Method on Collaborative Filtering Model to Overcome Sparse Data

Zeliang Hao

Nanjing University of Science and Technology, China

Keywords: Collaborative filtering, sparse data, Pear_ After_ SVD, Weighted Similarity.

Abstract: Personalized recommendation system is a very common system at present, which partly solves the problem of information overload. Among them, collaborative filtering technology is the most widely used. However, collaborative filtering technology always has some problems, such as cold start, sparse data and so on. In order to solve these problems, we have to use some method to improve the accuracy of system prediction, which involves the calculation of similarity, fuzzy clustering, user profile, item characteristics and other problems. This paper briefly introduces the relevant background knowledge of collaborative filtering technology, summarizes some similarity algorithms, summarizes and analyzes some methods to overcome data sparsity, and discusses two of them: weighted joint similarity algorithm based on user interest and Pearson_ After _ SVD algorithm. Then, some experimental results of the two methods are compared, and a new method is proposed: the similarity algorithm of the first method is fused to the similarity algorithm of the latter one. In the future work, it is necessary to constantly put forward new methods to give users a better recommendation experience.

1. Introduction

Jeff Bezos, the CEO of Amazon, once said: “if I have 3 million customers on the Web, I should have 3 million stores on the Web.” In today's information explosion society, how to recommend appropriate items according to users' preferences is a very important problem. There are two ways to solve this problem: information retrieval and information filtering. Information retrieval refers to storing information according to certain rules for the convenience of users. We are familiar with the search engines are information retrieval systems; Information filtering system refers to the computer according to the user's personal preferences, automatically select from the information stream users may like the information. There are some differences between information retrieval and information filtering shown in the following table. [1]

Table 1. Difference between information retrieval and information filtering.

	Information retrieval	Information filtering
Information sources	A relatively static and structured data search term	Dynamic unstructured or semi-structured data interest term
Demand representation	Key words	Interest in the template
The target	Select relevant information	Filter out irrelevant information
User characteristics	Large multi-user short-term use	Long-term use of a small range of users

Information filtering technology is widely used, such as information filtering for children and "recommendation" for TikTok and so on. In filtering system, the most popular system is the recommendation system.

The recommendation system's main task is to find the connection between users and items (such as users' preferences for items) in advance, and provide users with correct and personalized feedback in time [2-4]. For example, it can predict the content that users may be interested in and recommend the content to them by analyzing the user's historical information on the Internet, such as purchase

record and browsing history, etc. Among many recommendation systems, collaborative filtering recommendation system [5] is a very common method, whose core is the collaborative filtering algorithm. Collaborative filtering can be divided into model-based collaborative filtering and memory-based collaborative filtering according to different ways of processing information. Sarwar et al. [6] divided memory-based collaborative filtering into user-based collaborative filtering and item-based collaborative filtering based on the relationship between people or things.

Collaborative filtering which is based on memory is a method to obtain the neighbor user set with similar interests and hobbies by statistical method, and then make prediction based on the neighbor user set; model-based collaborative filtering is a method that uses the user's historical records to construct the user's model preferences through statistics or machine learning methods, and then generates recommendations. In memory-based system, user-based collaborative filtering assumes that there is a certain degree of similarity between people's preferences, that is, people who like the same item are likely to like other same items at the same time; project-based collaborative filtering means that there is a certain degree of similarity between items, that is, when customers like a certain item, there is a high probability that they will like other items related to the item in a certain aspect. This paper mainly discusses the memory-based collaborative filtering.

As we can see, the core of memory-based collaborative filtering algorithm is to calculate similarity between users or between items. For example, user-based collaborative filtering algorithm is mainly divided into three steps:

1) Calculate the similarity between users.

2) Determine the neighbor corresponding to the target user according to the similarity between users.

3) The weighted value of the ratings given by neighbors to the target project is used as the rating of the target user to the project.

However, memory-based collaborative filtering systems always have some problems: sparsity, cold start, scalability, accuracy and so on. Some problems can cause the quality of recommendation results to be greatly reduced. This paper mainly studies the sparsity of user-based collaborative filtering system, including the calculation of similarity and the method to overcome sparsity.

The purpose of all the improved methods is to improve the accuracy or overcome some problems such as the sparsity during the process of data. This paper briefly shows some methods of calculating similarity and some method of overcoming sparse data as well as improving the accuracy of predictive values.

2. Relative Theories

2.1 Definitions

Assume that the set of all users is $\mathbf{U} = \{u_p \mid p \in \{1,2,\cdots,M\}\}$, and there are M users which are $u_1, u_2, \cdots, u_m$. Assume that the set of all items is $\mathbf{I} = \{i_q \mid q \in \{1,2,\cdots,N\}\}$, and there are N items which are $i_1, i_2, \cdots, i_N$. Assume that the user's scoring matrix is $\mathbf{R} = \{r_{pq} \mid p \in \{1,2,\cdots,M\}, q \in \{1,2,\cdots,N\}\}$, in which r_{pq} represent the score that user p gives to the item q, if it exists.

2.2 Calculation Method of Different Similarities and Prediction Based on Users

2.2.1 Definition

Assume that a and b represent two random users, if they are not scoped out. $\mathbf{I}(a)$ And $\mathbf{I}(b)$ represent the set of all items scored by user a and user b respectively; $\mathbf{I}(a) \cap \mathbf{I}(b)$ represents the set of all items that scored by both user a and user b. r_{aq} And r_{bq} represent the scores of user a and user b on the item q respectively. In the discussion below, $q \in \mathbf{I}(a) \cap \mathbf{I}(b)$ must be required.

2.2.2 Calculation of Similarity

1) Euclidean Distance

$$d(a,b)=\sqrt{\sum_{\substack{q=1\\ q\in \mathbf{I}(a)\cap \mathbf{I}(b)}}^{n}(r_{aq}-r_{bq})^2} \tag{1}$$

In two and three dimensions, the Euclidean distance is the actual distance between two points. In order to ensure that the results are between 0 and 1, the above calculation results are transformed as follows.

$$sim(a,b)_{euc}=\frac{1}{1+d(a,b)} \tag{2}$$

2) Cosine Similarity and Adjusted Cosine Similarity

This is a very classic similarity algorithm. Among them, the modified cosine similarity algorithm improves the problem that cosine similarity is not sensitive to value.

$$sim(a,b)=\frac{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)} r_{aq}r_{bq}}{\sqrt{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)} r_{aq}^{\ 2}}\sqrt{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)} r_{bq}^{\ 2}}} \tag{3}$$

$$sim(a,b)_{\cos}=\frac{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)}(r_{aq}-\overline{r_a})(r_{bq}-\overline{r_b})}{\sqrt{\sum_{q\in \mathbf{I}(a)}(r_{aq}-\overline{r_a})^2}\sqrt{\sum_{q\in \mathbf{I}(a)}(r_{bq}-\overline{r_b})^2}} \tag{4}$$

3) Pearson Similarity

$$sim(a,b)_{pea}=\frac{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)}(r_{aq}-\overline{r_a})(r_{bq}-\overline{r_b})}{\sqrt{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)}(r_{aq}-\overline{r_a})^2}\sqrt{\sum_{q\in \mathbf{I}(a)\cap \mathbf{I}(b)}(r_{bq}-\overline{r_b})^2}} \tag{5}$$

4) Trajectory Similarity [7]

Trajectory similarity refers to the calculation of user similarity, which only considers the similarity degree of historical decisions between users, without considering the specific score. For example, if user A only scores items 1, 2, and 3, and user B only scores items 4, 5, and 6, then the trajectory similarity of users A and B is 0.

$$sim(a,b)_{tra}=\frac{|\mathbf{I}(a)\cap \mathbf{I}(b)|}{|\mathbf{I}(a)|+|\mathbf{I}(b)|-|\mathbf{I}(a)\cap \mathbf{I}(b)|} \tag{6}$$

$sim(a,b)_{tra}$ Is the trajectory similarity between user a and user b. It is not difficult to know that the value range of trajectory similarity is between 0 and 1, and the closer it approaches to 1, the closer the similarity between two users is.

5) Kendall Similarity [8]

Kendall similarity, which is also known as rank similarity, compares the degree of consistency between users on different items, so as to get the degree of consistency between two users.

$$sim(a,b)_{rank}=\frac{\sum_{q1,q2\in \mathbf{I}(a)\cap \mathbf{I}(b)} f((r_{aq1}-r_{aq2})(r_{bq1}-r_{bq2}))}{\frac{1}{2}\times n\times (n-1)} \tag{7}$$

In the equation, $sim(a,b)_{rank}$ is the rank similarity between user a and user b. q_1 And q_2 are two items that belongs to $\mathbf{I}(a) \cap \mathbf{I}(b)$. r_{aq1} And r_{aq2} are user a's scores to item q_1 and item q_2. r_{bq1} And r_{bq2} are user b's scores to item q_1 and item q_2. n Is the number of all items?

2.2.3 Calculation of Prediction

User-based collaborative filtering algorithm predicts the ratings of the unselected items of the target users based on the ratings of the neighbor users, usually by aggregation.

$$r_{aq}{}^{pre} = \overline{r_a} + \frac{\sum_{b \in \mathbf{N}(a)} sim(a,b) \times (r_{bq} - \overline{r_b})}{\sum_{b \in \mathbf{N}(a)} sim(a,b)} \tag{8}$$

In the equation, user a is the target user, user b belongs to the neighbour set of user a. $\overline{r_a}$ And $\overline{r_b}$ are the average of all known scores for user a and user b respectively. Item q belongs to the collection of items that user a has not scored. $r_{aq}{}^{pre}$ Is the score value predicted by user a for item q. $\mathbf{N}(a)$ Refers to the neighbor's sets of target user a, which can be determined by limiting the range of similarity? For example, we can select only users whose similarity to the target user is 0.2 or above 0.2. Also, $sim(a,b)$ can be replaced by any similarities above, such as $sim(a,b)_{euc}$, $sim(a,b)_{\cos}$, $sim(a,b)_{pea}$, $sim(u,v)_{tra}$ And so on.

2.3 Time Weight Function [9, 10]

Considering that the user's interest changes over time and in order to improve the accuracy of the results, the timeliness of the system is increased. When calculating the similarity, we introduce the Logistic time weight function to re-weight the user's score, so that the proportion of recent score increases.

$$f(t_{pq}) = \frac{1}{1 + e^{-t_{pq}}} \tag{9}$$

$$wr_{pq} = f(t_{pq}) * r_{pq} \tag{10}$$

In formula (7) and (8), t_{pq} is the difference between user p's current scoring time and the earlier scoring time. f Is the time weight function. wr_{pq} Is the new user score weighted by the previous user score?

2.4 Singular Value Decomposition (SVD) [11, 12]

For example, $\mathbf{R}$ is a $m \times n$ matrix? The formula of singular value decomposition is as follows:

$$\mathbf{R} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T \tag{11}$$

Where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, $\mathbf{S}$ is a matrix of $m \times n$ whose non-diagonal elements are all 0. The elements on the diagonal satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0 (m > n)$.

Where, all $\sigma_i (1 \leq i \leq n)$ are greater than 0 and are arranged in order from smallest to largest, called singular values. In general, $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{S}$ must be full rank. If you want to simplify the matrix, keep k maximum singular values of the S-matrix and replace the rest with 0. In this way, we can get new matrix $\mathbf{S}_k$. Similarly, $\mathbf{R}_k = \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k{}^T$ can be obtained in the same way. The matrix $\mathbf{R}_k$ is the closest to $\mathbf{R}$ of all matrices that can generate the initial matrix $\mathbf{R}$ whose rank is equal to k.

3. Methods of Overcoming The Problem of Sparse Data

3.1 Summery of Previous Methods

Many people have proposed ways to overcome data scarcity. Sun Xiaohua [13] summarized these methods and briefly discussed their advantages and disadvantages. In the following table:

Table 2. The traits and shortcomings of some previous methods.

Method	Traits	Weaknesses
Fixed defaults	simple	The values of ungraded items are never exactly the same
Clustering analysis	Transitive association is used to solve sparsity	Need to cluster according to the content of the items
Multi-level association rule	Multi-level association rules mining method	Need to figure out the co-relation rules according to the content of items
ICHM,UCHM calculation	Solving the problem that connections between users or projects cannot be passed	Clustering is required according to content of items or user's profiles
Trust dissemination method, SNACK algorithm	Joining the trust network or social network	Require the user's cooperation
Artificial agents	Reduce the sparse of data	The result is bad
Matrix clustering method	Extracting several submatrices from a sparse matrix	Small coverage rate
Transitive relation	Coverage can be improved	The result is bad

3.2 Weighted Joint Similarity Collaborative Filtering Algorithm Based on Interest [14]

The method combines the similarity of $sim(a,b)_{pea}$, $sim(a,b)_{rank}$ and $sim(a,b)_{tra}$, and it determines the final similarity through weighting. The prediction formula is as follows:

$$sim(a,b)_{weight} = \alpha \times sim(a,b)_{pea} + \beta \times sim(a,b)_{rank} + \gamma \times sim(a,b)_{tra} \tag{12}$$

In the equation, user a is the target user, user b belongs to the neighbour set of user a.

The parameters α, β, γ are the weights, and their value range is all [0, 1], and $\alpha + \beta + \gamma = 1$. Then, the final similarity degree is determined by the limitation of threshold value, the *level*, and the neighbor set $\mathbf{N}$ of each user (for instance, $\mathbf{N}(a)$ of the use a is determined. For example, when $sim(a,b)_{weight} \geq 0.5 (level = 0.5)$ is required, we can only select the users whose similarity with target user is above 0.5 as the element of the neighbor set, and 0.5 can be called similarity threshold(*level*). Finally, the neighbor set is used for prediction. The formula of prediction is the same as that mentioned at sector 2.2.3 in spite of the weight.

$$r_{aq}{}^{pre} = \overline{wr_a} + \frac{\sum_{b \in \mathbf{N}(a)} sim(a,b)_{weight} \times (wr_{bq} - \overline{wr_b})}{\sum_{b \in \mathbf{N}(a)} sim(a,b)_{weight}} \tag{13}$$

In the equation, $\overline{wr_a}$ and $\overline{wr_b}$ refers to the average of weighted scores the user a and user b gave respectively.

3.3 Method of Pear_ After_ SVD [13]

This method combines Pearson algorithm and SVD algorithm, and overcomes two problems: the first one is that SVD algorithm is less sensitive to the degree of data sparsity: the second one is that Pearson algorithm cannot calculate the similarity between users without common scored items. In this method, SVD method is used to fill the user-rating matrix, and a no-defect rating matrix is

obtained. The value of this matrix is then used to calculate Pearson similarity between users, and then the unrated items are predicted. The specific steps are as follows:

1) Normalize the scoring matrix $\mathbf{R}$ and get $\mathbf{R}_{norm}$.

2) Decompose $\mathbf{R}_{norm}$ by singular value decomposition, and obtain $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$.

$$\mathbf{R}_{norm} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T \tag{14}$$

3) $\mathbf{S}_k$ Is obtained by reducing $\mathbf{S}$ to a matrix of dimension k

4) Same method to get $\mathbf{U}_k$ and $\mathbf{V}_k$.

5) Calculate the square root of $\mathbf{S}_k$ to get $\sqrt{\mathbf{S}_k}$.

6) Calculate two correlation matrices $\mathbf{U}_k\sqrt{\mathbf{S}_k}$ and $\sqrt{\mathbf{S}_k}\mathbf{V}_k{}^T$.

7) Calculate the initial predicted score of user a on each item q and obtain a score matrix without missing. Formula is as follows

$$r_{aq} = \overline{r_a} + \mathbf{U}_k \cdot \sqrt{\mathbf{S}_k}'(a) \cdot \sqrt{\mathbf{S}_k}\mathbf{V}_k'(q) \tag{15}$$

8) Use the matrix to find the P similarity between users. Formula is as follows

$$sim(a,b)_{pea} = \frac{\sum_{q \in \mathbf{I}(a) \cap \mathbf{I}(b)} (r_{aq} - \overline{r_a})(r_{bq} - \overline{r_b})}{\sqrt{\sum_{q \in \mathbf{I}(a) \cap \mathbf{I}(b)} (r_{aq} - \overline{r_a})^2} \sqrt{\sum_{q \in \mathbf{I}(a) \cap \mathbf{I}(b)} (r_{bq} - \overline{r_b})^2}} \tag{16}$$

9) Set the similarity threshold, select the neighbor user set, and calculate the final score prediction value. Formula is as follows

$$r_{aq}{}^{pre} = \overline{r_a} + \frac{\sum_{b \in \mathbf{N}(a)} sim(a,b) \times (r_{bq} - \overline{r_b})}{\sum_{b \in \mathbf{N}(a)} sim(a,b)} \tag{17}$$

This method belongs to the feature increment algorithm and is suitable for the recommendation system with strong connection between items.

4. Conclusion

4.1 Source of Data and Criteria of Measurement

The source of data is from Movielens. In general, the mean absolute error (MAE) and the root mean square error (RMSE) are two standards to test the prediction accuracy. Assume that the user a 's original score set is $\mathbf{R}_a = \{r_{a1}, r_{a2}, \cdots r_{aN}\}$, and the predicted user a 's score set is $\mathbf{R}_a{}^{pre} = \{r_{a1}{}^{pre}, r_{a2}{}^{pre}, \cdots r_{aN}{}^{pre}\}$. The formulas of MAE and RMSE are as follows

$$MAE = \frac{1}{N} \sum | r_{ai} - r_{ai}{}^{pre} | (1 \le i \le N) \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N} \sum (r_{ai} - r_{ai}{}^{pre})^2} (1 \le i \le N) \tag{19}$$

4.2 Comparison between Two Methods.

On the one hand, the results of method 3.2 [13] were derived from data from 943 users and 1682 movies. In [13], the best value of α, β, γ and *level* trained with this data is 0.2, 0.4, 0.4, and the best similarity threshold level is 0.42.

The following figures (Curve Sim) shows MAE and RMSE values on the data set

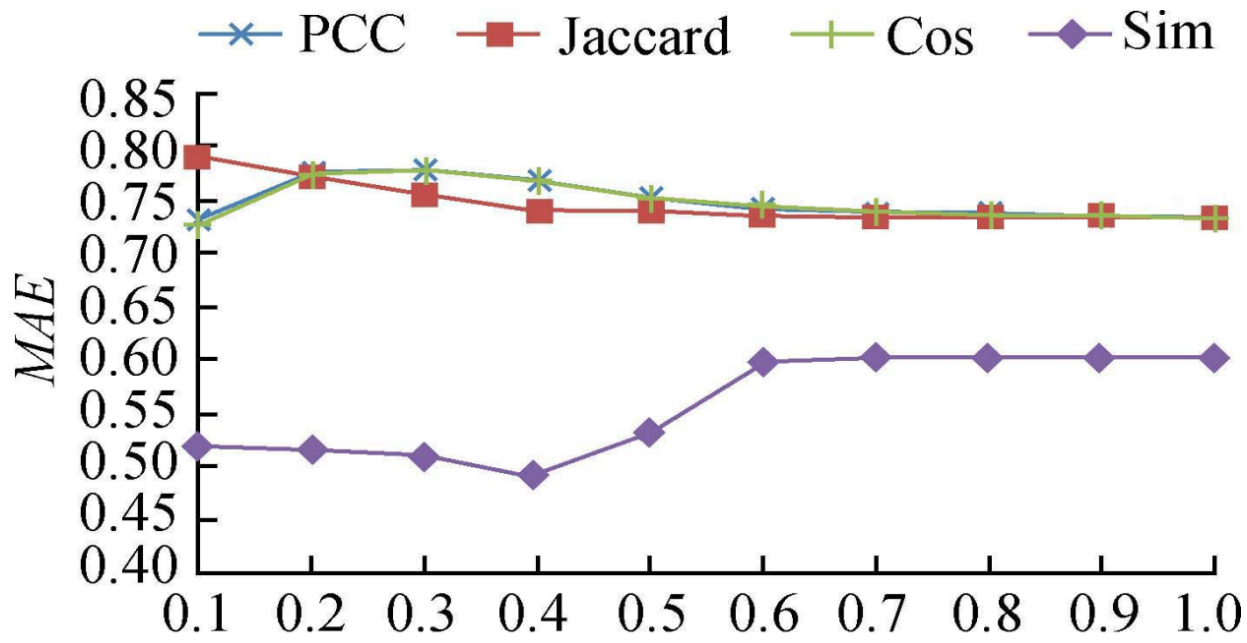

Fig 1. Changes of MAE with similarity threshold in method 3.2.

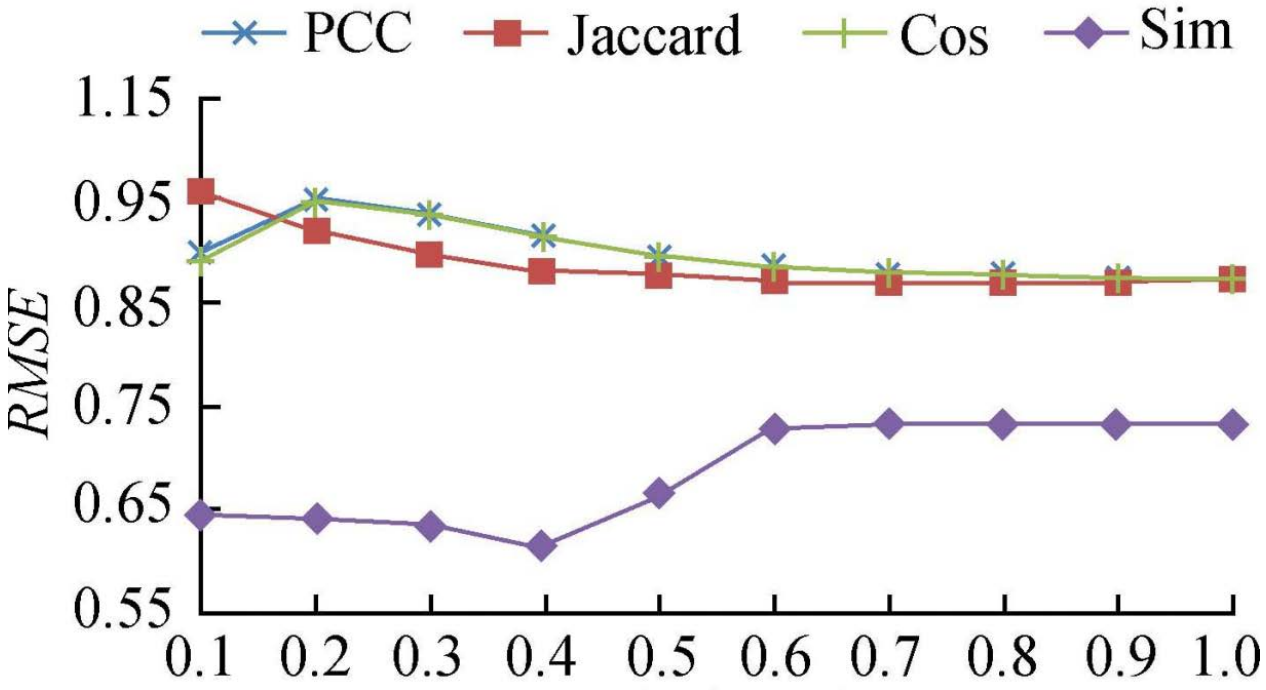

Fig 2. Changes of MAE with similarity threshold in method 3.2.

As can be seen from the figure, MAE varies between 0.45 and 0.65, and when the similarity threshold is 0.42, MAE has the lowest value. The RMSE varies from 0.55 to 0.75, and the RMSE has the lowest value when the similarity threshold was 0.42.

On the other hand, the results of method 3.3 [14] were calculated using data from 1000 users and 1127 movies. The best value of k trained with this data is the integer between 16 and 18.

The following figure (Curve Pear _ after _ SVD) shows MAE values on the data set

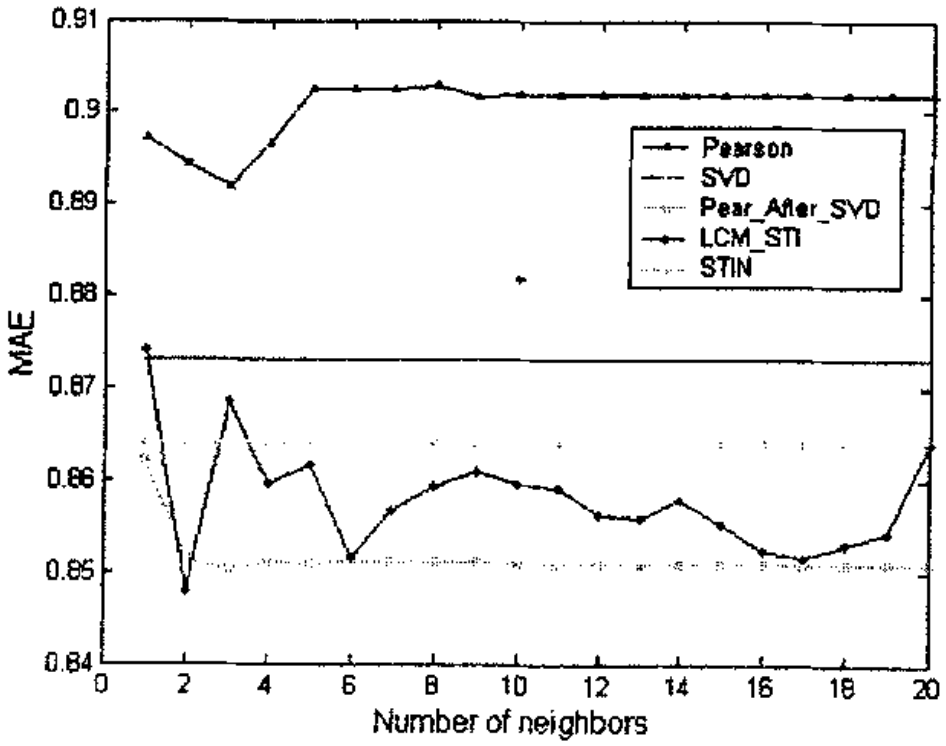

Fig 3. Changes of MAE with number of neighbors similarity threshold in method.

The number of neighbors is determined by the similarity threshold. So, we can use both as independent variables. As can be seen from the figure, MAE varies between 0.85 and 0.87, and when k is between 16 and 18, MAE is the lowest.

Through the comparison of the two methods, we can conclude that in method 3.2, although it is complicated to determine the value of α, β, γ, the result of this method is more accurate because the value of MAE is lower. In method 3.3, although only the value of k needs to be determined, the result is relatively imprecise. Therefore, both methods have advantages and disadvantages.

4.3 Improvement

Based on method 3.2 and 3.3, there is a new method: replace the similarity algorithm of method 3.2 with the similarity algorithm of the eighth step in method 3.3. In other words, step 8 in 3.3 is improved using the similarity algorithm of method 3.2. At the same time, the original data must be time-weighted (Section 2.4). This new method first needs to train the best α, β, γ and k with data. The results obtained by this method should be more accurate in some cases.

4.4 Conclusion

This paper briefly summarizes and discusses the joint similarity algorithm based on time weight and Pear_ After_ SVD methods, and discusses their advantages and disadvantages. Meanwhile, the two methods are fused, and the joint similarity algorithm is fused into Pear _ After_ SVD algorithm. Future work will focus on optimizing new methods to bring more accurate recommendations to users.

References

[1] Xu X L, Que X R, et al. Information Filtering technology and personalized Information Service [J]. Computer Engineering and application Jilt 2003.39 (9): 182-184.

[2] K0NSTAN J A. Introduction to recommender systems: algorithms and evaluation [J]. Acm Sigmod International Conference on Management of Data, 2011, 22 (1):1373-1374.

[3] AD0MAVICIUS G, HUANG Z, TUZHILIN A. Personalization and recommender systems [M]//State-of-the-Art Decision-Making Tools in the Information Intensive Age, 2008.

[4] AD0MAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J] .IEEE Trans. Knowl. Data Eng, 2005, 17(6):734-749.

[5] Syme0nidis P, Nan0p0ul0s A, Papad0p0uls A, et al. Collaborative filtering based on user trends [J]. Advances in Data Analysis, 2006, 44(25):375- 382.

[6] Sarwar,B.M.,Karypis, G, Konstan, J. A., and Riedl, J., Item-based collaborative filtering recommendation algorithm[C], Proceedings of the Tenth International World Wide Web Conference, PP, 285—295, ACM Press, 2001

[7] Seif0uddini H, Djassemi M. The production data-based similarity coefficient versus Jaccard's similarity coefficient [J]. Computers& Industrial Engineering, 1991, 21(1-4):263-266.

[8] ABDI H. The Kendall rank correlation coefficient [J]. Cognition, 2007, 11(1):1-7.

[9] Zhang H, WANG H. Research on collaborative filtering algorithm based on user score and common score item [J]. Journal of Computer Applications, 2018, 36 (1):28-34.

[10] FAN S Z, CHEN H. Personalized dish recommendation based on interest perception and time factor [J]. Journal of Computer Applications, 2018, 2(7):358-361.

[11] Deerwester, S., Dumais, S. T., Furnas, G. W, Landauer, T.K, and Harshman, R., Indexing by Latent Semantic Analysis [J], Journal of the American Society For Information Science, vol. 41(6), pp.391—407, 1990

[12] Golub, G.H. and Van Loan, C.F., Matrix Computations(Third Edition)[M], The Johns Hopkins University Press,1996

[13] SUN X H. Study on sparsity and cold start of collaborative filtering system [academic dissertation] doctor 2005

[14] WANG J F, HAN P F, MIA0 Y L, et al. A collaborative filtering algorithm based on user interest and joint similarity [J].Journal of Henan Polytechnic University (Natural science), 2019, 38(5):118-123. Doi: 10.16186/j. cnki. 1673-9787.2019.5.17