Journal of Electronic Research and Application, 2025, Volume 9, Issue 3
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Research on Recommendation Algorithm Based on TrustSVD++ and XGBoost

Gongjin Chen*, Yantao He, Yuting Kong

Department of Computing, Guangdong University of Science and Technology, Dongguan, Guangdong, China

**Author to whom correspondence should be addressed.*

***A*bstract:** Aiming at the problem that the data in the user rating matrix is missing and the importance of implicit trust between users is ignored when using the TrustSVD model to fill it, this paper proposes a recommendation algorithm based on TrustSVD++ and XGBoost. Firstly, the explicit trust and implicit trust were introduced into the SVD++ model to construct the TrustSVD++ model. Secondly, considering that there is much data in the interaction matrix after filling, which may lead to a rather complex calculation process, the K-means algorithm is introduced to cluster and extract user and item features at the same time. Then, in order to improve the accuracy of rating prediction for target users, an XGBoost model is proposed to train user and item features, and finally, it is verified on the data sets MovieLens-1M and MovieLens-100k. Experiments show that compared with the SVD++ model and the recommendation algorithm without XGBoost model training, the proposed algorithm has the RMSE value reduced by 2.9% and the MAE value reduced by 3%.

Keywords: TrustSVD++; XGBoost; Feature extraction; Clustering; Implicit trust information

Online publication: 29 May, 2025

1. Introduction

In the information age, the accumulation of data leads to information overload, which makes it a time-consuming and laborious process for users to filter out the information they need from massive data. In order to solve this problem, researchers have introduced recommendation algorithms that can recommend different items according to different users, which alleviates the above problem to some extent. However, the algorithm is not suitable for new users. Due to the lack of interactive information, it is impossible to analyze user preferences and further lead to the cold start problem.

To solve the above problems, Chen and Shu [1] combined the trust network and SVS++ algorithm to make recommendations, and combined the explicit trust factor with the user's implicit feedback to fill the missing values in the above matrix. However, they only started from the perspective of explicit trust information and ignored the implicit trust factor. Zhang *et al.* [2] proposed the application of the XGBoost algorithm in e-commerce commodity

recommendation, and used the XGBoost model to train the characteristics of users and items. However, due to the lack of consideration of the user's historical interaction information in the proposed model, the characteristics of the training are biased, which leads to poor recommendation performance. He and Hu [3] proposed to integrate matrix factorization and the XGBoost algorithm for recommendation, which solved the problem that the founder must be used in matrix factorization. However, they only introduced implicit feedback information on the basis of singular value decomposition, and ignored the trust relationship between users that would also have an impact on the recommendation effect. Yin *et al.* [4] adopted singular value decomposition dimension reduction technology of local optimization and K-means clustering technology for recommendation, and used a clustering algorithm to solve the problem of excessive data volume of completion scoring matrix, but ignored the shortcomings of K-means algorithm.

This paper makes an in-depth analysis of the above problems and makes the following improvements: Firstly, explicit trust and implicit trust were integrated into the SVD++ model, and then the way of selecting cluster centers was redefined to make up for the shortcomings of selecting cluster centers in the K-means algorithm. Secondly, the improved clustering algorithm was used to cluster the filled rating matrix and construct the features corresponding to users and items. Finally, the XGBoost model is used for training to obtain the predicted score value of the project by the target user [5]. In summary, the method mentioned in this paper not only considers the influence of the implicit trust factor between users but also improves the shortcomings of the clustering algorithm.

2. Related technologies

2.1. Singular Value Decomposition (SVD)

Traditional matrix factorization has strict requirements on the dimension of the matrix [6], that is, it must be a square matrix, but most data sets do not meet this condition. In order to overcome this limitation, Singular Value Decomposition (SVD) technology has been proposed by scholars. It is defined as follows.

$$\mathrm{R} = U^T S V \tag{1}$$

Where U and V are users and item feature matrix respectively, and S is used to represent the relationship between them

2.2. Introduction to SVD++

The idea of SVD++ model is that the Bias SVD model considers the bias of users and the bias of items [7]. When users lack interactive information about a certain item, they can use its implicit feedback, such as historical records or click-through rates, and other information to predict their prediction score for a certain item. It is defined by the formula:

$$\hat{r}_{ui} = \mu + \mathrm{b_u} + \mathrm{b_i} + \mathrm{q_j^T}\left(\mathrm{p_u} + |\mathrm{I_u}|^{-1/2}\sum_{\mathrm{j}\in \mathrm{I_u}} \mathrm{y_j}\right) \tag{2}$$

Where Iu is the user's historical interaction information, namely implicit feedback information. At the same time, experiments show that taking 1/2 can alleviate the influence of different set numbers on the prediction score.

2.3. K-means algorithm

The main goal of K-means algorithm is to cluster the given dataset into k clusters according to the distance formula under the condition of determining the initial K value [8]. For each cluster, it is hoped that the data difference within it is as small as possible, and the data difference between clusters is as large as possible. Its commonly used

distance formula is the Euclidean formula, which is defined as follows：

$$dist(x_i, x_j) = \sqrt{\sum_{d=1}^{D}(x_{i,d} - x_{j,d})} \tag{3}$$

The clustering center is updated as follows:

$$Center_k = \frac{1}{|C_k|}\sum_{x_i \in C_K} \mathrm{dist}(\mathrm{x_i},\ \mathrm{Center_k}) \tag{4}$$

In the formula, C_K denotes the kth cluster, and represents the number of points in that cluster.

The termination condition of K-means algorithm is controlled by two factors, that is, the number of iterations and the error value. When the number of iterations is reached or the sum of squared errors of all objects to the corresponding initial value center is minimum, the iteration is terminated.

$$E = \sum_{i=1}^{k}\sum_{j=1}^{n_j} d(x_j, c_i) \tag{5}$$

2.4. XGBoost model

XGBoost [3,9,10] is an improvement of the Boosting algorithm proposed by Dr. Tianqi Chen of the University of Washington in 2014 on the basis of the GBDT algorithm. The internal decision tree uses a regression tree, which has a series of characteristics such as fast speed, good effect, ability to deal with large-scale data, and a custom loss function. Since the XGBoost algorithm can improve the ability of a prediction model, it can be used to improve the prediction score of target users in personalized recommendation. Its objective function is as follows.

$$obj^t = \sum_{n=1}^{N} l\left(y_i, \hat{y}_i^t + f_t(x_i)\right) + \Omega(f_t) \tag{6}$$

3. Algorithm improvement

3.1. TrustSVD model

TrustSVD model integrates explicit trust information between users on the basis of SVD++ model. Suppose that user u trusts user v, and its trust matrix is defined as $T = [t_{u,v}]_{MxM}$ [11–13]. Therefore, the matrix T is decomposed into the trust matrix P_{Mk} and the trusted matrix WMK, such that T≈PWT. Therefore, the trust relationship between user u and user v can be expressed as follows.

$$\hat{t}_{u,v} = \boldsymbol{p}_u \boldsymbol{w}_v^T \tag{7}$$

For matrices P and W, we adopt the method of gradient descent in machine learning to calculate them, and define the loss function as follows:

$$L_t = \sum_u \sum_{v \in T_u} \left(t_{u,v} - \boldsymbol{p}_u \boldsymbol{w}_v^T\right)^2 \tag{8}$$

After introducing the trust factor, the formula becomes as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + \left(\boldsymbol{p}_u + |I_u|^{-\frac{1}{2}}\sum_{i \in I_u}\boldsymbol{y}_i + |T_u|^{-\frac{1}{2}}\sum_{V \in T_u}\boldsymbol{w}_v\right)\boldsymbol{q}_j^T \tag{9}$$

Where T_u is the trust set of user u, which contains all explicit trust information of user u.

3.2. TrustSVD++ model

Because the explicit trust information is generally decided by the user himself, the user considers whether to give

the trust relationship to his trust relationship item according to his own judgment, and its reliability is relatively high. The implicit trust information is obtained through the interaction information between users and commodities and between users [14]. Thus, implicit trust is much easier to obtain than explicit trust. Therefore, it is necessary to introduce an implicit trust factor into the TrustSVD model. The above equation (9) is improved as follows.

$$r_{u,j}^{*} = \mu + b_u + b_j + \mathbf{q}_j^{\mathrm{T}}\left(\mathbf{p}_u + \left|I_u\right|^{-\frac{1}{2}}\sum_{i\in I_u}\mathbf{y}_i + k\left|T_u^{\mathrm{E}}\right|^{-\frac{1}{2}}\sum_{v\in T_u^{\mathrm{E}}}\mathbf{w}_v + (1-k)\left|T_u^{I}\right|^{-\frac{1}{2}}\sum_{v\in T_u^{\mathrm{I}}}\mathbf{m}_v\right) \tag{10}$$

Where and represent the user's explicit trust information set and implicit trust information set respectively, and K is the weight value. When K is 0, it means that only the influence caused by implicit trust is considered; on the other hand, k = 1 means that only the influence caused by explicit information is considered. Several experiments were conducted, and 0.8 was found to be the best value for k.

In order to avoid overfitting in the process of model training, the loss function definition of the TrustSVD++ model is defined by referring to the loss function definition of the TrustSVD model, that is, considering that new users and new items may have a large impact, so a large penalty factor is selected, otherwise a small penalty factor is adopted. The formula is as follows:

$$\begin{aligned} L = {} & \frac{1}{2}\sum_u\sum_{j\in I_u}\left(r_{u,j}^{*} - r_{u,j}\right)^2 + \frac{\lambda_t^{\mathrm{E}}}{2}\sum_u\sum_{v\in T_u^{\mathrm{E}}}\left(t_{u,v}^{\mathrm{E}^*} - t_{u,v}^{\mathrm{E}}\right)^2 + \frac{\lambda_t^{\mathrm{I}}}{2}\sum_u\sum_{v\in T_u^{\mathrm{I}}}\left(t_{u,v}^{\mathrm{I}^*} - t_{u,v}^{\mathrm{I}}\right)^2 + \frac{\lambda}{2}\sum_u\left|I_u\right|^{-\frac{1}{2}}b_u^2 + \frac{\lambda}{2}\sum_j\left|U_j\right|^{-\frac{1}{2}}b_j^2 \\ & + \sum_u\left(\frac{\lambda}{2}\left|I_u\right|^{-\frac{1}{2}} + \frac{\lambda_t^{\mathrm{E}}}{2}\left|T_u^{\mathrm{E}}\right|^{-\frac{1}{2}} + \frac{\lambda_t^{\mathrm{I}}}{2}\left|T_u^{\mathrm{I}}\right|^{-\frac{1}{2}}\right)\|\mathbf{p}_u\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\sum_j\left|U_j\right|^{-\frac{1}{2}}\left\|\mathbf{q}_j\right\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\sum_i\left|U_i\right|^{-\frac{1}{2}}\mathbf{y}_i\ {}_{\mathrm{F}}^{2} + \frac{\lambda}{2}\left|T_v^{\mathrm{E}+}\right|^{-\frac{1}{2}}\left\|\mathbf{w}_v\right\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\left|T_v^{\mathrm{I}+}\right|^{-\frac{1}{2}}\left\|\mathbf{m}_v\right\|_{\mathrm{F}}^2 \end{aligned} \tag{11}$$

Where and represent the implicit trust set and explicit trust set of user v, respectively.

3.3. Optimizing the initial k-means clustering centers

In this paper, aiming at the defects of K-means clustering algorithm in selecting the initial cluster center, the algorithm based on point density is adopted to optimize it. In addition, considering that isolated points affect the cluster center to a certain extent, they are treated separately.

3.3.1. Optimizing the initial k-means clustering centers

Point density: Suppose that a data set U is given and a sample point x is selected from it. The number of all sample points contained in the spherical domain with x as the center of the sphere and r(r>0) as the radius is called the density of point x, which is denoted as D(x), namely:

$$D(x) = |\{p|\ d(x,p) \le r, p \in U\}| \tag{12}$$

Where d(x,p) is the distance between two points, and the similarity is taken as the distance between two points, that is, d(x,p) = sim(x,p). Formula (12) can be modified as follows.

$$D(x) = |\{p|\ \mathtt{sim}(x,p) \ge k, p \in U\}| \tag{13}$$

Where k is the similarity threshold

Outlier: If AV(x) is the mean of all the data points in the dataset, a given sample point x is an outlier if D(x) < mAV(x), where m is in the range [0,1]. In addition, a point is an outlier if the number of surrounding points is very small. Among them:

3.3.2. Optimal selection of initial cluster centers

The basic idea is to calculate the density of each point in the data set, then sort it, and take the point before the average density is greater than or equal to, and add it to set A. The first cluster center is the point with the highest value in A. According to Formula (13), for a certain local point, if its density is the largest, the neighboring points may also have the largest density due to their similarity. Therefore, a threshold d is set to avoid this situation and facilitate the selection of the next clustering center. When selecting the next point, if the similarity between two points is lower than this threshold, it is used as the second cluster center point, and so on until K points are selected. The detailed process is as follows:

(1) Input content: Given data U, the number K of cluster centers to be selected.

(2) Define the set M, set A and set N, and set them as empty sets.

(3) Calculate the similarity between any two points in the data set U, and construct the similarity matrix D.

(4) Use formula (13) to calculate the point density of each point. If D(xi)>A(x) is satisfied, the point will be added to the set A; otherwise, the point will be added to the set N.

(5) Take the largest point in set A as the first initial cluster center, add it to set M, assume it is a1, and mark it as processed.

(6) Then search for the next cluster center. If ①sim (a1,a2) <d②a2 is the point with the largest density remaining in the set, that is, D(a2)=Max{D(x)|x∈U-M}, add it to the set M.

(7) Repeat (5), and the algorithm ends when K cluster center points are found.

According to the above steps, the selection of cluster centers is no longer random, and the selected range is in the set A, which undoubtedly narrows the search range and reduces the required time.

3.3.3. Handling isolated values

The K cluster centers are obtained by the above algorithm, and the data in the set A is clustered by using the K cluster centers, the number of clusters is K, and then the points in the set N are divided into the corresponding categories according to the nearest principle.

4. Experimental verification and analysis

4.1. Experimental data set and evaluation index selection

In this paper, the public movie data MovieLens-1M and MovieLens-100k are used, and the scoring metrics adopted are MAE and RMSE, which are calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in T} \left(r_{ui} - r'_{ui}\right)^2}{|T|}} \tag{14}$$

$$MAE = \frac{\sum_{(u,i)\in T} \left|r_{ui} - r'_{ui}\right|}{T} \tag{15}$$

4.2. Analysis of experimental results

The experimental ideas of this paper are as follows:

(1) The TrustSVD++ model proposed in this paper is compared with Basic User (UCF), item-based (ICF), SVD, and SVD++ models to verify the influence of trust information on the model;

(2) Compare the influence of using matrix factorization on the XGBoost model;

(3) Compare the effects of different data sets on the performance of the model proposed in this paper.

Firstly, the first set of experiments was verified. On the dataset of MoviesLens-100k, RMSE and MSE index values were used. By comparing the TrustSVD++ model proposed in this paper with the User-based Collaborative filtering algorithm (UCF), item-based collaborative filtering (IFC), recommendation algorithm based on SVD model, and collaborative filtering algorithm based on SVD++ model, the specific effects are shown in **Figure 1**.

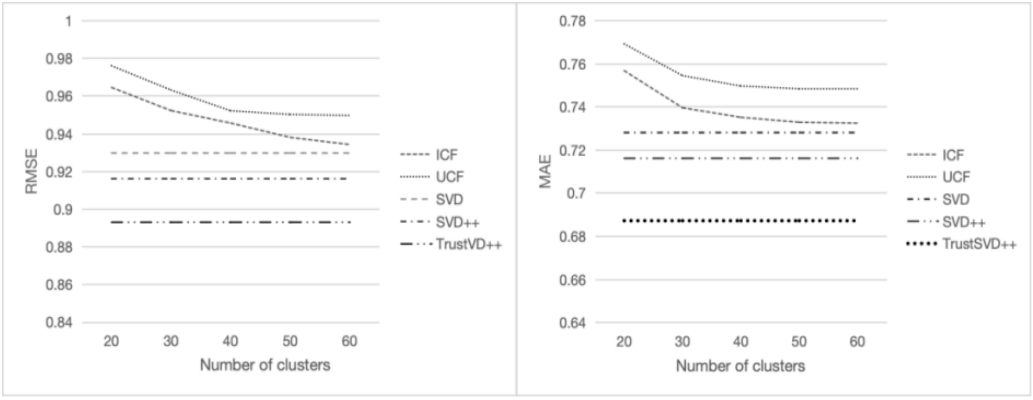

Figure 1. Comparison of RSME and MAE among different algorithms

Considering that the number of clusters need not be specified when SVD++ model is used, the TrustSVD++ model is evolved on the basis of SVD++ model, so the number of clusters need not be specified. According to the results of **Figure 1**, the following points can be obtained: (1) The RMSE values corresponding to ICF and UCF algorithms are basically around 9.33 and 9.5, and the MAE values corresponding to ICF and UCF algorithms are basically stable at around 0.73 and 0.75. (2) Compared with the SVD++ model, the RMSE and MAE of the TrustSVD++ model are reduced by 2.31% and 2.89% respectively, indicating that the introduction of implicit trust information improves the effect of the model. (3) It can be seen that the model used in this paper is slightly better than other models.

Secondly, the second group of experiments is used to verify the influence of the XGBoost model on rating prediction. On the dataset MoviesLens-100k, the method using matrix factorization and filling technology combined with the XGBoost model (TXGB) is compared with other methods only using the XGBoost model (XGB). Its specific effect is shown in **Figure 2**.

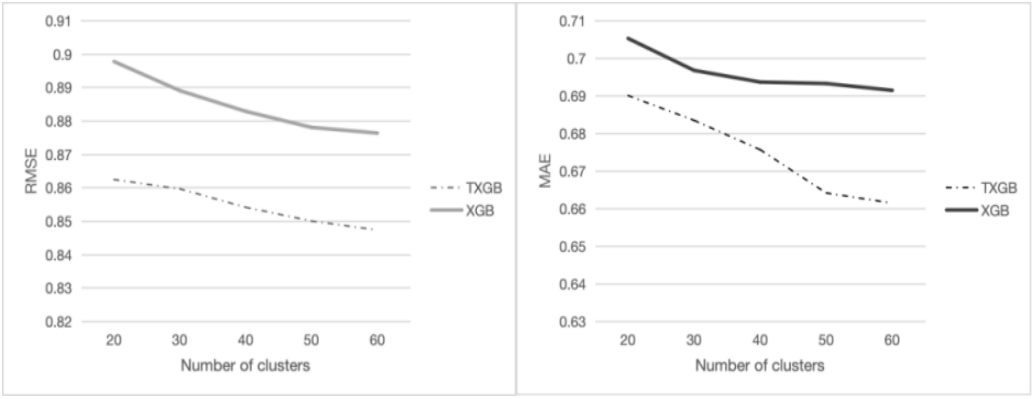

Figure 2. Comparison of RSME and MAE between TXGB and XGB models

Based on the content shown in **Figure 2**, the following conclusions can be drawn: The RMSE and MAE of the XGB model are stable at 0.8764 and 0.6915, while the two indicators of the TXGB model are stable at 0.8474 and 0.6615. It can be seen that filling the interaction matrix with the TrustSVD++ model can improve the performance of the model. When K = 60, RMSE decreases by 2.9% and MAE decreases by 3%.

Finally, the third set of experiments was verified. Considering that the data size of the adopted MoviesLens-100k dataset is small and there may be a chance, in order to avoid this uprising, the above experiments were carried out on the dataset MoviesLens-1M. The result is shown in **Figure 3**.

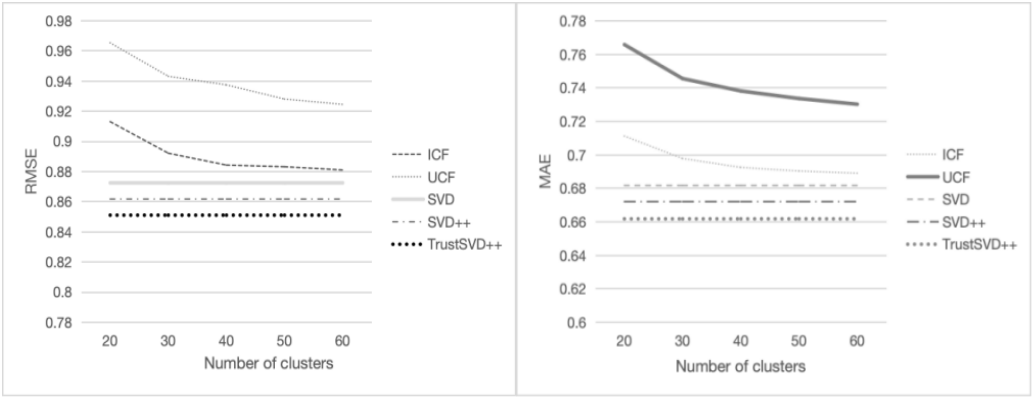

Figure 3. Comparison of RSME and MAE among different algorithms

According to **Figure 3** above, it can be concluded that: (1) On large datasets, this algorithm outperforms other algorithms. Compared with the SVD++ model, the RMSE value of the TrustSVD++ model has decreased by 1.07%, and the MAE value has decreased by 1.03%. (2) This method can still achieve good results on large datasets.

5. Conclusion

This paper proposed a recommendation algorithm based on TrustSVD++ and XGBoost. Firstly, by introducing users' implicit trust information on the basis of the TrustSVD model, the explicit trust and implicit trust between users are combined to make the similarity between users more accurate. Secondly, by redefining the method of cluster center selection in the K-means algorithm, the selection range of cluster centers is reduced to a set. It reduces the time of global selection of clustering centers and improves the clustering effect. Finally, the XGBoost model will be used to train the clustered features to predict the ratings of the target user. Experiments show that the proposed method improves the recommendation effect and alleviates the matrix sparsity problem to a certain extent.

Funding

(1) Guangdong Science and Technology University Young Projects (GKY-2023KYQNK-1 and GKY-2023KYQNK-10)

(2) Guangdong Provincial Key Discipline Research Capacity Improvement Project (2022ZDJS147)

Disclosure statement

The authors declare no conflict of interest.

References

[1] Chen P, Shu F, 2021, Trust Network Recommendation Algorithm Based on SVD++ Latent Factor Model. Big Data, 7(4): 105–116.

[2] Zhang H, Ji H, Zhang H, 2017, Application of XGBoost Algorithm in E-commerce Commodity Recommendation. Internet of Things Technology, 7(2): 102–104.

[3] He J, Hu J, 2021, Personalized Recommendation Algorithm Based on Matrix Factorization and XGBoost. Journal of Chongqing University, 44(1): 78–87.

[4] Yin F, Song Y, Li A, 2019, Collaborative Filtering Algorithm Based on Local Optimization Singular Value Decomposition and K-means Clustering. Journal of Nanjing University of Science and Technology (Natural Science Edition), 43(6): 720–726.

[5] Wang C, Li L, Liu X, 2018, Collaborative Filtering Algorithm Based on Explicit and Implicit Information. Information Systems Engineering, 2018(2): 147–148.

[6] Zhang D, Zhang W, Zhang L, et al., 2021, Research on Singular Value Decomposition Recommendation Algorithm Based on Comprehensive Trust. High-tech Communication, 31(1): 102–112.

[7] Du J, 2020, Deep Matrix Factorization Model Combined with User Reviews, dissertation, Huazhong University of Science and Technology.

[8] Qiu Y, Wang X, Bian S, et al., 2020, Load Balancing Optimization of Multi-Frequency Multi-Mode Network Based on Clustering Analysis and Deep Learning. Telecommunication Science, 36(7): 156–162.

[9] Cui Y, Qi W, Pang H, et al., 2020, Recommendation Algorithm Based on Collaborative Filtering and XG Boost. Application Research of Computers, 37(1): 62–65.

[10] Liu X, 2018, Research on Internet Advertising Click-Through Rate Prediction Method Based on Deep Learning, dissertation, Xi'an University of Technology.

[11] Li Z, 2021, News Recommendation Method Based on Improved Similarity and User Clustering, dissertation, Tianjin University of Commerce.

[12] Song Y, 2021, Research on Group Recommendation System Integrating Social Trust Relationship, dissertation, Yantai University.

[13] Liu Y, Cai X, Shen X, et al., 2019, Application of Collaborative Filtering Algorithm in Recommendation System. Information & Computer, 31(21): 38–40.

[14] Li G, Li L, Zhang J, 2021, Social Collaborative Ranking Recommendation Algorithm Based on Explicit and Implicit Feedback. Journal of Computer Applications, 41(12): 3515–3520.