A Clustering-Based Collaborative Filtering Recommendation Algorithm via Deep Learning User Side Information

Chonghao Zhao[1], Xiaoyu Shi[2](✉), Mingsheng Shang[2], and Yiqiu Fang[1]

[1] School of Computer Science and Technology, Chongqing University
of Posts and Telecommunications, Chongqing 400714, China
S180231916@stu.squpt.edu.cn, fangyq@squpt.edu.cn

[2] Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of
Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
{xiaoyushi,msshang}@cigit.ac.cn

Abstract. Collaborative filtering (CF) is a widely used recommendation approach that relies on user-item ratings. However, the natural sparsity of user-item ratings can be problematic in many domains, limiting the ability to produce accurate and effective recommendations. Moreover, in some CF approaches only rating information is used to represent users and items, which can lead to a lack of recommendation explained. In this paper, we present a novel deep CF-based recommendation model, which co-learns users' abundant attributes. To better understanding the user's preference, we explore user deeper and unseen factors on the user-item ratings and user's side information by adopting the AutoEncode network. After that, we conduct the k-mean algorithm with extracted deep user factors to classify users. Then the user-side CF algorithm is employed to produce the recommendation list based on the classification results, for alleviating recommendation speed. Finally, we conduct lots of experiments on real-world datasets. Compared with state-of-the-art methods, the results show that the proposed method has a significant improvement in recommendation performance, in terms of recommendation accuracy and diversity. Furthermore, it also enjoys high effectiveness, and the approach is useful when it comes to assigning intuitive meanings to improve the explainability of recommender systems.

Keywords: Collaborative filtering algorithm · Recommendation system · K-means++ · Clustering

1 Introduction

With the rapid development of the Internet and mobile technologies, the recommender system has become an essential part of e-business applications, which can help people to find the potential interesting information and services [1]. Collaborative Filtering (CF) [2, 3] is the most popular approach in RS and has received a great deal of attention in industry, such as Amazon, Netflix, Taobao, and so on. Generally, most CF approaches

Z. Huang et al. (Eds.): WISE 2020, LNCS 12343, pp. 331–342, 2020.
https://doi.org/10.1007/978-3-030-62008-0_23

rely on user-item ratings that predict the users' preferences based on the users or the items having similar ratings. In this kind of RS, a typical matrix of user-item ratings is exploited to compute similarities between users (user-based) or items (item-based), then make a prediction based on the computed user/item similarities.

However, the UCF algorithm also has several limitations such as low scalability when dealing with large amounts of data [4], and the problem of a cold start. Further, the traditional CF algorithm needs to compute the similarities of the increasing number of users to all other users, and it requires higher computation efficiency. It is a significant challenge to improve computation speed for an online recommender system. Also, the number of users and items is vast; however, most users just rate a small part of items, so the data used to calculate similarities between users and items is sparse. Finally, it comes to the condition that the recommendation results may not be satisfactory.

In fact, many social media are obtaining user side information when they register. It is an effective way to deal with the cold start problem. Therefore, the combination of item interaction information and user side information can get a better recommendation effect theoretically.

In recent years, deep learning has made breakthrough progress in image processing [5], natural language processing [6], and speech recognition [7]. Meanwhile, deep learning has a subversive effect on the recommendation system, which brings more opportunities to improve the recommendation performance. However, neither of them has modeled the product's comments and user's auxiliary information simultaneously.

In this paper, we present a novel collaborative filtering (CF) method for a top-N recommendation named Autoencoder and K-means++ in Collaborative Filtering (AK-UCF) AK-UCF generalizes several previously mentioned clustering technology, deep learning technology, and user attribute information. But its structure is much more flexible. For instance, it is easy to incorporate nonlinearities into the model to achieves better top-N recommendation results. We compare the performance of AK-UCF with other collaborative filtering methods in different data sets. Experimental results show that AK-UCF consistently outperforms other recommended algorithms by a significant margin on a number of common evaluation metrics.

Our contributions can be summarized as follows:

- In the cluster stage, we use deep learning technology to reduce the dimension of the scoring matrix and improve the problem of the sparse scoring matrix.
- We use the clustering data for collaborative filtering recommendation and reduce the time consumption of collaborative filtering recommendation.
- We use the user side information to improve the cold start problem in the recommendation system and collaborate to produce user portraits.

The remainder of this paper is organized as follows. Related works to our contributions are presented in Sect. 2. The implementation details of our AK-UCF method are shown in Sect. 3. The datasets are described and the performance is analyzed via experiments in Sect. 4. Finally, we summarize our paper with some concluding remarks in Sect. 5.

2 Related Works

We describe the current situation of methods Auto-Encoder dimensionality reduction algorithm, K-means++ clustering algorithm, and user side information used in AK-UCF.

2.1 Auto-Encoder

Literature [8] proposes to use AutoEncoder to extract the compressed representation of users and projects in the scoring matrix. As a deep feature of users and projects, the extracted features are used for scoring prediction. Experiments prove that the number of RMSE indicators is better than traditional models such as collaborative filtering. On the other hand, literature uses an automatic encoder that does not extract the deep features of the user. It can be considered to use a stack-type noise reduction encoder, so that deep feature vectors can be obtained and the recommendation quality can be improved.

The CDAE model [9] takes the row of the user-item evaluation matrix as input, obtains the hidden representation of the user through a layer of neural network coding, and restores the user's interaction behavior through a layer of neural network. Unlike the simplest Autorec model, the CDAE model incorporates user-specific considerations when coding for hidden representations, with more semantics. In order to make the model more robust, the CDAE model performs noise processing on the input features, either by dropout or by adding Gaussian noise. A common shortcoming of both approaches is that they do not take into account user side information.

2.2 K-Means++

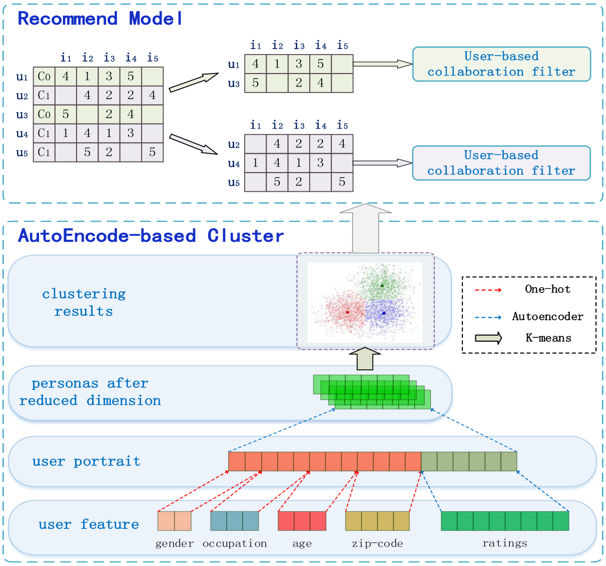
There has been diverse research to enhance recommendation accuracy by means of clustering methods. In [10], CF and content-based filtering methods were conducted by finding similar users and items, respectively, via clustering, and then a personalized recommendation to the target user was made. As a result, improved performance on the precision, recall, and F1 score was shown. Similarly, as in [10], communities (or groups) were discovered in [11] before the application of matrix factorization to each community. In [12], social activeness and dynamic interest features were exploited to find similar communities by item grouping, where items are clustered into several groups using cosine similarity. As a result of grouping, the K most similar users based on the similarity measure were selected for recommendation. The performance of user-based CF with several clustering algorithms including K-means++, self-organizing maps (SOM), and fuzzy C-Means (FCM) clustering methods was shown in [13]. It was shown that K-means++ user-based CF has the best performance in comparison with user-based CF based on the FCM and SOM clustering methods.

2.3 User Side Information

In this paper, we mainly consider some basic information, like users' age, gender, and occupation. Further, consider the deep statistical information of users and items, user-item rating matrix. We directly integrate the side information and the rating information of users in the deep neural network. Through combining the two parts, we jointly build the recommendation model to fully characterize the interaction between users and items.

3 Methodology

For exploiting the full advantage of the available user side information, we propose a clustering-based collaborative filtering recommendation algorithm via deep learning user side information, which uses the Auto-Encoder and K-means++ in User-based Collaboration Filter algorithm. The model consists of two-stage. As shown in Fig. 1.



		i_1	i_2	i_3	i_4	i_5
u_1	C_0	4	1	3	5	
u_2	C_1		4	2	2	4
u_3	C_0	5		2	4	
u_4	C_1	1	4	1	3	
u_5	C_1		5	2		5

	i_1	i_2	i_3	i_4	i_5
u_1	4	1	3	5	
u_3	5		2	4	

	i_1	i_2	i_3	i_4	i_5
u_2		4	2	2	4
u_4	1	4	1	3	
u_5		5	2		5

Fig. 1. The architecture of AK-UCF model

3.1 Auto-Encoder-Based Clustering

In the clustering stage, we use the user's characteristic information to add user category information to the user. We believe that the interests of users with the same attributes are also similar. The user feature information used in our model includes the user's gender, age, occupation, geographic location. According to experience, we divide the user feature information into corresponding number categories. For example, we divide gender data into two categories.

The categories are independent of each other, so we use one-hot encoding to encode this information. For example, the one-hot encoding value for male gender is [1, 0], and the encoding value for female gender is [0, 1]. Finally, a user feature information code $\boldsymbol{f}_u = [\boldsymbol{g}_u, \boldsymbol{o}_u, \boldsymbol{a}_u, \boldsymbol{z}_u]$ with d_f-dimension is obtained. Where $\boldsymbol{g}_u$, $\boldsymbol{o}_u$, $\boldsymbol{a}_u$, $\boldsymbol{z}_u$ represent the One-Hot encoding of gender, age, occupation, and geographic location respectively.

Another important feature information must be considered when clustering users: user-score information. List user rating information into the matrix as Table 1.

Use the rating vector for all items of user u_i to represent the rating features $\boldsymbol{r}_u$: $[r_{ui,v1}, r_{ui,v2}, \ldots, r_{ui,vn}]$. Where r∈{0, 1, …, 5} donate users' ratings for movies. AE is used to

Table 1. Information of user-item rating

	v_1	v_2	…	v_n
u_1	$r_{u1,v1}$	$r_{u1,v2}$	…	$r_{u1,vn}$
u_2	$r_{u2,v1}$	$r_{u2,v2}$	…	$r_{u2,vn}$
…	…	…	…	…

reduce the high dimensional sparse vector $\boldsymbol{r}_u$. The sample data $\boldsymbol{r}_u$ of the AE is encoded by the encoder function f to obtain the coding feature $r_u^{(n)}$, and $\boldsymbol{r}_u$ and $r_u^{(n)}$ satisfy the following Eq. 1:

$$\begin{aligned} r_u^{(1)} &= f_{\theta_1}(r_u) = s(W_1 r_u + b_1) \\ r_u^{(2)} &= f_{\theta 2}(r_u^{(1)}) = s(W_2 r_u^{(1)} + b_2) \\ &\cdots \\ r_u^{(n)} &= f_{\theta n}(r_u^{(n-1)}) = s(W_n r_u^{(n-1)} + b_n) \end{aligned} \tag{1}$$

where: s is a neural network excitation function, generally using a nonlinear function such as sigmoid Function; $\theta = \{W, b\}$ is a set of parameters. Then pass the following Eq. 2:

$$\begin{aligned} \hat{r}_u^{(n)} &= g_{\theta_n}(r_u^{(n)}) = s(W_n' r_u^{(n)} + b_n') \\ \hat{r}_u^{(n-1)} &= g_{\theta_{(n-1)}}(\hat{r}_u^{(n)}) = s(W_{(n-1)}' r_u^{(n-1)} + b_{(n-1)}') \\ &\cdots \\ \hat{r}_u &= g_{\theta_1}(\hat{r}_u^{(1)}) = s(W_1' \hat{r}_u^{(1)} + b_1') \end{aligned} \tag{2}$$

Converting the coded d_r-dimension vector $r_u^{(n)}$ into a reconstructed representation of the original input $\boldsymbol{r}_u$, Eq. 3 is the optimization goal of AE.

$$L = \left\| r_u - \hat{r}_u \right\|^2 \tag{3}$$

By continuously correcting the parameters θ, the average reconstruction error L is minimized, and the obtained $r_u^{(n)}$ can be considered to retain most of the information of the original sample, the equivalent feature of the sample $\boldsymbol{r}_u$.

Add the rating features $\boldsymbol{r}_u^{(n)}$ to the feature information code to get personas $\boldsymbol{p}_u$= [$\boldsymbol{f}_u$, $r_u^{(n)}$] with d_p-dimension. Where d_p= d_f+ dr. The dimension of the user profile may be too high to use K-means++ clustering algorithm. Then use Auto-encoder again to reduce the dimension of the user profile from d_p-dimension to d_u-dimension.

In order to solve the time-consuming problem of the collaborative filtering algorithm, we cluster the obtained user portraits. Since the category of each user is not known in advance, we use an unsupervised K-means++ clustering algorithm. Each user portrait is taken as a cluster sample, and a user portrait vector is randomly selected as the first cluster center. K-means++ algorithm is used to calculate K categories of users.

The K value is related to specific datasets and usually determined by approaches based on Silhouette Coefficient [14] or Elbow method [15]. In our work, we will adjust the K value to meet our specific requirements. It has been shown that the initial clustering centers should be selected uniformly to get a good clustering result [16]. Thus, we will use the K-means++ algorithm given by D. Arthur to determine the initial cluster center. The K-means++ algorithm is shown in Algorithm 1.

```
Algorithm 1: The K-means++ algorithm
Input: The amount of cluster K.
Output: k initial cluster center $c_1, c_2, ..., c_k$.
Choose a user randomly as the first initial cluster center $c_1$.
for all i = 2:k do
    Calculate the shortest distance $D(u)$ between each user and all current cluster centers.
    Sample every user $u \in U$ with probability $P(u)$ as the next cluster center $c_i$.
        $P(u) = \frac{D(u)^2}{\sum_{u \in U} D(u)^2}$
end for
for $c_i$ changes do
    For each user in the set, calculate the distance from each center, and classify this user as the nearest center.
    For each category $c_i$, recalculate its cluster center $c_i = \frac{1}{|c_i|} \sum_{x \in c_i} x$ (which is the center of mass of all samples belonging to the class).
end for
```

3.2 Recommended Model

UCF algorithm mainly includes two steps as follows:

Calculate Interest Similarity Between Users. Given user u, v which belongs to the same cluster as user u, $N(u)$ represents a collection of items where u has had positive feedback, and $N(v)$ represents a collection of items where v has had positive feedback. To calculate the similarity between users based on the user-item rating matrix, we use cosine similarity to calculate the similarity, which is expressed as follows Eq. 4:

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}} \tag{4}$$

In the traditional UCF recommendation system, the user similarity needs to be calculated for any two users, so the time complexity is $O(|U|*|U|)$, which is very time-consuming when the number of users is large. Our AK-UCF improves this method. It calculates user similarity in the same category instead of calculating user similarity in all users. For example, K-means++ clustering algorithm divides users into K clusters on average in the preprocessing stage, and the time complexity of calculating user similarity in each cluster is O(($|U|/K)^2$). So the overall time complexity is O($|U|^2/K$). AK-UCF is of great significance in solving the time-consuming problem in the recommendation stage of the traditional UCF recommendation system.

Find the Items that the Target User May Like from the Items that the Users with Similar Interests have Interacted With. After obtaining the interest similarity between users, our AK-UCF will recommend K items that the user has the most similar interests to the user. The Eq. 5 measures the interest of user u in the AK-UCF algorithm for items.

$$p(u, i) = \sum_{v \in S(u,K) \cap N(i)} w_{uv} r_{vi} \tag{5}$$

where S(u, K) donates the K users who are most similar in interest to user u, $N(i)$ is the set of users who have interacted with item i, w_{uv} is the similarity of interest between user u and user v, r_{vi} represents user v's interest in item i, because it uses implicit feedback data of a single behavior, so all r_{vi}=1. In the top-N recommendation algorithm, we first rank the user u's interest in the items and then recommend the first n items that user u have not interacted with to the user u.

In the method of solving the cold start problem, we calculate the distance between the feature embedding provided by the new user u and K clustering centers and find the nearest clustering center which is the category c_u of the new user u. Then the top n items with the highest popularity that users in cluster c_u have ever interacted with are recommended to user u. The popularity of item i (P_i) is calculated as Eq. 6:

$$P_i = \sum_{v \in c_u} I_{N(v)}(i) \tag{6}$$

where $I_{N(v)}(i)$ is the indicator function. When $i \in N(v)$, the value of $I_{N(v)}(i)$ is 1, else the value of $I_{N(v)}(i)$ is 0.

4 Experimental Results and Analysis

4.1 Datasets

Movielens is a rating-based movie recommendation system, created by the GroupLens research team of the University of Minnesota which is specifically used to study recommendation technology. This article uses the Movielens_100k dataset and Movielens_1m dataset to verify the performance of the algorithm. These datasets are recognized as the main datasets for evaluating recommended algorithms. The information on these two datasets is shown in Table 2:

Table 2. Statistics of dataset

Dataset	Number of users	Number of movies	Number of ratings	Rating range	Least number of one user interacted with	Sparsity
Movielens_100k	943	1,682	100,000	1–5	20	93.69%
Movielens_1m	6,040	3,900	1,000,209	1–5	20	95.75%

4.2 Measurement

We compare our AK-UCF with the baseline models through the evaluation indicators Precision, Coverage, MAP, and running time which are commonly used in the recommendation system.

Precision. Precision is a measure widely used in the field of information retrieval and statistical classification to evaluate the quality of results. At present, it is widely used in the evaluation of the TOP-N recommendation system. Precision is the ratio of the number of correct items recommended and the number of all recommended items. The quality of the recommended results can be judged by the Precision.

Coverage. Coverage describes the ability of a recommendation system to explore the long tail of items. Coverage has different definitions of methods. The simplest definition is the proportion of items that the recommendation system can recommend to the total set of items. It is an important index to measure the novelty of recommendation results.

Mean Average Precision (MAP). In Eq. 7, Average Precision (AP) is a ranked precision metric that gives larger credit to correctly recommended items in top ranks. AP@N is defined as the average of precisions computed at all positions with an adopted item, namely.

$$AP@N = \frac{\sum_{k=1}^{N} \Pr ecision@k \times rel(k)}{\min\{N, \left|C_{adopted}\right|\}} \tag{7}$$

where Precision(k) is the precision at cut-off k in the top-N list $C_{N,\,\mathrm{rec}}$, and rel(k) is an indicator function equaling 1 if the item at rank k is adopted, otherwise zero. Finally, Mean Average Precision(MAP@N) is defined as the mean of the AP scores for all users.

4.3 Analysis of Experimental Results

To test our proposed method is efficient, three algorithms are compared. The first compared algorithm is traditional UCF. The second compared algorithm is User-based Collaborative Filtering with K-means++ clustering (K-UCF). And the third compared algorithm is CDAE as we mentioned in Sect. 2.

For every user from the input set their ratings were divided into training (80%) and testing parts (20%). All the models have some parameters to tune. In our experiment, The dimension of rating features $\boldsymbol{r}_u$ was reduce from 3952-dimension to 80-dimension. The dimension of the user profile $\boldsymbol{p}_u$ from 209-dimension to 20-dimension. Individual users are divided into four clusters in K-UCF and AK-UCF.

The impact of the number of movies recommended on recommendation quality: Firstly, this paper will set the number of users to be 20, set the number of movies recommended to be the independent variables, and set precision and MAP to be the dependent variable. Figure 2 compares Precision results on UCF, K-UCF, CDAE, and our AK-UCF. And Fig. 3 compares MAP results on UCF, K-UCF, CDAE, and our AK-UCF.

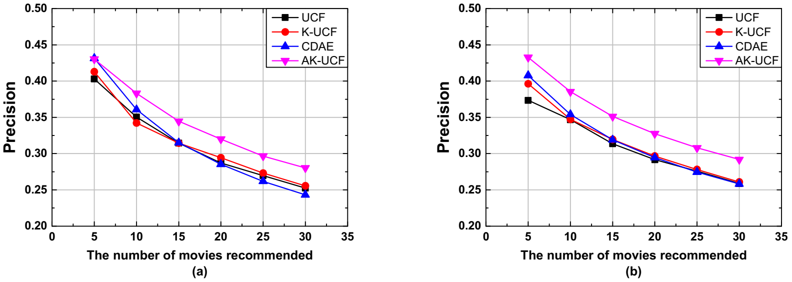


Fig. 2. Comparing Precision results about UCF, K-UCF, CDAE, and our AK-UCF. (a) Experiment on Movielens_100k dataset. (b) Experiment on Movielens_1m dataset.

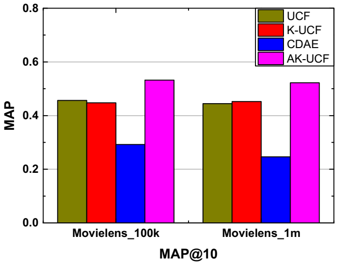


Fig. 3. Comparing MAP results on UCF, K-UCF, CDAE, and our AK-UCF.

In Fig. 2, the x-axis denotes the number of movies recommended is 5, 10, 15, 20, 25, 30. Obviously, the Precision of the other three algorithms is lower than our AK-UCF, which means our AK-UCF gains better performance than UCF, K-UCF, and CDAE. Besides, compared to the results in the UCF, the Precision increased by 9% in the Movielens_100k dataset and 16% in the Movielens_1m dataset. In Fig. 3, the x-axis denotes experiments on the UCF, K-UCF, CDAE, and our AK-UCF algorithms in Movielens_100k and Movielens_1m. The MAP results are obtained by recommending 10 movies to each user. The MAP value of our AK-UCF is higher than other baseline models in the two datasets, which means our model is more accurate and sensitive to the order of recommendation at the same time.

The impact of the number of movies recommended on recommendation diversity: Our results of the diversity of recommendation systems through Coverage are represented in Fig. 4.

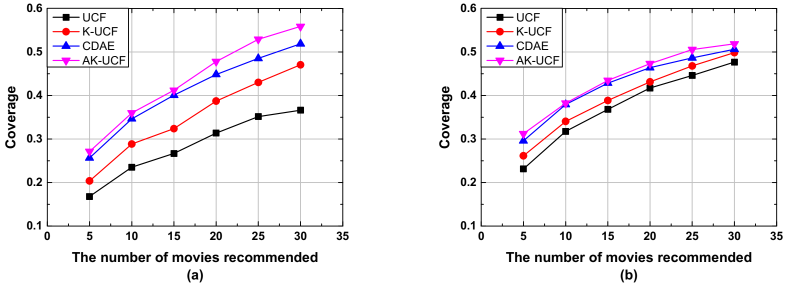


Fig. 4. Comparing Coverage results about UCF, K-UCF, CDAE, and our AK-UCF. (a) Experiment on Movielens_100k dataset. (b) Experiment on Movielens_1m dataset.

In Fig. 4, the x-axis denotes the number of movies recommended is 5, 10, 15, 20, 25, 30. The coverage value of our AK-UCF is higher than other baseline models in the two datasets, which means our model not only gets persona recommendation quality but also gets high recommendation diversity. The user portrait for this phenomenon is that products with low popularity become more popular in a small range of user clusters.

Figure 5 shows the difference in the running time of distinct recommendation models in the recommendation process. K-UCF and AK-UCF both run collaborative filtering algorithms in the same cluster while the number of clusters K is equal to 4, so the results are similar. As we can see from Fig. 5, although the recommended quality of CDAE algorithms is better than UCF and K-UCF, the time consumption is the most. In comparison, our algorithm is not only better than the other three baseline algorithms in terms of recommendation quality, but also consumes the least time in the recommendation process.

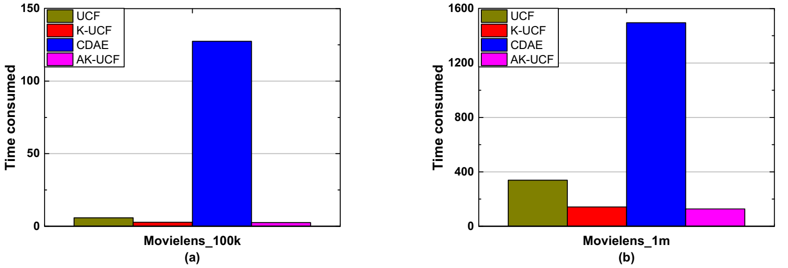


Fig. 5. Comparing running time results on UCF, K-UCF, CDAE, and our AK-UCF.

5 Conclusion

A multi-task recommendation system based on collaborative filtering is proposed in this paper. Firstly, user category information is derived from the user's auxiliary information and user-item rating matrix in the data preprocessing stage. In the recommendation stage, we only calculate the most similar users among the same cluster users. Compared with the traditional collaborative filtering algorithm, K-means++ clustering only recommendation algorithm, deep learning only algorithm, our algorithm has a great improvement in recommendation quality and time consumption in the recommendation process.

The future work includes tuning the algorithm based on an online experiment and trying other clustering algorithms such as Mixture-of-Gaussian clustering to optimize the collaborative filtering algorithm.

Acknowledgments. This work was supported by the Chongqing Research Program of Technology Innovation and Application under grants cstc2019jscx-zdztzxX0019, in part by Chongqing Natural Science Foundation under grants cstc2018jcyjAX0047, and Youth Innovation Promotion Association CAS, No. 2017393.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Konstan, J.A., Riedl, J.: Recommender systems: from algorithms to user experience. User Model. User-Adap. Inter. **22**(1–2), 101–123 (2012)
3. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 12 (2009)
4. Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering. In: International Conference on Computer and Information Technology. IEEE, Dhaka (2002)
5. Gu, F., Zhang, H., Wang, C.: A two-component deep learning network for SAR image denoising. IEEE Access **8**, 17792–17803 (2020)
6. Deep Learning for Natural Language Parsing: S. Jaf, C. Calder. IEEE Access **7**, 131363–131373 (2019)
7. Khalil, R.A., Jones, E., Babar, M.I., Jan, T., Zafar, M.H., Alhussain, T..: Speech emotion recognition using deep learning techniques: a review. IEEE Access **7**, 117327–117345 (2019)
8. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: AutoRec: autoencoders meet collaborative filtering. In: WWW 2015 Companion Proceedings of the 24th International Conference on World Wide Web, pp. 111–112 (2015)
9. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-N recommender systems. In: WSDM 2016 Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (2016)
10. Huang, C.-L. Yeh, P.-H., Lin, C.-W., Wu, D.-C.: Utilizing user tag-based interests in recommender systems for social resource sharing websites. Knowl. Based Syst. **56**, 86–96 (2014
11. Yin, B., Yang, Y., Liu, W.: Exploring social activeness and dynamic interest in a community-based recommender system. In: Proceedings of the 23rd International Conference World Wide Web, Seoul, Korea, pp. 771–776 (2014)

12. Guerraoui, R., Kermarrec, A.-M., Patra, R, Taziki, M.: D2P: distance-based differential privacy in recommenders. VLDB **8**(8), 862–873 (2015)
13. Koohi, H., Kiani, K.: User based collaborative filtering using fuzzy c-means. Measurement **91**, 134–139 (2016)
14. Rousseeuw, P.J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**(1), 53–65 (1987)
15. Ketchen, D.J., Shook, C.L.: The application of cluster analysis in strategic management research: an analysis and critique. Strat. Manage. J. **17**(6), 441–458 (1996)
16. Arthur, D., Vassilvitskii, S..: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp. 1027–1035 (2007)