International Journal of Hybrid Information Technology
Vol.9, No.4 (2016), pp. 1-10
http://dx.doi.org/10.14257/ijhit.2016.9.4.01

Research on User Clustering Collaborative Filtering Algorithm

Lihua Tian [1,2] , Liguo Han [1] and Junhua Yue [3]

[1]College of geoexploration science and technology Jilin University, Changchun 130012, china
[2]College of optical and electronical information, Changchun university of science and technology, Changchun 130012, china
[3]Jilin Jianzhu University, Changchun 130012, china
[1,2]lihua_tian18@sina.com , [1]68854058@qq.com and [3]529388235 @qq.com

Abstract

Memory-based CF algorithms have the weakness of low real-time ability and scalability. For these issues, a SVD-based K-means clustering CF algorithm is proposed. Traditional clustering-based CF algorithms have low recommendation precision because of data sparsity. So we first fill the missing ratings by SVD prediction, and then implement k-means clustering in the filled matix. This algorithm overcomse the data sparsity issue via SVD and keep the advantage of clustering, such as good real-time ability and scalability. Experiments results show that this algorithm outperforms Pearson CF, svd CF and k-means CF.

***Keywords**: Recommendation Systems, Collaborative Filtering, Clustering, Singular Value Decomposition*

1. Introduction

The algorithm based on the global came into birth earlier and is easily understood. The nearest search over the entire database makes recommendation results more accurate [1]. Therefore, it’s extensively applied in some existing collaborative filtering systems. But for the massive Web system, since user data volume is enormous, if we use the collaborative filtering algorithm based on memory, it needs to search the whole user space, which can’t guarantee instantaneity of the recommendation system; moreover, when user data change or there’s new coming user or item, the system has to re-search frequently the whole user space, leading to poor scalability. To overcome problems mentioned above, some scholars suggested model-based collaborative filtering approaches, combing machine learning techniques and artificial intelligence to improve classical collaborative filtering algorithms e.g. Bayesian network[2-3], clustering[4-6], neural network[7], SVD[8-9], latent topic analysis and so on. The common point of them is to forecast pre-trained models with the use of historical rating matrix. Those models can be built in an offline manner. For active users, we can predict them based on their matching with models. Such algorithms improve their scalability a lot. Besides, those methods can reduce effectively the dimension of user rating data, some of which can even alleviate data sparseness. On the contrary to traditional methods which compute directly user or item similarity, model-based approaches advance the effect of prediction. Also, since models are established offline, the burden of the recommendation system is relieved and online load is reduced.

In Bayesian network method, Miyahara and Pazzani et al[10]. firstly transformed user data to binary data and constructed a Boolean type rating feature matrix. Such transformation simplified Bayesian network learning and reasoning; however, for multi-class data, it lost user rating hierarchical information and impeded further improvement of model extendibility. In light most of user rating data are multi-class in the reality, literature [11] presented a Bayesian network for multi-class data recommendation. The experiment demonstrated that it realized better model scalability in spite of not so good prediction precision as the collaborative filtering based on Pearson correlation. [12] showed the collaborative filtering method of using together Bayesian network and decision trees. Each leaf node in Bayesian network stands for the possibility of rating. It generated one decision tree model for every leaf node to foresee user's marks. Through tests, it proved nearly as good recommendation accuracy as that based on Pearson correlation, but better than Bayesian-based clustering and cosine similarity method.

Clustering technologies have favorable scalability and usability to big data. So some researchers proposed clustering-based collaborative filtering models. By clustering, it's possible to narrow the range of searching the nearest neighbors to the target object to a few clusters which have the highest similarity with the target, cutting down effectively computing workload and increasing instant responsiveness. Sarwar et al[13]., O'Connor and Herlocker et [14]al. utilized clustering technologies to divide data into different classes; for each sub-class, they used traditional collaborative filtering methods to produce recommendations; Deng Ailin et al[15]. came up with the method based on item clustering, which generated different clustering centers based on user-item rating similarity; by calculating that similarity, they determined item belonging to which class and thus to produce recommendations. [16-18] employed improved clustering technologies to collaborative filtering and made excellent results. [19-20] introduced the co-clustering which is commonly used in bioinformatics and for text analysis to be used in collaborative filtering. With co-clustering of two dimensions—user and commodity as per the rating pattern in the original matrix, the impact by data sparsity was alleviated and the recommendation effect was enhanced.

In the paper, we proposed K-means clustering collaborative filtering algorithm based on SVD matrix completion technique. When data are very sparse, conventional clustering-based collaborative filtering methods have very low accuracy of prediction. The approach here uses firstly SVD descending dimension technique to make predictive completion of original hi-dimension sparse matrix to get one rating matrix without missing values; then, through K-means clustering, it clusters users in the completed data to finish the prediction of unknown ratings on the testing set. The algorithm overcomes sparsity problem by virtue of the latent relationship between user and item; simultaneously it keeps merits like offline modeling and good scalability, which are found in general clustering methods.

2. K-means Collaborative Filtering Algorithm based on SVD

2.1 Singular Value Decomposition

Singular value decomposition (SVD) is a common technique for matrix decomposition, able to replace feature extraction method and reveal deeply the internal structure of matrix[21-22]. Currently, SVD has been broadly applied in machine learning, statistical computation, information retrieval, image and signal processing, control theory and the like. To be specific in information retrieval, SVD's main technique is latent semantic index (LSI), which can project the hi-dimension expression of documents in vector space model to low-dimension latent semantic space, declining the retrieval complexity and on the other side attenuating data sparsity[23].

For a $m \times n(m > n)$ matrix R, the SVD can be decomposed into three matrices" : U, S, V

$$R = U \times S \times V^T \quad (1)$$

Where U is a $m\times m$ orthogonal matrix, satisfying $UU^T=1$,V is a $n\times n$ orthogonal matrix, satisfying $VV^T=1$, S is a $m\times n$ diagonal matrix, its diagonal elements greater than 0 and in accordance with the order of the order from the large to small.

Sarwar et al. applied SVD technique in collaborative filtering. They filled up missing values in rating matrix with user's average scores; then, perform SVD decomposition of rating matrix after pre-treatment. It works in the following procedure:

Algorithm 1 SVD algorithm

Input: matrix R

Output: matrix S, V, U

Step:

Setp1 : Substitute missing values in matrix R with the average value r_i of relevant column;

Setp2 :Replace the element r_{ij} in matrix R with $r_{ij}-r_i$ to get another matrix R'; perform singular value decomposition of R' to have three matrices $U^{'}, S^{'}, V^{'}$ ';

Setp3 : Simplify S' and replace values which are below 1 on their diagonal lines with 0; then, delete all lines or columns which are 0 to get one k-dimension diagonal matrix S;

Setp4 : Use matrix S to simplify U' ,$V^{'}$ to get U,V; then matrix R' is simplified to $R^{"}=U\times S\times V$; $R^{"}=U\times S\times V$

Setp5 : Compute $\sqrt{S}$ and get matrix $U\times\sqrt{S}$ and $\sqrt{S}\times V^T$;

In step (3) above, if original data is of high dimension, dimension k of the simplified matrix S is far less than original dimension n. After singular value decomposition, original data sparsity is greatly reduced and hence better recommendation precision is achieved.

SVD can be used to implement matrix decomposition in an offline way and work very well with collaborative filtering techniques, finding out easily characteristics of both users and items, based on that, recommendations are made. In practical recommendation system, tremendous user and item data often make prediction model much complicated; meanwhile, the prediction result is not satisfying due to existence of plentiful missing ratings. By descending dimension, it's possible to increase date density, solve data sparsity and discover latent information of data as well. The collaborative filtering algorithm based on SVD has better applicability. For each item, it can constitute different features in different ratios for different dimensions. Likewise, for each user, it can express different dimensions with different percentages according to user interest; finally based on the cross product of user feature vector and item feature vector, it can have user's rating of an item. It reveals that SVD method can achieve sound robustness and noise immunity through dimension reduction. With reference to significances of matrix decomposition and solving eigenvalue, it can find out user's potential hobbies. On the premise of assured recommendation quality, it can generate more diversified recommendation results.

K-means clustering

2.2 K-means Clustering

K-means algorithm is a very classical clustering method in machine learning and data mining. It was developed in 1967 by Macqueen. Because of its simplicity and efficiency especially good scalability for massive dataset, it's widely applied in pattern recognition, business data mining and optimization. It has the basic idea: through iteration, it divides dataset to k clusters, making bigger similarity of data in the same cluster and smaller among different clusters, the number of which needs being pre-determined.

K-means algorithm has wide applications owing to good points mentioned above. It includes these steps:

Algorithm2 K-means collaborative filtering algorithm

Input: scoring matrix R
Output: Top-N recommendation collection
Step:
Setp1: Utilize K-means to cluster rating matrix R and separate users to c clusters; distance function uses Pearson correlation or cosine similarity;
Setp2: For a currently active user a, compute distances among it and c class centers, c regarded as the cluster which is the closest to the class center;
Setp3: In the cluster to which user a belongs to, compute $sim(U_a, U_i)$ and choose k most similar users as a's nearest neighbors;
Setp4: Based on rating data of the nearest neighboring user set, make weighting prediction of ratings of user a's un-scored items;
Setp5: Choose Top-N of predicted ratings and output them.

The collaborative filtering method based on clustering avoids shortcomings with the traditional ones based on memory. Specifically, in methods based on memory, they need to search the closest neighbors to the target object in the whole item space, which will lead to huge computing amount and affect heavily efficiency. Through clustering technologies, the scope of searching the nearest neighbors to the target object can be limited to some clusters with the highest similarity to the target, calculated quantity reduced and real-time responsive capability improved, along with better scalability. When data are very sparse, cluster-based collaborative filtering methods get very low recommendation precision; contrarily, SVD performs better in dealing with high-dimension data. Hence in the paper, we combine good qualities of SVD and clustering: use SVD to predict original high-dimension sparse data and complete them; then perform K-means clustering of the matrix without missing values to generate recommendations to users.

3. Design of the Algorithm

Step 1 Utilize SVD to smooth original sparse matrix and fill up missing values

In training dataset, as per [2], target user u's predicted evaluating score of item i can be obtained by the following equation:

$$P_{u,i} = \bar{R}_u + U_k \times \sqrt{S_k}^{'}(u) \times \sqrt{S_k} V_k^{'}(i) \qquad (2)$$

$\bar{R}_u$ is rating mean value of all rated items by target user u; U, S and V is three matrices got after equation 1 calculated user-item rating matrix R; U_k, S_k, V_k refers respectively to output matrix after reduction of U, S and V. K is the data dimension preserved after SVD decomposition. The appropriate values can be selected by further experiments.

After all users' missing scores are calculated, we can get one complete user-item rating matrix R'.

Step 2 Cluster users in the completed dataset R'

The paper chooses k-means method for user clustering, which has merits of rapid clustering speed and easy implementation, suitable for processing large-scale data. It's used widely in the field of text mining and information retrieval. By k-means clustering, we can get several user modes $C_1, C_2, ..., C_j$ with similar rating habits; j is number of cluster and its value can be got by experiment tests.

Step 3 Predict user rating with testing dataset

When tested user a enters the system, we determine it's the closest to which class center based on its rating vector and assign it to the class $C_m (m \in [i, j])$ which is nearest to the center.

Next, when the nearest neighbor user a is calculated, it can only be searched on the user's subset C_m, saving the computing time, and the algorithm has good scalability. According to the similarity between the user a and its nearest neighbor, this paper uses the classical nearest neighbor prediction algorithm to predict the A for the unknown project score as:

$$P_{a,i} = \bar{R}_a + \frac{\sum_{y \in NBS} Sim(a,y) \times (R_{y,i} - \bar{R}_y)}{\sum_{y \in NBS} |Sim(a,y)|} \qquad (3)$$

4. Experiment Design and Discussion

4.1 Experimental Data

To validate the effectiveness of the proposed algorithm here, we tested with MovieLens dataset. We chose 100000 rating data about 1682 movies by 943 users, 80% of which used as training set and the rest 20% as testing set. As usual we recommend the commonest performance evaluation indicators: mean absolute error (MAE) and Precision

4.2 Experimental Results

First of all, in first step of the method, preserved dimension k is significant after decomposition of singular values. If k is too small, it will lose too much information and thus structural information of the original rating matrix can’t be acquired; contrarily, if k is too big, reducing the dimension becomes meaningless. On the regard, it’s necessity to determine properly the reserved dimension k.

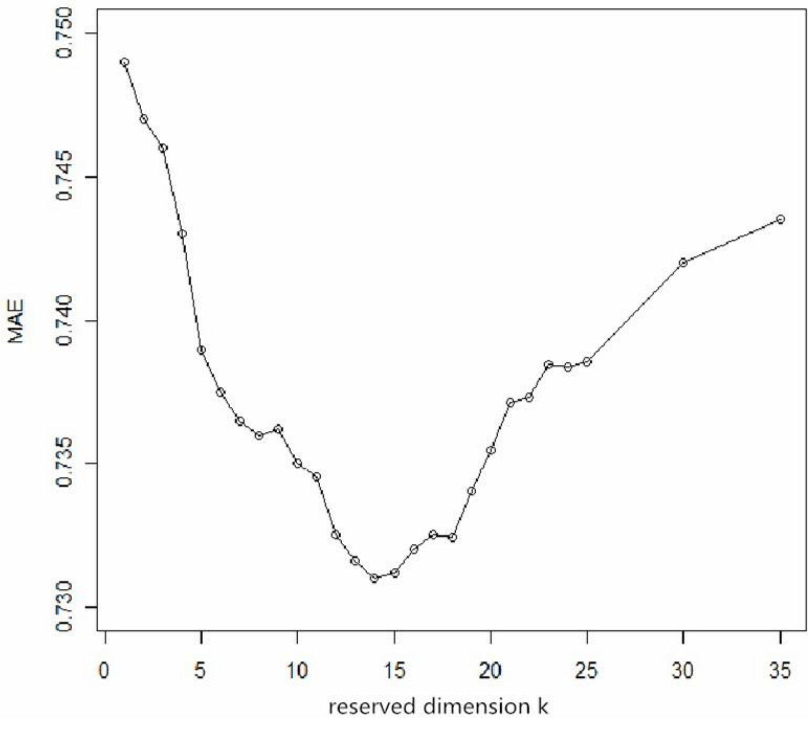

Figure 1. Different K Values Correspond to the MAE

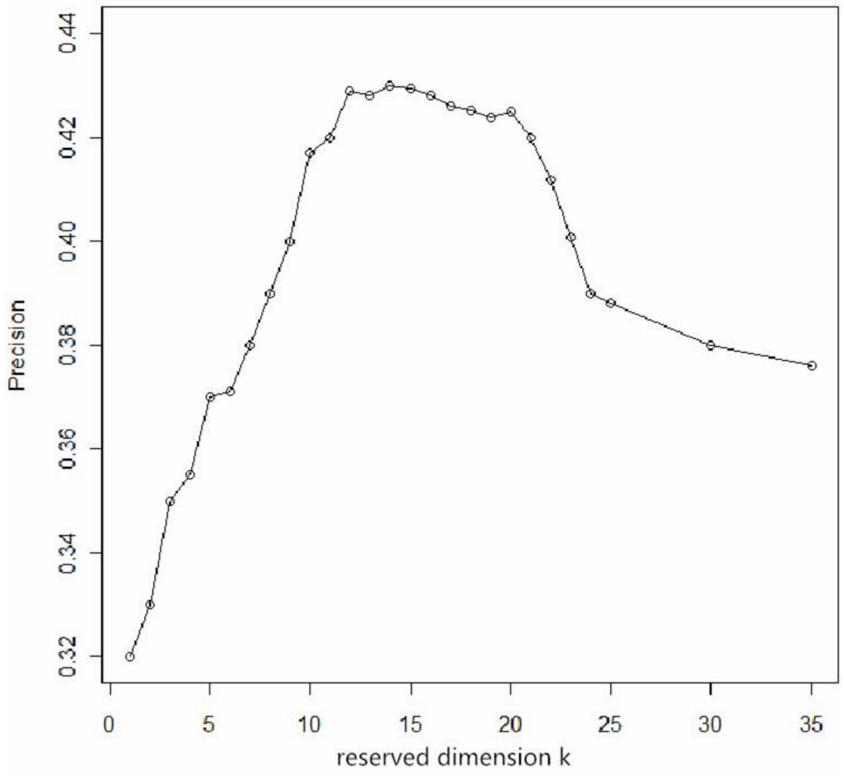

Figure 2. Different K Values Correspond to the Precision

Fig. 1 and 2 describes respectively for different values of k, how MAE and precision values change. As seen, when the preserved dimension k=14, MAE is minimum; when k is in the interval [12,16], the recommendation precision is better. For both MAE and precision, we find in the experiment when k=14, they reach the best effect. So when using SVD dimension reducing technique to predict completed matrix, we set reserved dimension k=14.

After SVD used to predict the completed original rating matrix, we can cluster users. Fig. 3 gives how the recommendation precision evaluating criterion MAE is influenced by varied number of clusters. [1] stated when the neighbor collection size was 20-50, the collaborative filtering algorithm performed the best. So we test MAE values for different number of clusters when neighbor collection size is 40. From Fig. 3, we conclude that the cluster number c should be the best among 12-20 for the tested dataset here. In later tests, we can try to make c=16. The point is in the proposed algorithm, SVD steps and clustering procedure are carried out independently. Therefore it’s not a matter of optimizing concurrently SVD reducing dimension k and cluster number c.

In the next, to verify the effectiveness of the mentioned method, we compare it with traditional collaborative filtering methods respectively based on Pearson correlation, SVD (k=14), and K-means (c=16). Results of the four methods on testing dataset for different number of neighbors are shown in figure4 and figure5.

From Fig.4, we note that the proposed algorithm reaches better MAE than Pearson and K-means collaborative filtering methods. When the number of neighbors is above 20, the method is superior over SVD-based one. From Fig. 5, the method gets higher precision rate than Pearson and K-means. When the number of neighbors is over 28, it outperforms SVD collaborative filtering. It’s concluded that the presented method not only acquired better predictive accuracy and keeps good scalability.

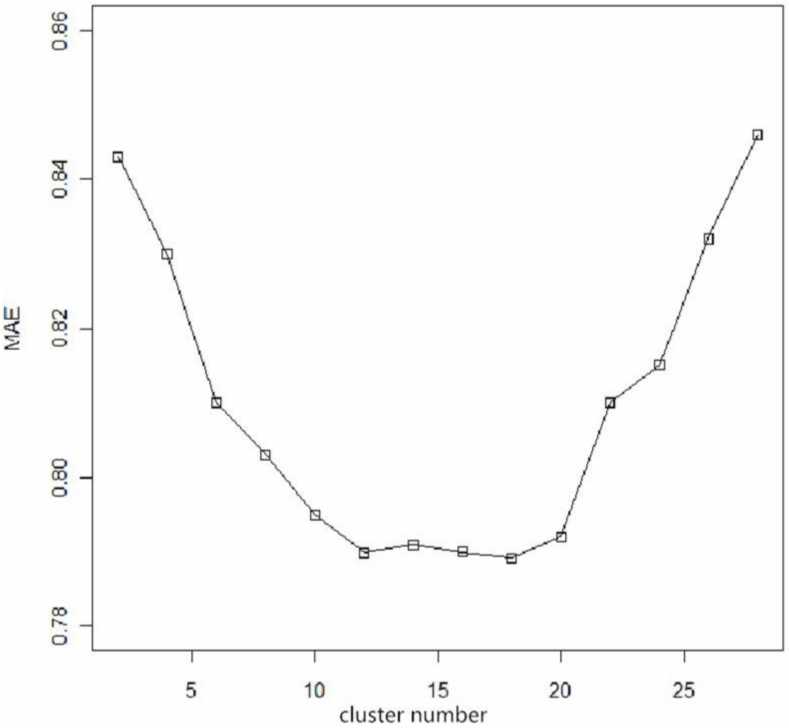

Figure 3. Comparison of Recommended Performance of Different Clustering Numbers

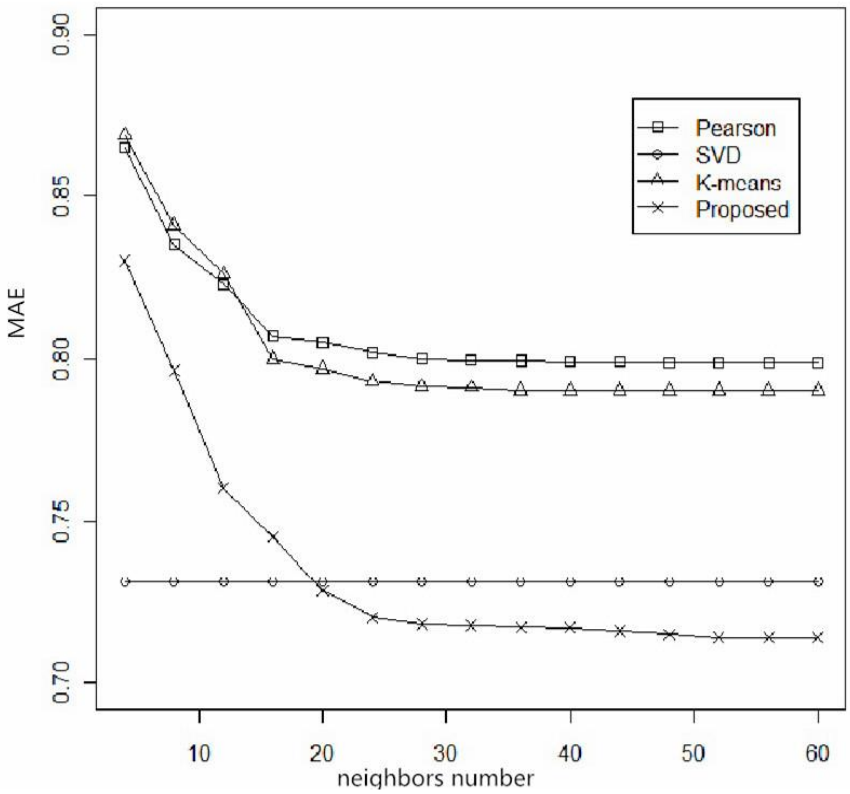

Figure 4. MAE Comparison of Different Collaborative Filtering Algorithm

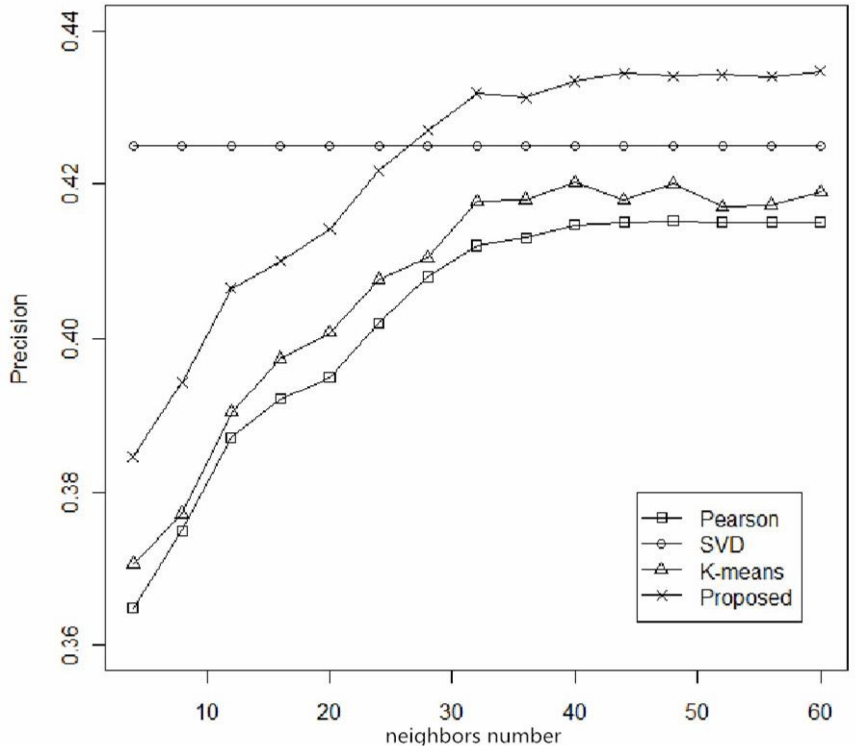

Figure 5. Precision Comparison of Different Collaborative Filtering Algorithm

5. Conclusion

In the paper we introduced a K-means clustering collaborative filtering algorithm based on SVD matrix completion technique. If data are too sparse, cluster-based collaborative filtering predicts very inaccurately. To address it, we used SVD dimension descending strategy to complete original data and increase data density as to get one rating matrix without missing values; then employed K-means clustering to cluster users in the completed data and thus finished prediction of unknown ratings on testing dataset. The method gets rid of data sparsity with SVD technique and inherits merits like rapid clustering speed and excellent scalability. Experiments confirmed its better predictive effect and scalability than conventional collaborative filtering based on Pearson correlation, SVD and K-means.

References

[1] P. Resnick, N. Iakovou, M. Sushak, “ GroupLens: An open architecture for collaborative filtering of netnews”, Proc 1994 Computer Supported Cooperative Work Conf , Chapel Hill , **(1994)**, pp. 175 -186

[2] Zhangfeng, “Using BP neural network to alleviate sparsity issue in collaborative filtering recommendation algorithm”, Journal of computer research and development, vol. 43, no. 4, **(2006)**, pp. 667-672

[3] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, “Application of Dimensionality Reduction in Recommender System-A Case Study”, ACM 2000 KDD Workshop on Web Mining for e-commerce-Challenges and Opportunities,Boston, MA, **(2000)**.

[4] G. Linden, H. B. Smit, J. York, “Amazon.com recommendations: Item-to-item collaborative filtering”, IEEE Internet Computing, vol. 7, no. 1, **(2013)**, pp. 76-80.

[5] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, “Application of Dimensionality Reduction in Recommender System-A Case Study”, ACM 2000 KDD Workshop on Web Mining for e-commerce-Challenges and Opportunities,Boston, MA, (2010).

[6] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, "Eigentaste:A constant time collaborative filtering algorithm", Information Retrieval, vol. 4, no. 2, **(2001)**, pp.133-151.

[7] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, **(2000)**, pp.679–689.

[8] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms", Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI '06), **(2006)**, pp. 497–504.

[9] J. S. Breese, D. Heckerman, C. Kadie, "Empirical analysis of Predictive Algorithms for collaborative filtering", Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, **(1998)**, pp. 43-52

[10] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Recommender systems for large-scale E-commerce: scalable neighborhood formation using clustering", Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT '12), **(2012)**.

[11] M. O'Connor and J. Herlocker, "Clustering items for collaborative filtering," in Proceedings of the ACM SIGIR Workshop on Recommender Systems (SIGIR '99), **(1999)**.

[12] Deng Ailin, Zhu Yangyong, "Recommendation algorithm for collaborative filtering based on project clustering", Mini microcomputer system, vol. 25, no. 9, **(2013)**, pp. 1665- 1670

[13] D. Feng, Z. Haiyan, J. Lihong, "Recommendation method of collaborative filtering based on fuzzy clustering", Computer simulation, vol. 22, no. 8, **(2012)**, pp. 144-148.

[14] F. R. Gao, W. Shan, "A sparse matrix partitioning based personalized recommendation algorithm", Computer and microelectronics, vol. 21, no. 2, **(2014)**, pp. 58 – 62.

[15] S. Duo, "The design of the recommendation system based on interest based clustering collaborative filtering system", Journal of Anhui University (Natural Science Edition), vol. 31, no. 5, **(2009)**, pp. 19-22

[16] T. George, S. Merugu, "A Scalable Collaborative Filtering Framework Based on Co-clustering", Proceedings of the 5th International Conference on Data Mining.Washington, DC: IEEE Computer Society Press, **(2005)**, pp. 625-628

[17] W. chao, W. Yongji, "two stage combined clustering collaborative filtering algorithm", Software Journal, vol. 21, no. 5, **(2010)**, pp. 1042-1054

[18] D. Kalman, "A Singularly Valuable Decomposition: The SVD of a Matrix", The College Mathematics Journal, vol. 27, no. 1, **(1996)**, pp. 2-23.

[19] G.H. Golub, C. F. Van Loan, "Matrix Computations (3rd edition)", Johns Hopkins University Press, **(1996)**.

[20] http://en.wikipedia.org/wiki/Singular_value_decomposition

[21] S. Deerwester, S. T. Dumais, G. W. Furnas, "Indexing by Latent Semantic Analysis", Journal of the American Society for Information Science, **(1990)**, vol. 41, no. 6, pp. 391-407

[22] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering", Proceedings of KDD Cup and Workshop 2007, San Jose, CA, USA, **(2007)**.

[23] M.H. Pryor, "The Effects of Singular Value Decomposition on Collaborative Filtering", Technical Report: PCS-TR98-338, **(1998)**.

Authors

Lihua Tian. She received her M.S degree from Changchun Institute of optics, Fine Mechanics and Physics, Chinese Academy of Sciences. She has been Ph.D student in College of geoexploration science and technology Jilin University. She is an associate professor in college of optical and electronical information, changchun university of science and technology. Her research interests include digital image processing.

Liguo Han. He received his Ph.D degree from Jilin University. He is a professor in college of geoexploration science and technology of Jilin university. His research interests include Inversion and imaging of the complex earthquake wave field digital image processing.

Junhua Yue. She received her Ph.D degree from Changchun Institute of optics, Fine Mechanics and Physics, Chinese Academy of Sciences. She is an associate professor in Jilin Jianzhu university. Her research interests include Data image processing.