ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

Improved Algorithm of Similarity Measure Based on Matrix Factorization Filling and Filling Confidence

Liu Xiaoyu[1,a]

[1]*School of Information, Engineering Nanjing University of Finance and Economics, Nanjing, China*
[a]*1542679767@qq.com*

Abstract: *Although collaborative filtering recommendation has been very mature in practical application, the problem of data sparsity has been difficult to solve. Sparse data seriously weakens the accuracy of similarity measurement, and also affects the recommendation accuracy of recommendation system. Therefore, this paper makes a deep study on this problem and proposes an improved similarity measure algorithm based on matrix factorization filling and filling confidence. The core idea of the algorithm is to fill the original sparse data set first, and then calculate the similarity based on the filling. In the process of matrix filling, considering that the matrix decomposition model can effectively alleviate the sparsity of data by associating users and items with hidden features, this paper chooses the BiasSVD matrix decomposition model as the basis for filling improvement. Considering that the value of user's interest or behavior will decline with the change of time, the user's recent interaction data can better reflect their current interest than the long-term interaction data, so the improved algorithm integrates the time decline function into the BiasSVD algorithm when filling, and uses the time decline function to give different weights to the user's historical data at different stages, In order to achieve better filling effect. At the same time, after filling, considering that the traditional matrix filling algorithm ignores the credibility difference between the real data and the filled data, and there is no distinction in the next step of similarity calculation, which affects the performance of the recommendation. Therefore, the concept of filling confidence is introduced in the calculation of similarity, which fully distinguishes the reference of the real score and the predicted score.*

Keywords: *collaborative filtering; data sparsity; user similarity measure; data filling*

1. Introduction

With the rapid development of e-commerce, the current situation of data sparsity has become the main obstacle faced by the traditional collaborative filtering algorith. In order to alleviate this problem, most researchers choose the most direct way of data filling. In this paper, the direction of data filling is divided into two categories. One is based on fixed value filling, the other is based on prediction data filling.

Fixed Value Filling. In this way, the default value is usually replaced by the mean, median or mode, and fixed value weighting can also be used to fill the matrix. .For example, Zheng et al. fully considered the internal relationship between user preferences and item attributes by calculating the user preference weight and average score of different item attributes. Kant et al. weighted the average score of users and items and then filled the matrix. This kind of method has the advantages of simplicity and low interference. But this method will greatly reduce the personalization of the scoring matrix and the interaction with users.

Prediction Data Filling. Nasiri et al. used the clustering model to pre fill the non evaluated items in the matrix, so that the scoring data is no longer sparse. Liji et al. improved the effect of recommendation by constructing network clustering model and iteratively grouping filling matrix. Yang et al integrated the trust model into the recommendation algorithm, and fully explored the trust rules among users, so as to preliminarily fill the scoring data. Liu et al. improved the slope one algorithm and improved the accuracy and efficiency of the recommendation algorithm by pre filling part of the unsatisfied items. The above filling algorithms fully tap the value of user rating data, but the predictability of filling value also makes the calculated filling similarity have certain assumptions.

In recent years, matrix factorization technology stands out from many algorithms because of its outstanding filling effect. Liu et al. use SVD to pre fill the user rating data, and then carry out the

ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

recommendation based on mixed similarity. And then the Latent Factor Model model has quickly attracted the attention of the academic community to the matrix factorization algorithms, and then there are many improved matrix factorization filling algorithms based on LFM model, such as the BiasSVD algorithm which adds the bias factor of users and items, and the SVD++ algorithm which adds the influence of user's historical behavior. Zhang et al. put the personalization and difference of user privacy into the probabilistic matrix decomposition model for matrix filling. Guo et al combined trust model with SVD + + decomposition model to form a new improved matrix decomposition technology for matrix filling. Ranjbar et al. also proposed a filling algorithm based on non negative matrix factorization model, which improves the final performance of filling and recommendation.

However, the above filling algorithms based on matrix factorization do not pay attention to the importance of the concept of information expiration. In order to express the trend that user preferences change with time, we can usually use the time decay function to assign corresponding weights to the user's interaction data in different periods. At the same time, the above matrix filling algorithms do not fully distinguish the confidence difference between the real data and the predicted data after filling, which is not conducive to the accuracy of the subsequent similarity measure. Therefore, this paper finally forms an improved algorithm of similarity measure based on matrix factorization filling and filling confidence.

2. Improved Algorithm Of Similarity Measure Based On Matrix Factorization Filling And Filling Confidence

2.1 Biassvd Matrix Filling Algorithm with Time Decay Function

2.1.1 Biassvd Matrix Filling Algorithm

The expression of matrix factorization algorithm is to decompose the rating matrix R into the multiplication result of two potential eigenmatrices U and V. U matrix can be regarded as user's potential characteristic matrix, V matrix can be regarded as items's potential characteristic matrix.For the score matrix R of m * n dimension, the matrix decomposition algorithm can determine the user matrix U of m * k dimension and the item matrix V of k * n dimension by adjusting the number of potential features, which is the value of k.The requirement of matrix decomposition is that the matrix R can be recovered through the following formula:

$$R_{m*n} \approx (U_{m*k}).(V_{k*n}) \tag{1}$$

r_{ui} is the score of user u on item i in rating matrix R ,and it can be approximately expressed by $\widehat{r_{ui}}$, which is the result of corresponding vector dot product in U and V. $\widehat{r_{ui}}$ can also be understood as the score prediction value of user u for item i.And in reality, a real rating data is not only related to the implicit characteristics of users and projects, but also affected by some bias factors of scoring system, users and projects. Therefore, BiasSVD model adds these three factors into the scoring prediction formula and makes improvements:

$$\widehat{r_{ui}} = \mu + b_i + b_u + p_u.{q_i}^T \tag{2}$$

The loss function is used to measure the difference between the predicted value and the real score.In order to minimize the error of the prediction score on the test set, we can usually find the optimal loss function to decompose the appropriate U and V matrix to minimize the error of the training set. The formula of loss function L is:

$$L = \min \sum_{(u,i)\epsilon k} (r_{ui} - \widehat{r_{ui}})^2 + \lambda (\|q_i\|^2 + \|p_u\|^2 + {b_i}^2 + {b_u}^2) \tag{3}$$

After the loss function is established, the next step is to seek the optimization of the loss function with the help of optimization algorithm. Here, the Stochastic Gradient Descent method is selected for optimization.The first step is to calculate the first-order derivative of the loss function. The cost of the first-order derivative is small, which can be better used in large data sets.The fastest descent direction is determined by calculating the first-order partial derivative of the parameters, and then the parameters are continuously updated through repeated iterations.

2.1.2 Improved Algorithm of Biassvd Filling with Time Decay Function

In real life, people's daily memory of information will decline with the change of time. Some scholars put forward the concept of information expiration. In order to describe the declining trend of user's interest over time, the time decay function can be used to assign corresponding weights to the

ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

user's historical information in different stages.In this paper, we choose exponential function to reflect the influence of time factor on the change of user interest, and further highlight the proportion of user preference in the short term. The value of the function is (0,1). The specific formula is as follows:

$$W_t = \frac{1}{1+\exp^{\Delta t}} = \frac{1}{1+\exp^{(t_{u,i}-t_{min})}} \tag{4}$$

Among them, $t_{u,i}$ represents the time when user u scores item i, t_{min} represents the earliest scoring time of user u, which is the benchmark for calculating the time interval.

After incorporating the time decay function W_t into the BiasSVD algorithm, the loss function formula also needs to make the following changes:

$$L = \min \sum_{(u,i)\epsilon k} (r_{ui}.W_t - (\mu + b_i + b_u + p_u.q_i^T))^2 + \lambda (\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2) \tag{5}$$

At the same time, for the convenience of recording, the improved biassvd filling algorithm with time decay function is called TB-SVD algorithm.

2.2 Improved Similarity Algorithm Based On Filling Confidence

The traditional matrix filling algorithm ignores the confidence difference between the real data and the filled prediction score, and does not distinguish between them in the subsequent similarity calculation, which affects the recommendation effect. However, the reference provided by the original score and the filled score for further similarity calculation and score prediction is obviously different, so the real score and the predicted score should be fully distinguished when calculating the similarity. In order to better understand the algorithm, this paper first gives the concepts of score filling confidence and item filling confidence.

Score Filling Confidence: if the credibility of real data is defined as 1, then the credibility of filling data should be less than that of real data. Here, filling confidence coefficient c is used to represent the credibility weight of score, and c is a number between 0 and 1.At the same time, after completing the prediction filling of scoring data, it is necessary to distinguish the real data from the filling data to generate the filling confidence matrix C_{m*n}.The elements in this matrix are composed of 1 and 0. 1 means that the corresponding score of the element is real data, and 0 means that the corresponding score is filled data.

Item Filling Confidence: With the generation of filling behavior, users have scored all items, and the scoring of common evaluation items among users can be divided into three categories:(1) It consists of two real scores（2) It consists of a real score and a filling score（3) It consists of two filling scores. Obviously, whether a project contains real score data and how much real score data it contains are also different for the reference of similarity calculation. In addition, because the two scores are filled with data items, the reference degree of user similarity calculation is not high, this part of data is filtered in the experiment. Therefore, this paper defines that the credibility of a project composed of two real data is 1, and that of a project with one filled data is c.

The similarity algorithm based on filling confidence is based on Pearson correlation coefficient formula:

$$sim(a,b) = \frac{\sum_{i\in I}(r_{a,i}-\overline{r_a})(r_{b,i}-\overline{r_b})}{\sqrt{\sum_{i\in I}(r_{a,i}-\overline{r_a})^2}\sqrt{\sum_{i\in I}(r_{b,i}-\overline{r_b})^2}} \tag{6}$$

sim(a,b) represents the similarity value between users a and b,I represents the common scoring item set of a and b, $r_{a,i}$ represents the scoring of user a on item i, $r_{b,i}$ represents the scoring of user b on item i, $\overline{r_a}$ represents the average score of user a, and $\overline{r_b}$ represents the average score of user b.

The formula of similarity algorithm based on filled confidence is as follows:

$$sim*(a,b) = \frac{\sum_{i\in I}(r_{a,i}-\overline{r_a})(r_{b,i}-\overline{r_b})*c^{(2-(C_{a,i}+C_{b,i}))}}{\sqrt{\sum_{i\in I}(r_{a,i}-\overline{r_a})^2*c^{(2-(C_{a,i}+C_{b,i}))}}\sqrt{\sum_{i\in I}(r_{b,i}-\overline{r_b})^2*c^{(2-(C_{a,i}+C_{b,i}))}}} \tag{7}$$

$$\overline{r_a} = \frac{\sum_{i\in I}(r_{a,i}*C_{a,i})}{\sum_{i\in I}C_{a,i}} \tag{8}$$

$$\overline{r_b} = \frac{\sum_{i\in I}(r_{b,i}*C_{b,i})}{\sum_{i\in I}C_{b,i}} \tag{9}$$

ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

In this paper, sim*(a,b) is recorded as the similarity between user a and user b based on the improved similarity algorithm of filling confidence. I is the common scoring item set of a and b. $r_{a,i}$ represents the scoring of user a on item i, $r_{b,i}$ represents the scoring of user b on item i.$C_{a,i}$ represents the filling confidence coefficient of user a to item i in the filled confidence matrix C_{m*n}. $C_{b,i}$represents the filling confidence coefficient of user b to item i in the filled confidence matrix C_{m*n}.

3. Conclusions

3.1 Data Sets and Evaluation Indicators

3.1.1 Data Sets

In this experiment, two open film scoring data sets were selected: Movielens-100k film scoring data set and Movielens-latest-small film scoring data set, which were provided by the GroupLens Research Institute of the University of Minnesota.

Movielens-100k data contains 100000 scoring data of 943 users and 1682 movies, and the number of movies evaluated by each user is no less than 20, and the scoring data value is between 1 and 5 and is an integer. The sparsity of the data set is about 93.7%.

Movelens latest small data set contains 100837 rating data of 610 users and 9742 movies. Similarly, the number of movies evaluated by each user is no less than 20, and the scoring value adopts five-star system, increasing by half star (0.5-5.0 stars). The sparsity of the data set is about 98.3%

3.1.2 Scoring Prediction

After the end of user similarity calculation, the next step is to complete the prediction for the scoring data in the test set, and select the top N users whose similarity with the target user is from large to small to form a similar neighbor set. Finally, it is necessary to predict the items that have not been evaluated by the target users according to the scoring of similar neighbors. The calculation formula is as follows:

$$\mathrm{pred}(a,i) = \bar{r_a} + \frac{\sum_{b\in N} \mathrm{sim}(a,b)*(r_{b,i}-\bar{r_b})}{\sum_{b\in N} \mathrm{sim}(a,b)} \tag{10}$$

3.2 Evaluating Indicator

In order to verify the performance of the results, two evaluation indexes, Mean Absolute Error and root Mean Square Error, were used in the experiment. The formulas are as follows:

$$\mathrm{MAE} = \frac{\sum_{u,i\in T} |\widehat{r_{u,i}} - r_{u,i}|}{|T|} \tag{11}$$

$$\mathrm{RMSE} = \frac{\sqrt{(\sum_{u,i\in T} |\widehat{r_{u,i}} - r_{u,i}|)^2}}{|T|} \tag{12}$$

Above, $r_{u,i}$ represents the actual score of u for item i, $\widehat{r_{u,i}}$ represents the user u's forecast score for item i, T represents the test set, and $|T|$ is the size of the test set.

3.3 Analysis of Experimental Results

3.3.1 Experimental Results of Improved Algorithm

The model involves the establishment of many constant parameters, including hidden vector dimension k and regularization factor λ、 Learning rate α、and the number of iterations n.According to the literature and the experiment in this paper, it is finally determined that the dimension k of hidden vector is 20 and the regularization factor is 1λ0.05, learning rateα0.05 and the iteration vector n is 40.

At the same time, the model involves the establishment of filling confidence C. the specific establishment process and the final experimental results are shown in Figure 1.

ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

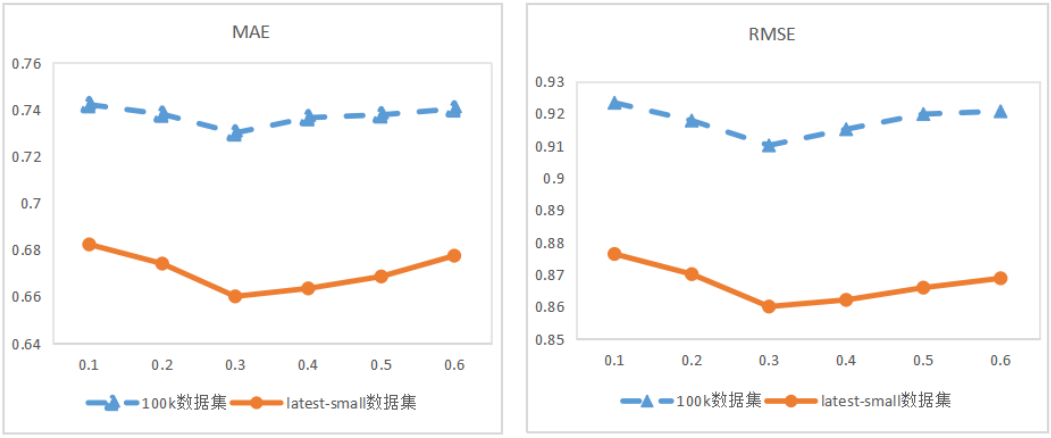

Figure 1 Influence of filling confidence on recommendation results in different data sets

The horizontal axis describes the change of filling confidence coefficient C, the vertical axis describes the effect of the filling algorithm on different data sets, and the vertical axis represents the corresponding MAE or RMSE value. The experimental results show that when the filling confidence is 0.3, the results of the improved algorithm on the two data sets are the best.

3.3.2 Comparison of Experimental Results

In order to test the effect of the improved algorithm, this paper will compare the final MAE and RMSE results of the following algorithms on two sets of data sets.

(1) Improved algorithm of similarity measure based on matrix factorization filling and filling confidence (MFFC) is proposed in this paper;

(2) Collaborative filtering recommendation algorithm based on Pearson correlation coefficient (Per-CF) proposed in literature;

(3) Adjusted Cosine Correlation-Based Collaborative Filtering (ACC-CF) proposed in literature.

(4) Collaborative filtering recommendation algorithm based on singular value decomposition (SVD) proposed in literature;

(5) Personalized Recommendation Based On User Attributes Clustering And Score Matrix Filling(PR-ACMF)proposed in literature

The experimental results of the above algorithms in different data sets are shown in Figure 2.

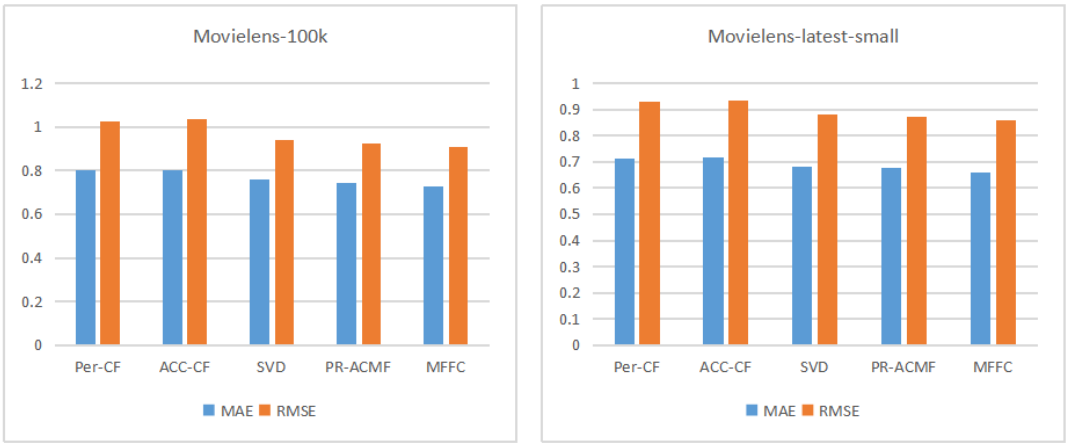

Figure 2 Recommendation effect of different algorithms on different datasets

The experimental results show that the Mae and RMSE values of MFFC proposed in this paper are smaller than those of other algorithms, and the score prediction accuracy is higher. It shows that the algorithm can effectively improve the recommendation results.

ISSN 2616-5775 Vol. 4, Issue 2: 49-54, DOI: 10.25236/AJCIS.2021.040209

References

[1] Surati A.K. and Jaydeep G. (2018)A Survey of Recommendation System. International Conference on Inventive Research in Computing Applications, 398-401.
[2] Bindu K.R., Visweswaran R.L., Sachin P.C., et al. (2017) Reducing the Cold-User and Cold-Item Problem in Recommender System by Reducing the Sparsity of the Sparse Matrix and Addressing the Diversity-Accuracy Problem. International Conference on Communication and Networks,561-570.
[3] Zheng X.N,,Tan Q.H.,Ma H.,et al.(2020)Improved hybrid recommendation algorithm based on filling user preference matrix. Computer engineering and design,41(10):2784-27
[4] Kant S. and Mahara T. (2018) Merging user and item based collaborative filtering to alleviate data sparsity. International Journal of System Assurance Engineering & Management, 9(1):1-7.
[5] Nasiri M., Minaei B., Sharifi Z. (2017)Adjusting data sparsity problem using linear algebra and machine learning algorithm. Applied Soft Computing, 61(01):1153-1159.
[6] Li J. U., Chai Y., Chen J. (2017) Improved personalized recommendation based on user attributes clustering and score matrix filling. Computer Standards & Interfaces, 57(01):59
[7] Yang X.Y.,Yu J.,Tu,Y.et al.(2015)Collaborative filtering recommendation model based on trust model filling. Computer Engineering,41(05):6-13.
[8] Liu L.J., Lou W.G., Feng G.Z. (2016) Weighted Slope One algorithm based on user similarity. Computer application research, 33(009):2708-2711.
[9] Liu Q.Q., Luo Y.L., Wang Y.F., et al. (2019)Hybrid Recommendation Algorithm Based on SVD Filling. Computer Science.
[10] Xiang L. (2012) Recommended system practice. People's Posts and Telecommunications Press, 186-192.
[11] Koren Y. (2010) Factor in the Neighbors: Scalable and Accurate Collaborative Filtering.ACM Transactions on Knowledge Discovery from Data (TKDD),4(1), 1-24.
[12] Zhang S., Liu L., Chen Z., et al. (2019) Probabilistic Matrix Factorization with Personalized Differential Privacy. Knowledge-Based Systems, 183(01): 104864.
[13] Guo G., Zhang J., Yorke-Smith N. (2015)TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. AAAI Conference on Artificial Intelligence, 29.
[14] Ranjbar M., Moradi P., Azami M., et al.(2015) An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. Engineering Applications of Artificial Intelligence, 46(NOV.PT.A):58-66.
[15] Pan T.T.,Wen F.,Liu Q.R.(2017)Collaborative filtering algorithm based on matrix filling and item predictability. Journal of automation,43(09):1597-1606.
[16] Zhang W.B. (2019) Research and implementation of matrix factorization recommendation algorithm based on multi factors.Beijing University of Posts and Telecommunications.
[17] https://grouplens.org/datasets/movielens/
[18] Bell R.M. and Koren Y.(2007) Lessons from the Netflix prize challenge. ACM SIGKDD Explorations Newsletter, 9(2):75-79.
[19] Javari A., Gharibshah J., Jalili M. (2014) Recommender systems based on collaborative filtering and resource allocation. Social Network Analysis & Mining, 4(1):234.
[20] Ahn H.J. (2008)A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences, 178(1):37-51.