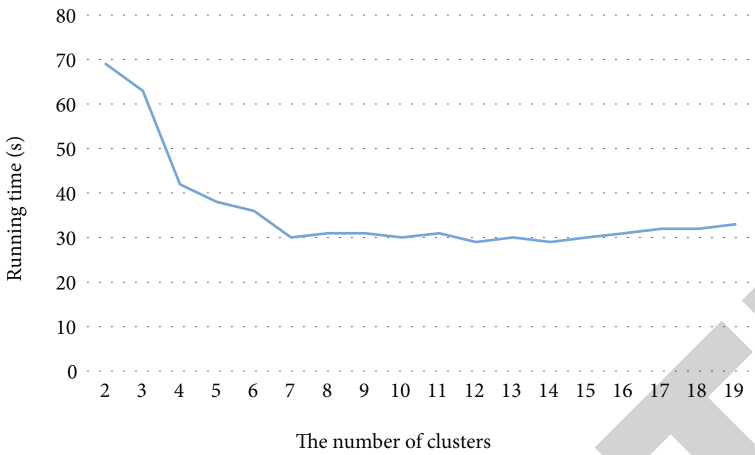

Figure 6: The influence of cluster number k on Running time.

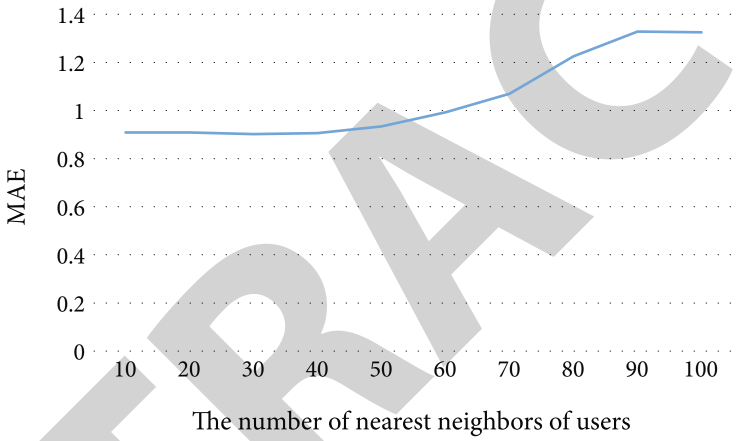

Figure 7: The influence of the number of nearest neighbours UserNum on MAE.

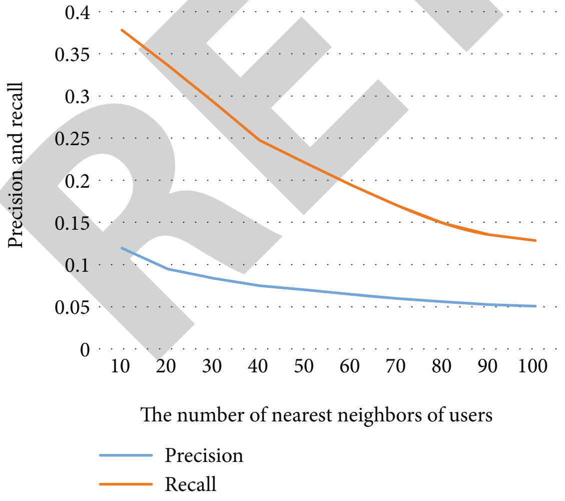

Figure 8: The influence of the number of nearest neighbours UserNum on Precision and Recall.

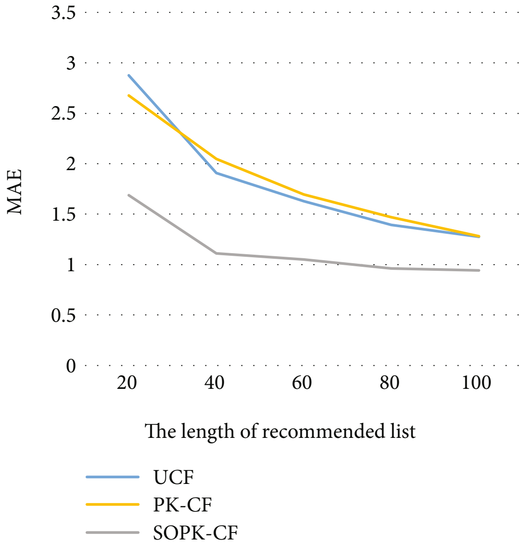

Figure 9: Comparison of average MAE of each algorithm on *movieslen*.

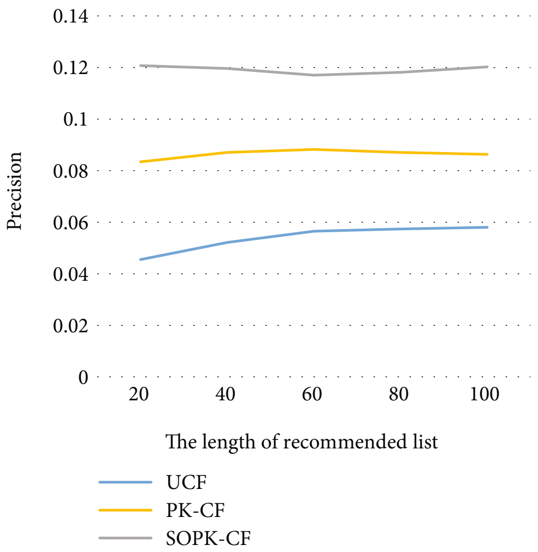

Figure 10: Comparison of average Precision of each algorithm on *movieslen*.

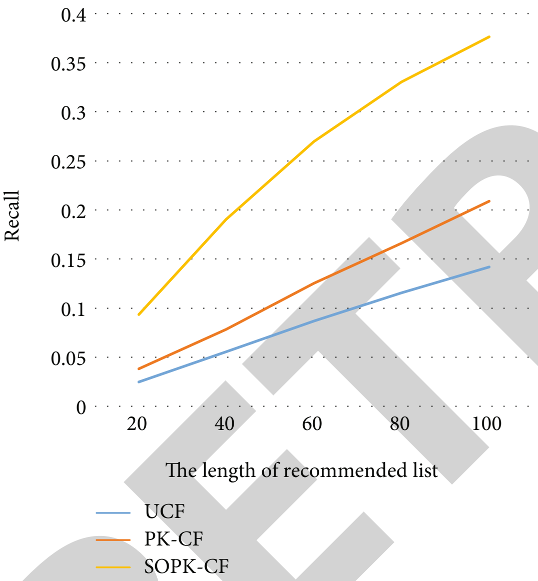

Figure 11: Comparison of average Recall of each algorithm on *movieslen*.

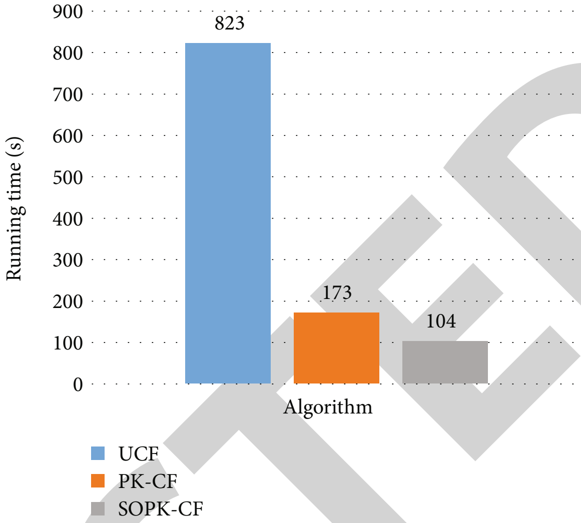

Figure 12: Comparison of Running time of each algorithm on *movieslen*.

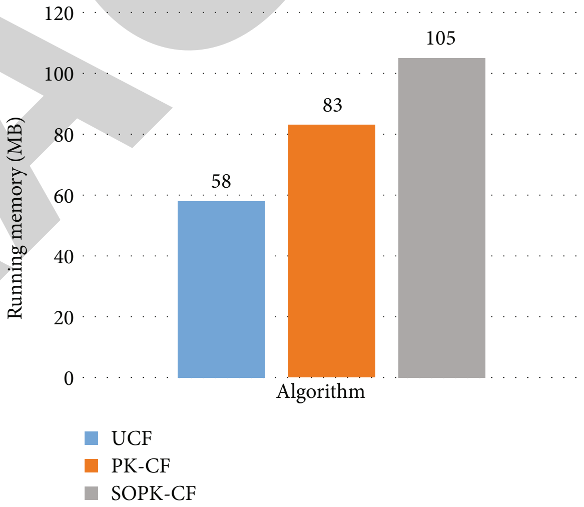

Figure 13: Comparison of Running memory of each algorithm on *movieslen*.

coefficient is 0.9, the MAE (average absolute error between the filled data and the corresponding data in the test set) is the smallest, so λ is 0.9 in this paper.

(2) The Training of Cluster Number k. It can be seen (see Figure 4) that when the number of clusters is 7, the MAE is the smallest (see Figure 5). It can be seen that the number of clusters basically does not affect the accuracy and recall rate. Because the algorithm mainly uses the nearest neighbour set of the target user when recommending items to the target user, no matter the number of clusters, as long as the part of the target user's cluster that is most similar to the target user is guaranteed (10 users). However, the number of clusters will affect the running time of the algorithm, because the number of users in the cluster affects the calculation of user similarity. Therefore, it can be seen (see Figure 6) that when the number of clusters is 7 to 16, the running time is the minimum (this time refers to the time from binary K-means to the generation of recommendations), so in this paper, k is 7.

(3) The Training of the Number of Nearest Neighbours of Users UserNum. It can be seen (see Figure 7 and 8) that when the number of nearest neighbours of users is 10, the average MAE is the smallest, the average precision and the average recall rate are the largest, so UserNum should be 10 in this paper.

3.4. Experimental Result. When the similarity weighting coefficient λ is 0.9, the number of clusters k is 7, and the

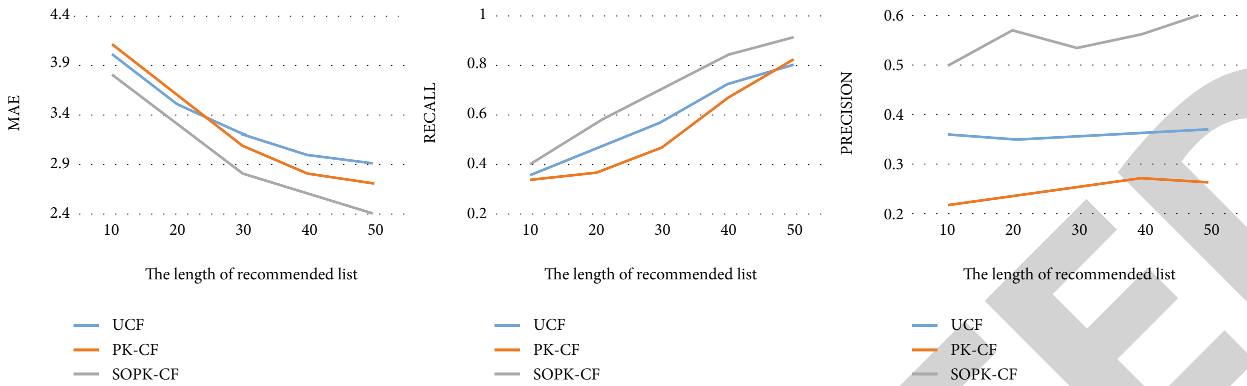

Figure 14: Comparisons of MAE, Recall, and Precision on *Household.*

number of users' nearest neighbours UserNum is 10, the traditional user-based collaborative filtering algorithm UCF, the collaborative filtering algorithm based on PCA and binary K-means clustering PK-CF, the collaborative filtering algorithms based on the slope one matrix prefilling model and principal component analysis and binary K-means clustering SOPK-CF are compared. Among them, Figures 9–11 are the comparison charts of average MAE of each algorithm, the average precision of each algorithm, and the average recall of each algorithm. Figures 12 and 13 are drawn according to the experimental results of using different algorithms to recommend 20, 40, 60, 80, and 100 items to all users in the test set in order.

The comparison of MAE Recall and Precision on dataset Household is shown in Figure 14, as it plots, we can see that the proposed algorithm also outperforms its competitor.

4. Result Analysis

Experiments are carried out on the movielens dataset and Figures 9–13 are drawn. It is a series of the comparison figures of each algorithm on average absolute error, average accuracy, average recall rate, running time, and running memory.

4.1. Recommended Quality Analysis

(1) On the average absolute error (see Figure 9): the SPOK-CF algorithm proposed in this paper is better than UCF and PK-CF algorithm in the average absolute error and the average absolute error values of the three algorithms show a downward trend with the growing recommendation list

(2) On the average accuracy (see Figure 10): it can be seen that the average accuracy of SPOK-CF algorithm proposed in this paper is higher than UCF and PF-CF algorithm, and with the growing recommendation list, the average accuracy of the three algorithms shows a gentle trend

(3) On the average recall (see Figure 11): it can be seen that the average recall of SOPK-CF algorithm proposed in this paper is higher than UCF and PK-CF algorithm; the average recall of the three algorithms is increasing with the growing recommendation list

It can be seen from the appeal analysis that the quality of recommendation can be improved by prefilling sparse original scoring matrix with slope one algorithm incorporating item similarity and by using principal component analysis algorithm to reduce the dimension of the filled matrix and retain its main features.

4.2. Time and Space Consumption Analysis

(1) On the running time (see Figure 12): the running time of the three algorithms are 823s, 173s and 104s respectively, that is, SOPK-CF < PK-CF < UCF. SOPK-CF algorithm in this paper is much less than UCF algorithm and about half of PK-CF algorithm in running time

(2) On the running memory (see Figure 13): the running memory of the three algorithms are 105mb, 83mb, and 58mb, respectively, that is, SOPK-CF > PK-CF > UCF. SOPK-CF algorithm in this paper is a little larger than PK-CF algorithm and about twice of UCF algorithm in running memory

It is shown that using binary *k*-means algorithm to cluster users and only calculate the similarity between the target users and other users in the cluster can save a lot of time. In the system with high time requirement and loose memory size requirement, the algorithm proposed in this paper is readily acceptable.

5. Conclusion

Aiming at the problem of sparse data and low scalability of traditional collaborative filtering algorithm, a collaborative filtering algorithm based on slope one matrix prefilling model, principal component analysis, and binary K-means clustering SPOK-CF is proposed in this paper. The experimental results show that the algorithm proposed in this paper is superior to the traditional user-based collaborative filtering algorithm and the collaborative filtering algorithm based on PCA and binary K-means in average MAE, average precision, average recall, and algorithm running time.

However, this algorithm only uses one filling method, one dimension reduction method, and one clustering method and does not try other algorithms. Therefore, the next step is to try other mainstream and efficient algorithms on matrix filling and dimension reduction [33] and other clustering [34–37].

Symbols

m:	The number of user
n:	The number of item
i:	Item i
j:	Item j
u:	User u
v:	User v
Ui, j:	The common rating user set of item i and item j
s:	The dimension of reduced matrix
R:	The original scoring matrix (m*n)
$sim\alpha(i, j)$:	The modified cosine similarity of item i and item j
$sim\beta(i, j)$:	The category similarity of item i and item j
Ru, i:	The user u's rating of item i
Ru, j:	The user u's rating of item j
Error:	The data error before and after dimension reduction
$FR^{T^{(i)}}$:	The column i of matrix FR^T
$FR^{T^{(i)}}_{approx}$:	The column i of projected matrix FR^T
k:	The number of clusters
co:	The cluster centre of the o-th cluster
dist:	Euclid distance
SSE0:	The initial SSE value
f:	The number of clusters to divide
$\mathrm{dist}(u, v)$:	The Euclidean distance between user u and user v
cuc:	The nearest neighbour set of user u
Rvi:	The user v's rating of item i
I:	The intersection of recommended items and items in the test set
Ti:	The score of item i in the test set
$R(u)$:	The list of recommended items for target user u
Recall:	Recall rate
$\overline{R_U}$:	The average score of user u
Ui:	The user set for comment item i
Uj:	The user set for comment item j
λ:	The weighting coefficient of modifying cosine similarity and category similarity
$\mathrm{sim}(i, j)$:	The final similarity between item i and item j
Pu, i:	The user u's forecast score for item i
$simi$:	The nearest neighbour set of item i
$numi, j$:	The number of users commented on both item i and item j
$Disi, j$:	The average difference between the user's scores on item i and item j
FR:	The filled matrix (m*n)
FR^T:	The transpose matrix of filled matrix **FR** (n*m)
C:	The covariance matrix (n*n)
PR:	The reduced dimension matrix (m*s)
P:	The eigenvector matrix (n*s)
P^T:	The transpose matrix of eigenvector matrix **P** (s*n)
SSE:	The sum of squares of errors
o:	The o-th cluster
x:	The data in cluster with cluster centre co
h:	The variable representing the number of cluster centres
SSE1:	The first SSE value
$\mathrm{sim}(u, v)$:	The similarity between user u and user v
Uu:	The collection of other users in the cluster of target user u
$\overline{R_V}$:	The average score of user v
MAE:	The mean of absolute error
Pi:	The forecast score for item i
Precision:	Accuracy rate
$T(u)$:	The set of items that target user u have commend in test set
erral:	The upper limit of data error before and after PCA dimension reduction.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

Xiao Li is responsible for the supplement and revision of the paper; Heping Peng puts forward the research proposition, designed the research ideas, and wrote the paper; Hongbin Wang is responsible for the overall structure of the paper; Qingdan Huang and Zhong Xu are responsible for reviewing and revising some of the algorithms.

Acknowledgments

Research works in this paper are supported by the research and application of the key technology of electromagnetic transient cloud simulation platform for extremely large urban distribution network (080037KK52170012/GZJKJXM20170023).

References

[1] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," *Knowledge-Based Systems*, vol. 82, pp. 163–177, 2015.

[2] T. M. Suryakant, "A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment," *Procedia Computer Science*, vol. 89, pp. 450–456, 2016.

[3] H. Yanan, C. Han, and L. Liangliang, "Collaborative filtering recommendation algorithm based on score matrix filling and user interest," *Computer Engineering*, vol. 42, no. 1, pp. 36–40, 2016.

[4] M. Nasiri, Z. Sharifi, and B. Minaei, "Alleviate sparsity problem using hybrid model based on spectral co-clustering and tensor factorization," in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 285–289, Mashhad, Iran, 2015.

[5] T. Tran, D. Phung, and S. Venkatesh, "Collaborative filtering via sparse Markov random fields," *Information Sciences*, vol. 369, pp. 221–237, 2016.

[6] P. Wang, Q. Qian, Z. Shang, and J. Li, "An recommendation algorithm based on weighted Slope one algorithm and user-based collaborative filtering," in *2016 Chinese Control and Decision Conference (CCDC)*, pp. 2431–2434, Yinchuan, China, 2016.

[7] M. K. Najafabadi, M. N.'r. Mahrin, S. Chuprat, and H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Computers in Human Behavior*, vol. 67, pp. 113–128, 2017.

[8] M. Nasiri and B. Minaei, "Increasing prediction accuracy in collaborative filtering with initialized factor matrices," *The Journal of Supercomputing*, vol. 72, no. 6, pp. 2157–2169, 2016.

[9] B. Yang, L. Yu, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[10] L. Xiaojun, "An improved clustering-based collaborative filtering recommendation algorithm," *Cluster Computing*, vol. 20, no. 2, pp. 1281–1288, 2017.

[11] F. Alqadah, C. K. Reddy, J. Hu, and H. F. Alqadah, "Biclustering neighborhood-based collaborative filtering method for top-n recommender systems," *Knowledge and Information Systems*, vol. 44, no. 2, pp. 475–491, 2015.

[12] H. You, H. Li, Y. Wang, and Q. Zhao, "An improved collaborative filtering recommendation algorithm combining item clustering and Slope One scheme," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 2015.

[13] Q. Ba, X. Li, and Z. Bai, "Clustering collaborative filtering recommendation system based on SVD algorithm," in *Proceedings of 2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp. 997–1001, Beijing, 2013.

[14] Z. Shichang, "Research on recommendation algorithm based on collaborative filtering," in *2021 2nd International Conference on Artificial Intelligence and Information Systems*, pp. 1–4, Chongqing China, 2021.

[15] J. Chen, H. Yang, and L. Duan, "An improved merchant recommendation algorithm based on user reviews," in *2021 The 4th International Conference on Information Science and Systems*, pp. 102–110, Edinburgh United Kingdom, 2021.

[16] A. B. Melchiorre, E. Zangerle, and M. Schedl, "Personality bias of music recommendation algorithms," in *Fourteenth ACM Conference on Recommender Systems*, pp. 533–538, Virtual Event Brazil, 2020.

[17] V. W. Anelli, A. Bellogín, T. Di Noia, D. Jannach, and C. Pomo, "Top-N recommendation algorithms: a quest for the state-of-the-art," in *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 121–131, Barcelona Spain, 2022.

[18] M. Montanari, C. Bernardis, and P. Cremonesi, "On the impact of data sampling on hyper-parameter optimisation of recommendation algorithms," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1399–1402, Virtual Event, 2022.

[19] D. Kim, C. Park, O. Jinoh, S. Lee, and Y. Hwanjo, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 233–240, Boston Massachusetts USA, 2016.

[20] S. Li, J. Kawale, and F. Yun, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 811–820, Melbourne Australia, 2015.

[21] F. M. Almutairi, N. D. Sidiropoulos, and B. Yang, "XPL-CF: explainable embeddings for feature-based collaborative filtering," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2847–2851, Virtual Event Queensland Australia, 2021.

[22] D.-K. Chae, J. Kim, D. H. Chau, and S.-W. Kim, "AR-CF: augmenting virtual users and items in collaborative filtering for addressing cold-start problems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1251–1260, Virtual Event China, 2020.

[23] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, and J. Huang, "Hypergraph contrastive collaborative filtering," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 70–79, Madrid Spain, 2022.

[24] Y. Zhang, C. Li, X. Xie et al., "Geometric disentangled collaborative filtering," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 80–90, Madrid Spain, 2022.

[25] O. Barkan, R. Hirsch, O. Katz, A. Caciularu, and N. Koenigstein, "Anchor-based collaborative filtering," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2877–2881, Virtual Event Queensland Australia, 2021.

[26] B. Barreau and L. Carlier, "History-augmented collaborative filtering for financial recommendations," in *Fourteenth ACM Conference on Recommender Systems*, pp. 492–497, Virtual Event Brazil, 2020.

[27] R. Jing-xia and Z.-f. Wu, "Collaborative filtering algorithm based on dynamic trust attenuation," in *Proceedings of the 2020 3rd International Conference on Big Data Technologies*, pp. 121–125, Qingdao China, 2020.

[28] W. Zhang, J. Yan, Z. Wang, and J. Wang, "Neuro-symbolic interpretable collaborative filtering for attribute-based recommendation," in *Proceedings of the ACM Web Conference*, pp. 3229–3238, Virtual Event, Lyon France, 2022.

[29] D. Li, H. Liu, C. Chen, Y. Zhao, S. M. Chu, and B. Yang, "Neu SE: a neural snapshot ensemble method for collaborative filtering [J]," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 6, pp. 1–20, 2021.

[30] H. Liu, J. Luo, Y. Li, and W. Zhonghai, "Iterative compilation optimization based on metric learning and collaborative filtering," *ACM Transactions on Architecture and Code Optimization*, vol. 19, no. 1, pp. 1–25, 2022.

[31] A. N. Nikolakopoulos and G. Karypis, "Boosting item-based collaborative filtering via nearly uncoupled random walks,"

ACM Transactions on Knowledge Discovery from Data, vol. 14, no. 6, pp. 1–26, 2020.

[32] F. Maxwell Harper and J. A. Konstan, "The movie lens datasets: history and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.

[33] Y. W. Chen, J. L. Wang, Y. Q. Cai, and J. X. Du, "A method for Chinese text classification based on apparent semantics and latent aspects," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 4, pp. 473–480, 2015.

[34] M. Yan, Y. Chen, X. Hu, D. Cheng, Y. Chen, and J. du, "Intrusion detection based on improved density peak clustering for imbalanced data on sensor-cloud systems," *Journal of Systems Architecture*, vol. 118, article 102212, 2021.

[35] M. Yan, Y. Chen, Y. Chen, G. Zeng, X. Hu, and J. du, "A lightweight weakly supervised learning segmentation algorithm for imbalanced image based on rotation density peaks," *Knowledge-Based Systems*, vol. 244, article 108513, 2022.

[36] Y. Chen, X. Hu, W. Fan et al., "Fast density peak clustering for large scale data based on kNN," *Knowledge-Based Systems*, vol. 187, article 104824, 2020.

[37] Y. Chen, L. Zhou, S. Pei et al., "KNN-BLOCK DBSCAN: fast clustering for large scale data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3939–3953, 2021.