Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 5, September 2017, pp. 610-619
DOI: 10.23940/ijpe.17.05.p6.610619

NE-UserCF: Collaborative Filtering Recommender System Model based on NMF and E^{2}LSH

Yun Wu, Yiqiao Li*, Ren Qian

College of Computer Science and Technology, Guizhou University, Guiyang, 550025,China

Abstract

With the rapid development of big data and cloud computing, recommender systems (RSs) have gained significant attention in recent decades. However, there are still many challenges and drawbacks existed in RSs, such as complex and high-dimensional data, low recommendation accuracy, time-consuming and low-efficiency, which to a large extent restrict its applications. Non-negative Matrix Factorization algorithm (NMF) is a matrix factorization algorithm which finds the positive factorization of a given positive matrix. It can eliminate invalid and redundant features in user-rating matrix (URM), reduce URM's dimension. Exact Euclidean Locality Sensitive Hashing (E^{2}LSH) is an advanced algorithm for solving the approximate or exact Near Neighbor Search in high dimensional spaces. It can cluster similar-interest users (SIUs) of URM efficiently. Therefore, the authors propose an improved recommender system model named NE-UserCF (NMF-E^{2}LSH-UserCF) based on NMF and E^{2}LSH to improve the quality and performance of recommendation. The authors first utilize the NMF to process original URM, get a new-URM without invalid and redundant features. Then use E^{2}LSH to cluster users in new-URM based on their interests and produce the similar-interest-user matrix (SIUM). The authors further process the Top-10 recommendations by adopting the user-based collaborative filtering algorithm (UserCF). Finally evaluate experimental results by analyzing metrics Precision, Recall, Coverage and Popularity. Experiments indicate that NE-UserCF proposed in this paper improves the quality of recommendation and has a good performance.

Keywords: NMF; E^{2}LSH; UserCF; Collaborative Filtering; Recommender Systems

(Submitted on April 8, 2017; Revised on July 10, 2017; Accepted on August 23, 2017)

1. Introduction

Recommender systems (RSs) are becoming increasingly important with the advent of the overload of information and data [17]. RSs provide people with a very efficient approach to find relevant contents and items. Presently, collaborative filtering recommendation algorithms are the most widely and extensively used. It can analyze the user behaviors and recommend the hidden and complicated information with a high popularity by collecting and processing users' behaviors [2,13]. However, there are various challenges and flaws in collaborative filtering recommendation, such as complex and high-dimensional data, low recommendation accuracy, time-consuming and low-efficiency.

To improve the quality and performance of recommendation, the authors put forward a novel recommendation system model, namely NE-UserCF, by making the utmost of NMF and E^{2}LSH. First, the authors adopt the NMF to solve the complexity, high dimension, and sparseness problem of original user-rating matrix (URM). This process produces a new URM by filtering out invalid and redundant features. Thus, the new URM has a lower dimension than original URM, which facilitates similar-interest users (SIUs) search. Then, the authors utilize E^{2}LSH to process the new URM. By building data indexes on the new URM, E^{2}LSH can find similarities between various users smoothly and rapidly, and further build similar-interest-user matrix (SIUM), which improves the accurately and efficiency of recommendation. Furthermore, the authors apply user-based collaborative filtering algorithm (UserCF) to SIUM instead of original URM and take the Top-10 strategy for recommendation. Regarding Precision, Recall, Coverage and Popularity as evaluation criterions, the authors finally analyze experimental results.

* Corresponding author.
E-mail address: leeyiqiao@qq.com.

The remaining part of the paper is organized as follows. Section II reviews relevant researches. The principles and applications of NMF and E²LSH are presented in Section III. In Section IV, the authors first focus on the holistic framework of NE-UserCF, then illuminate the algorithms the authors utilized in this paper. Section V illustrates the data sets the authors employed in this paper, proposes the evaluation metrics, analyzes the experimental results, and verifies the validity of NE-UserCF. Section VI summarizes the paper, and covers the future research prospects.

2. Related Works

Along with the rapidly development of cloud computing and big data, overload of data and information is increasingly obvious, which results in lower accuracy and efficiency of relevant contents and items search. Thus, RSs are of vital importance in human being's daily life [24]. Generally, RSs are used to predict users' affection for items and then recommend the most appropriate items (Top-N items) to users according to their affections. Prediction is the core part of RSs: the more precision of the predictions are, the more accuracy of recommendations will be [15]. User-based collaborative filtering recommendation technology evaluates the unknown item scores for target users to produce recommendation clusters by analyzing the activities of target users' nearest neighbors. Similarity calculation is an essential factor in the process of recommendation and can affect recommendation result and quality directly.

The original URM is generated from MovieLens. Rows represent users, columns represent movies, and values represent scores that users rate for movies. As the number of movies is hundreds and not every movie has a rated score (because they are not watched by users or users do not give a rating score), original URM is a high-dimensional and sparse matrix. In recent years, researchers and experts focus on URM's sparse problem and put forward different solutions. Xu et al. [21] extracted external information and calculated their Jaccard similarity to produce prediction values, which were filled into test data abstemiously to improve recommendation accuracy. However, this strategy left the high dimension problem unsolved. Wu et al. [20] proposed an algorithm that combined reasoning based on cases with collaborative filtering technology to solve data sparse problem. Hu et al. [6] put forward a Web recommendation method with time perception based on an improved collaborative filtering technology that is not sensitive to data sparse. They utilized a mixture personalized random walk algorithm to solve data sparse problem. Jianga et al. [8] used items online-browsing history as an extra information and designed a mixture collaborative filtering recommendation to solve data sparse problem. Bokde et al. [3] used matrix decomposition based on potential model to reduces matrix sparse. Zhang et al. [27] adopted non-negative matrix factorization to improve the recommendation quality, but it utilized traditional similarity measurement method, thus its recommendation precision was not satisfied.

Similarity measure is also one of the most important processes of recommendation algorithm, therefore similarity measurement method is extremely important. Many scholars and researchers have focused their research on this issue in recent decades. Gao et al. [16] suggested that users' similarities could be calculated personal correlation based on users' interests or angle cosine correlation based on users' interests. Collaborative filtering recommendation based on item clustering [19] and Collaborative filtering recommendation based on user clustering[1] could reduce time for nearest neighbor and improve similarity accuracy, however, clustering analysis had classification category multi-dimensionality and measurement control difficult, which further affected recommendation precision. Zeng et al. [25] obtained low-order approximate matrix by using singular value decomposition to complete collaborative filtering recommendation. Fang et al. [4] improved the method in [5] to reduce space dimension and improve the prediction accuracy, but its recommendation efficiency decreased largely when to process large scale and high-dimensional data.

Traditional recommendation methods either failed to solve the original URM's complexity, high dimension, and sparse, or utilized low efficiency similarity measurement methods. Even if in several papers, they proposed solutions to original URM's sparse, they used the traditional similarity measurement methods, which resulted in unsatisfied recommendation quality. Therefore, based on the analysis of traditional nearest neighbor collaborative filtering technology, the authors employ NMF to solve original URM's complex, high dimension, and sparse problems, and E²LSH to calculate the nearest neighbor users and cluster similar-interest users, which solve the low efficiency of traditional similarity measurement methods and improve recommendation quality.

3. NMF and E²LSH

3.1. NMF

Lee and Seung published the latest research results of Non-negative Matrix Factorization (NMF) in "Nature" in 1999. NMF is able to decompose a large matrix V ($a_{m\times n}$ **Error! Reference source not found.** non-negative matrix) into two small

matrixes with non-negative elements. That is to say, it will produce non-negative matrixes W and H , and meanwhile, they satisfy $V = WH$. This can be further described as below：

$$V \approx WH, W \in R^{m \times r}, H \in R^{r \times n} \tag{1}$$

Usually， $r << \min(m,n)$. W is called the base matrix, H is called the coefficient matrix. Then W can be used to present the original V , which can reduce the dimensions successfully and save the storage space effectively.

NMF has a variety of decomposition methods, but it usually uses an improved matrix decomposition method based on probability. It includes iterative additive decomposition method and iterative multiplicative decomposition method [22]. The base matrix W and the coefficient matrix H then can be further described as below:

$$W_u = \max(0, W_u + \partial \times [H_i \times (r_{ui} - W_u \cdot H_i)]] \tag{2}$$

$$H_u = \max(0, H_u + \partial \times [W_i \times (r_{ui} - W_u \cdot H_i)]] \tag{3}$$

Formula 2 and formula 3 is a constrained stochastic gradient descent iterative formula. In each iteration, it will select the maximum from 0 to a new value, thus it can guarantee the results being non-negative of each iteration. This method is widely used in matrix decomposition.

Xu et al. [14] utilized an improved NMF in the text clustering and obtained a good experimental result. Huang X et al. [7] presented the application of NMF in medical literature search and enhanced the clustering capability of biomedical documents greatly. Mehmood A et al. [12] introduced the application of NMF in seismic data analysis and detection, of which the performance was very promising. Li et al. [11] adopted NMF to mining the web mail information and the method proposed was able to describe user behavior patterns more intuitively and conveniently. Wang et al. [18] utilized the improved NMF based on weight to mining the face information and was used in human face recognition system.

3.2. E²LSH

Locality Sensitive Hashing (LSH) is one of the most popular methods in Approximate Nearest Neighbor (ANN) search, which is similar to the index technology and is mainly used to accelerate the search process and handle the high-dimensional data sets. LSH is the best solution to deal with the c-Approximate Neighbor Search problem and can avoid the "dimension disaster" problems of the traditional index method [23]. E²LSH improves LSH on $p-Stable$ distribution. It calculates the Euclidean distance directly and solves the (R,c) -Approximate Neighbor problem. The $p-Stable$ is defined as below.

If a distribution D on the set of real numbers R is a $p-Stable$ distribution, then the follow requirement should be meet:

For any n real numbers $v_1, v_2, ..., v_n$ and any n independent random variables $X_1, X_2, ..., X_n$, random variables $\sum_i v_i X_i$ and $(\sum_i |v_i|^p)^{1/p} X$ ($p \geq 0$) should have the same distribution.

E²LSH is locally sensitive. Assume that the distance of v_1 and v_2 is very close, then they are likely to have the same hash values and the probability of being put into the same bucket might be very high[29]. According to the principle of $p-Stable$ distribution, the mapping distance of v_1 and v_2 , $a \cdot v_1 - a \cdot v_2$ and $\| v_1 - v_2 \|_p X$ should have the same distribution. The distance of E²LSH is below:

$$h(v) = \left\lfloor \frac{\alpha \cdot v + \beta}{\varpi} \right\rfloor \tag{4}$$

Where ⌊ ⌋ denotes rounded down, α is d -dimensional random vector that meets the requirement of $p-Stable$ distribution, β is a random variable that is distributed evenly in $[0, \varpi]$.

E²LSH is able to combine several position sensitive hash functions, show as below:

$$f = \{g : S \rightarrow U^k\} \tag{5}$$

Where $g(v) = (h_1(v),...,h_k(v))$, each data point $v \in R^d$ is mapped by $g(v) \in f$ to obtain a lower-dimensional vector a $(a_1, a_2,..., a_k)$,then use the hash function $hash_1$ and $hash_2$ to process the vector a , and finally establish a hash table and store data. $hash_1$ and $hash_2$ can be defined as below:

$$hash_1(a) = ((\sum_{i=1}^{k} r_i' a_i) \bmod m) \bmod s \tag{6}$$

$$hash_2(a) = (\sum_{i=1}^{k} r_i'' a_i) \bmod m \tag{7}$$

Among them, r' and r'' are random integers, S is the size of the hash table, m is a large prime number, usually it is set as $2^{32} - 5$. After the map process of $hash_1$ and $hash_2$, E²LSH stores the data which are both the map results of $hash_1$ and $hash_2$ into the same bucket to realize the space division of data points. Then, E²LSH is able to cluster the similar interest users. The users with the same or similar interest will be put into the same buckets; otherwise, users will be stored into different buckets.

Jiao et al. [9] utilized E²LSH to mining image information, which had a good performance. Zhang et al. [28] proposed an improved multi kernel learning method based on E²LSH and applied it in image information analysis. Li et al. [10] took the advantage of E²LSH to optimize the application of large-scale document index. Zhang et al. [26] put forward a multi kernel E²LSH-MKL algorithm, using E²LSH for clustering.

4. NE-UserCF Recommender System Model

4.1. NE-UserCF Framework

To solve the existed problems of the traditional collaborative filtering recommendation, in particular, the original URM's complexity, high dimension, and sparse problem, and traditional similarity measurement's low efficiency, inferior quality and low accuracy of recommendation, the authors propose a novel recommender system model named NE-UserCF based on NMF and E²LSH. NMF ensures the non-negative of matrix decomposition, which is significant to URM. More importantly, NMF is able to eliminate the invalid and redundant features, reduce the dimensions of URM, and speed up the user similarity search. Furthermore, E²LSH is a $p-Stable$ distribution LSH. It can conduct the similarity retrieval of high-dimensional data rapidly with high accuracy. Thus, it can be used to cluster users and produce the similar interest user matrix. The holistic framework of NE-UserCF is shown as Figure 1.

Data Preprocessing: Utilize DataPreprocess() function developed by C# to process the original MovieLens dataset. Translate the data format from [user_id, item_id, rating, timestamp] to [rating_for_item_1, rating_for_item_2, rating_for_item_3, … , rating_for_item_n]. This process will generate the original URM in which numbers present the preference degree: the greater number means larger preference degree, while the value zero means no watch record on this movie.

NMF Dimension Reduction Method (eliminate invalid and redundant features): As the number of movies that users watched is considerable large, the dimension of URM is very high. Worse still, there are records that are generated as minority or contain extraordinary low scores, all of which are useless for similar interest users search and can be ignored. Thus, it is seriously necessary to filter out these meaningless features and reduce the dimension of the URM. By utilizing NMF, the authors can eliminate the invalid and redundant features successfully, maintain the paramount and useful information of URM. Therefore, it can facilitate the similarity search and improve the recommendation quality.

E²LSH Clustering Method (similar-interest-user matrix): By adopting E²LSH with the capability of similarity search on large scale and high dimension data, the authors cluster the new URM which is produced by NMF Dimension Reduction Method, establish indexes and search for the nearest neighbors (R-NN), and eventually build up the similar-interest-user matrix.

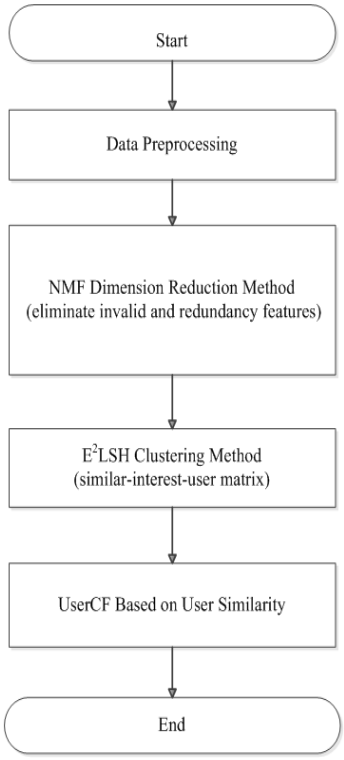

Figure 1. NE-UserCF Framework

UserCF Based on User Similarity: Utilize UserCF to process the similar-interest-user matrix generated by E^{2}LSH Clustering Method. Store it by descending order according to the similar degree of users. Produce the corresponding recommendation candidate sets. Then use the Top-N method to recommend the TOP-10 movies in the recommendation candidate sets to target users. Finally, verify its performance by adopting Precision, Recall, Coverage and Popularity.

4.2. NE-UserCF Algorithms

(1) NMF Dimension Reduction Method Algorithm

Input: X , R , *MAXITER* .(X is the original URM, R is the rank of base matrix, *MAXITER* is the max iteration).

Output: Base matrix named W , coefficient matrix named H .

Step1：Initialize the base matrix W and the coefficient matrix H with negative numbers, and normalization process each column of the base matrix W ;

Step2: Update the line element of coefficient matrix H by using $H(i,j) = \frac{H(i,j) \times (W' \times X)(i,j)}{(W' \times W \times H)(i,j)}$, update the column element of base matrix W by using $W(k,j) = \frac{W(k,j) \times (X \times H')(K,j)}{(W \times H \times H')(k,j)}$ and normalization process the base matrix W ;

Step3: Determine whether have reached the max iteration *MAXITER* .If so, go to Step4; Otherwise, go to Step2;

Step4: End.

(2) E²LSH Clustering Method Algorithm

Input: User-rating matrix produced by NMF, target users matrix (query set file).

Output: Similar-interest-user matrix.

Step1: Create L functions $g_1(\cdot), g_2(\cdot), \cdots, g_L(\cdot)$, and $g_i(\cdot) = (h_1^i(\cdot), h_2^i(\cdot), h_k^i(\cdot))(i = 1,2,\cdots, L), h_1(\cdot), h_2(\cdot), \cdots, h_k(\cdot)$ are hash functions which are selected and produced from the Local Sensitive Hash (LSH) independently and randomly by using $h_{a,b}(v) = \lfloor (a \bullet v + b) / w \rfloor$;

Step2: Utilize LSH function $g_i(v) = (h_1^i(v), h_2^i(v), \cdots, h_k^i(v))$ to process each score vector v in the user-rating matrix and get new score matrix $D_{n \times k} = (u_1, u_2, \cdots, u_k)$;

Step3: Use hash function $h_1(u_1, u_2, \cdots u_k) = ((\sum_{i=1}^{k} r_i' u_i) \bmod prime) \bmod N$ and $h_2(u_1, u_2, \cdots u_k) = (\sum_{i=1}^{k} r_i'' u_i) \bmod prime$ to process the $D_{n \times k} = (u_1, u_2, \cdots, u_k)$ produced in Step2 respectively to get the hash values $h_1(g_i(v))$ and $h_2(g_i(v))$; Store the same data between $h_1(g_i(v))$ and $h_2(g_i(v))$ into the same bucket b_j of hash table L_i;

Step4: Generate the Index key (Value, Index) for each bucket of each hash table;

Step5: Calculate the index key of target user matrix in L hash tables. Then the vector u in L hash tables and unit the search results to present the nearest neighbors S;

Step6: For each vector v in S, calculate the Euclidean distance between v and u, and maintain those vectors that satisfy $(1+\varepsilon)$;

Step7: Return the top m vectors with larger similar degree; Generate the similar-interest-user matrix;

Step8: End.

(3) UserCF Based on User Similarity Algorithm

Input: Similar user matrix, number of similar user K, Recommendation number NItem, Objective Dataset, Test Dataset.

Output: Precision, Recall, Coverage and Popularity.

Step1: To determine whether a target user is a new user; If it is a new user, then go to Step6; Otherwise, calculate the candidate set C. According to the similar users set and their movie watch records, identify the users candidate set $C\{C_{U_1}, C_{U_2}, \cdots, C_{U_n}\}$ having the similar interests with target users set $\{U_1, U_2, \cdots, U_n\}$;

Step2: Unit the users candidate set $C\{C_{U_1}, C_{U_2}, \cdots, C_{U_n}\}$, that is $C = C_{U_1} \cup C_{U_2} \cdots C_{U_n}$, and store the C by using the similar degree;

Step3: Select the first user in C_{u1}, find out its high score movies and recommend to target user U_1 if the target user has no watch record on those movies;

Step4: Repeat Step3 until each element in user candidate set has been process, then produce the final recommended collection $R = R_{C\_Ranked} \cup C_{S-R}$;

Step5: Store the movies in recommendation collection by order descending; Select the TOP-10 movies to recommend to the target users; Go to Step7;

Step6: Recommend the TOP-10 score movies in its similar user set to new users; Go to Step7;

Step7: Utilize the Test Dataset to verify recommendation performance on the Precision, Recall, Coverage and Popularity;

Step8: End.

5. Experiments

5.1. Datasets

In this paper, the authors use MovieLens， which was collected by the GroupLens Research Project at the University of Minnesota. It was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998.

MovieLens dataset contains users' rating on movies, the rating score ranging from 1 to 5. It includes two different sizes of libraries, which is suitable for various algorithms. The large-scale library includes 10,000 ratings data of 3,900 movies produced by 6040 individuals. The small-scale library includes 10,000 ratings data of 1682 movies produced by 943 individuals. In addition, ratings data sets were divided into training set and test set, and have been classified into 5 groups randomly. Detail information with respect to training dataset and test dataset is shown as Table 1.

Table 1. Datasets

Training Dataset	**Numbers of Record**	**Test Dataset**	**Numbers of Record**
u1.base	80000	u1.test	20000
u2.base	80000	u2.test	20000
u3.base	80000	u3.test	20000
u4.base	80000	u4.test	20000
u5.base	80000	u5.test	20000

5.2. Evaluation Metrics

There are various evaluation metrics that can be used to evaluate the performance of recommender systems. Some of them can be quantitatively calculated, while some of them can only be qualitatively described. Some of them can be obtained by off-line calculation, whereas some of them can be gained only through investigations. In general, the authors choose the different types of calculation indexes based on the types of recommender systems (Rating Prediction Recommendation or TOP-N Recommendation). The TOP-N Recommendation is more suitable for movies recommendation. Therefore, in this paper, the authors utilize these evaluation metrics: Precision, Recall, Coverage and Popularity.

Assume that $R(u)$ is the recommendation list, $T(u)$ is the test dataset, I is the items set. Then Precision, Recall, Coverage and Popularity can be defined as below.

Definition1 Precision: the ratio of correct recommendation records in the final recommendation list.

$$\Pr ecision = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |R(u)|} \tag{8}$$

Definition2 Reacall: the ratio of existed user-rating records in the final recommendation list.

$$\mathrm{Re}call = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |T(u)|} \tag{9}$$

Definition3 Coverage: the ability to discover long tail items. It is the ratio of the items being recommended in all items.

$$Coverage = \frac{|\bigcup_{u \in U} R(u)|}{I} \tag{10}$$

Definition4 Popularity: the capability of new items recommendation. A high Popularity means that the recommender systems have recommended a lot of new items to target users.

5.3. Result Analysis

In this paper, the authors adopt the MovieLens dataset, and the authors use the u1.base ~ u5.base to train the NE-UserCF model, u1.test~u5.test to test the performance. The authors utilize off-line experimental methods and recommend TOP-10

movies. The authors take Precision, Recall, Coverage and Popularity as the evaluation metrics to evaluate the performance of NE-UserCF. The authors build up five experiment groups (u1.base-u1.test; u2.base-u2.test; u3.base-u3.test; u4.base-u4.test; u5.base-u5.test). And in each experiment group, the authors change K (number of similar users) from 5 to 400. In order to ensure the accuracy and objectivity of the experimental results, the authors finally calculate the arithmetic mean of the results of five experiment groups. The specific experimental results are shown in Table 2. The authors also conduct the corresponding experiments with the traditional UserCF. Results are shown Table 3.

Table 2. Experimental Results of NE-UserCF

K	Precision	Recall	Coverage	Popularity
5	20.0 %	4.24%	33.758%	5.312
10	25.5%	5.41%	22.061%	5.524
20	27.0%	5.73%	15.212%	5.671
40	28.0%	5.94%	10.182%	5.765
80	31.0%	6.57%	7.394%	5.816
160	31.5%	6.68%	4.848%	5.840
180	30.5%	6.47%	4.909%	5.842
200	31.0%	6.57%	4.848%	5.844
250	31.0%	6.57%	4.485%	5.848
300	31.0%	6.57%	4.424%	5.850
350	29.5%	6.26%	4.121%	5.852
400	32.0%	6.79%	4.303%	5.853

Table 3. Experimental Results of Traditional UserCF

K	Precision	Recall	Coverage	Popularity
5	18.3%	4.6%	58.225%	3.475
10	21.9%	5.8%	50.722%	3.648
20	22.5%	5.9%	36.652%	3.864
40	25.0%	6.1%	25.325%	4.034
80	27.9%	6.8%	17.316%	4.156
160	27.5%	6.9%	13.853%	4.248
180	27.5%	6.9%	13.709%	4.259
200	27.5%	6.9%	13.636%	4.270
250	27.5%	6.9%	13.420%	4.284
300	22.9%	5.8%	13.420%	4.294
350	22.9%	5.8%	13.420%	4.301
400	22.9%	5.8%	13.420%	4.305

As shown in Table 2, we can conclude that NE-UserCF has the best performance on Precision and Recall when K is 160. But from Table 3, traditional UserCF has the best performance on Precision and Recall when K is 80. Therefore, the authors compare their performance respectively when K is 80 and K is 160, as shown in Table 4.

Table 4. Result Comparison

	K	Precision	Recall	Coverage	Popularity
NE-UserCF	80	31.0%	6.57%	7.394%	5.816
UserCF	80	27.9%	5.8%	17.316%	4.156
NE-UserCF	160	31.5%	6.68%	4.848%	5.840
UserCF	160	27.5%	6.9%	17.316%	4.156

By analyzing Table 2, Table 3, and Table 4, the authors obtain the information below:

(1) Precision and Recall: The relationship between Precision, Recall and K is non-linear. When K is 160, both Precision and Recall have the higher values. But, they are not particularly sensitive to K, which means, K can change its value in a reasonable range without affecting Precision and Recall obviously.

(2) Popularity and Coverage：NE-UserCF has a higher popularity than traditional UserCF. As we can see in Table 1, the greater K is, the higher popularity will be and the lower coverage will be. It is because the greater K means the NE-UserCF will complete its recommendation by referring more similar users, therefore its recommendation can be more comprehensive. But, on the other side, the NE-UserCF will have the tendency to recommend the most popular movies but ignore the long tail movies, which leads to a lower coverage.

Consequently, it is rational for us to achieve the conclusion that NE-UserCF has better performance and higher recommendation quality than traditional UserCF, especially on the indexes of Precision and Recall. Moreover, K is an

essential parameter for NE-UserCF, modify K will definitely lead to different results of Precision, Recall, Coverage and Popularity, that means, different recommendation quality.

6. Conclusion

This paper proposes an improved recommender system model named NE-UserCF. Firstly, the authors utilize NMF to process the original user-rating matrix, which eliminates the invalid rating data and redundant rating data, maintains the most effective, valid and useful rating data. Then, the authors adopt E^2LSH to cluster users based on their interests and generate similar-interest-user matrix. Furthermore, the authors use the UserCF Based on User Similarity to further process similar-interest-user matrix produced by E^2LSH and conduct the TOP-10 recommendation. Finally, the authors conduct experiments to verify the performance of NE-UserCF and analyze the experimental results by utilizing evaluation metrics: Precision, Recall, Coverage and Popularity. Consequently, NE-UserCF has a better performance and higher recommendation quality than the traditional UserCF recommender system.

The authors pay little attention to the similarity between items (movies), and time sequence relationships between users and items in this paper. The authors will further focus the research on items' similarity and temporal relationships.

Acknowledgements

This work is implemented when the authors study at Guizhou University. We thank the anonymous reviewers sincerely for their significant and valuable feedback. We further want to extend our sincere gratitude to Ruizhang Huang (Professor) for her insightful comments on this paper. Also, we are grateful to all authors of the references for their great contributions.

References

1. D. Anand and K. K. Bharadwaj, "Utilizing Various Sparsity Measures for Enhancing Accuracy of Collaborative Recommender Systems Based on Local and Global Similarities," *Expert systems with applications*, vol. 38, no. 5, pp. 5101-5109, May 2011
2. J. U. Bin, Y. T. Qian, and Y. M. Chao, "Preference Transfer Model in Collaborative Filtering for Implicit Data," *Frontiers of IT & EE*, vol. 17, no. 6, pp. 489-500, June 2016
3. D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey," *Procedia Computer Science*, vol. 49, pp. 136-146, 2015
4. Y. N. Fang, Y. F. Guo, X. T. Ding, and J. L. Lan, "An Improved Singular Value Decomposition Recommender Algorithm Based on Local Structures," *Journal of Electronics & Information Technology*, vol. 35, no. 6, pp. 1284-1289, June 2013
5. X. X. Geng, L. Y. Ji, and S. U. N. Kang, "Non-negative Matrix Factorization Based Unmixing for Principal Component Transformed Hyperspectral Data," *Frontiers of Information Technology & Electronic Engineering*, vol. 2016, no. 05, pp. 403-412, May 2016
6. Y. Hu, Q. Peng, X. Hu, and R. Yang, "Time Aware and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782-794, May 2015
7. X. Huang, X. Zheng, W. Yuan, F. Wang, and S. Zhu, "Enhanced Clustering of Biomedical Documents Using Ensemble Non-negative Matrix Factorization," *Information Sciences*, vol. 181, no. 11, pp. 2293-2302, November 2011
8. C. Jiang, R. Duan, H. K. Jain, S. Liu, K. Liang, "Hybrid Collaborative Filtering for High-involvement Products: A Solution to Opinion Sparsity and Dynamics," *Decision Support Systems*, vol. 79, pp. 195-208, 2015
9. M. X. Jiao and Y. B. Yang, "Semantic Hashing with Image Subspace Learning," *Journal of Software*, vol. 16, 2014
10. H. M. Li, W. Hao, G. Chen, and X. Liao, "Large-scale Documents Reduction Based on Domain Ontology and E^2LSH," *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*, IEEE, 2014
11. Q. G. Li, J. Q. Shi, Z. G. Qin, and T. W. Liu, "Mining User Behavior Patterns for Event Detection in Email Networks," *Chinese Journal of Computers*, 2014
12. A. Mehmood, T. Damarla, and J. Sabatier, "Separation of Human and Animal Seismic Signatures Using Non-negative Matrix Factorization," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2085-2093, 2012
13. P. Rathod, P. Nguyen, A. Kunjithapatham, M. Sheshagiri, and A. Messer, "Method and System for Extracting Relevant Information from Content Metadata," U.S. Patent No. 8, 115, 869. 14, February 2012
14. X. Sen, Z. M. Lu, and G. C. Gu, "Integrating K-means and Non-negative Matrix Factorization to Ensemble Document Clustering," *Journal of Jilin University(Engineering and Technology Edition)*, vol. 41, no. 4, pp. 1077-1082, April 2011
15. Z. Y. Tian, T. Jung, Y. Wang, F. Zhang, L. Tu, C. Z. Xu, C. Tian, and X. Y. Li, "Real-time Charging Station Recommendation System for Electric-vehicle Taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3098-3109, November 2016
16. G. Uribe, A. Carlos, and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 13, April 2016

17. K. Veningston, and R. Shanmugalakshmi, "Personalized Location Aware Recommendation System," *Advanced Computing and Communication Systems, 2015 International Conference on*, IEEE, 2015
18. M. M. Wang, W. L. Zuo, and Y. Wang, "A Multidimensional Personality Traits Recognition Model Based on Weighted Non-negative Matrix Factorization," *Chinese Journal of Computers*, vol. 39, pp. 1-18, 2016
19. S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, "Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity," *Business Intelligence and Financial Engineering (BIFE), 2012 Fifth International Conference on*, IEEE, 2012
20. D. Y. Wu, "An Electronic Commerce Recommendation Algorithm Joining Case-based Reasoning and Collaborative Filtering," *2015 International Industrial Informatics and Computer Engineering Conference*, Atlantis Press, 2015
21. J. Xu, "A Research of Data Sparsity Problem and Real-time Recommender in Collaborative Filtering," *Doctoral dissertation, Lanzhou University*, 2016
22. F. Yanez and F. Bach, "Primal-dual Algorithms for Non-negative Matrix Factorization with the Kullback-leibler Divergence," *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017
23. P. S. Yuan, S. C. Feng, and W. X. Ling, " C-approximate Nearest Neighbor Query Algorithm Based on Learning for High-dimensional Data," *Journal of Software*, vol. 23, no. 8, pp. 2018-2031, August 2012
24. Z. M. Yuan, C. Huang, X. Y. Sun, X. X. Li, and D. R. Xu, "A Microblog Recommendation Algorithm Based on Social Tagging and a Temporal Interest Evolution Model," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 7, pp. 532-540, July 2015
25. X. B. Zeng, Z. K. Wei, and J. H. Kim, "Research of Matrix Sparsity for Collaborative Filtering," *Journal of computer Applications*, vol. 4, pp. 66, 2010
26. R. J. Zhang, Z. G. Guo, L. I. Bicheng, and H. L. Gao, "A Visual Semantic Concept Detection Algorithm Based on E^{2}LSH-MKL," *Acta Automatica Sinica*, vol. 38, no.10, pp. 1671, 2012
27. S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning From Incomplete Ratings Using Non-negative Matrix Factorization," *Proceedings of the 2006 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, 2006
28. R. Zhang, F. Wei, and B. Li, "E^{2}LSH Based Multiple Kernel Approach for Object Detection," *Neurocomputing*, vol. 124, pp. 105-110, 2014
29. Y. W. Zhao, B. C. Li, and H. L. Gao, "Bag-of-visual-words Based Object Retrieval with E^{2}LSH and Query Expansion," *Instrumentation, Measurement, Circuits and Systems*, pp. 713-725, 2012

Yun Wu received the Ph.D. degree from Guizhou University , Guizhou, China, in 2009. Now he is an associate professor, graduate supervisor, and the member of China Computer Society. His research interests include Distributed Computing, Game Theory, Recommender System, Big Data and its Application.

Yiqiao Li is a master student in the College of Computer Science and Technology, Guizhou University. Her current research interests include Recommender System, Distributed Computing and Data Mining.

Ren Qian is a master student in the College of Computer Science and Technology, Guizhou University. His current research interests include Distributed System, and Recommender System.