ATLANTIS PRESS

*Advances in Engineering Research, volume 166*

3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)

# Recommended Algorithm of Latent Factor Model Fused with User Clustering

Junwei Ge[1, a)] , Chun Yang[2, b)] and Yiqiu Fang[1,c)]

[1]*School of Computer science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065,China.*
[2]*School of Software Engineering, Chongqing University of Posts and Telecommunications,Chongqing 400065,China.*

[a)]gejw@cqupt.edu.cn
[b)]986732056@qq.com
[c)]994607515@qq.com

**Abstract.** To solve the problems of partial implicit feature information loss and long-time model training caused by matrix factorization on recommended algorithm of latent factor model (LFM), a recommended algorithm of user clustering fused with latent factor model is put forward. Firstly, the users' preference information is used to cluster them, and then the similarity calculation method is used to find the cluster and the nearest neighbor users that are most similar to the target user. Next, training similar clusters with the improved LFM to obtain the user's implicit features Matrix p and item's implicit feature Matrix q, and then generating the predictive score matrix of similar clusters. Finally, the predictive score of similar clusters are weighted and summed to gain the final user score. Compared with the traditional collaborative filtering and LFM, the improved model effectively reduces the training time and the root-mean-square error of predictive score, also improves the accuracy of predictive recommendation based on the experiments on Movielens datasets.

**Keywords**: Personalized Recommendation; Latent Factor Model (LFM); User Clustering; Collaborative Filtering

## INTRODUCTION

In recent years, with the rapid development of Internet technology, the amount of information on the Internet is increasing, which caused users can not choose satisfactory resources in a short time. How to make users choose their resources quickly and effectively becomes an urgent problem. Therefore, the recommendation technology came into being. Currently, the mainstream recommendation technologies including collaborative filtering recommendation, content-based recommendation, knowledge-based recommendation, hybrid recommendation method, rules-based recommendation and so on. Among them, collaborative filtering algorithm is the most widely used. Generally speaking, collaborative filtering algorithms are mainly divided into user-based [1], item-based [2] and model-based collaborative filtering [3, 4].

The collaborative filtering algorithms based on users and items are mainly recommended by the similarity between users or items. According to the most similar N users or items to predict the target, a number of items with the highest score are recommended to the users. However, because of the sparse and high dimension of the user rating matrix, the quality of the recommendation has been not ideal. At present, many researchers have proposed improved methods for the above problems. Using matrix decomposition [5, 6] can solve these problems well, among which the most famous is the latent factor model.

Latent factor model [7] is the most successful collaborative filtering algorithm based on the model today, which decomposes the high-dimension user rating matrix into two low-dimension users and items implicit feature matrix by the use of matrix decomposition technique, then multiplying the decomposed matrix to get the predictive score by users.

Aiming at the problem of user information loss and long-time model training caused by matrix decomposition in the latent factor model, this paper proposes a recommended algorithm of user clustering fused with latent factor model. This algorithm is, reducing the dimension of users and items through clustering technology [8], and then use similar user score to correct the predictive score in the latent factor model. This method effectively alleviates the model training time and the loss of user information.

# USER PREFERENCE CLUSTERING PROCESS

## Item attribute rating matrix based on user preference

Step 1: In general, the recommended system includes m user set U={$u_1$, $u_2$, $u_3$, ..., $u_m$} and n item set $I$={$i_1$, $i_2$, $i_3$, …, $i_n$}. Constructing user-item scoring matrix and transforming the score data of the user set U to item set I into user-item scoring matrix R (m, n), where $r_{ui}$ represents the score of user u to item i. The range of $r_{ui}$ is 0-5, which indicates the user's preference for the item. As shown in Table 1 below:

**TABLE 1.** User-Item Scoring Matrix

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | … | $i_n$ |
|---|---|---|---|---|---|---|
| $u_1$ | 2 | 3 | 0 | 2 | … | 2 |
| $u_2$ | 3 | 2 | 1 | 1 | … | 0 |
| $u_3$ | 1 | 2 | 0 | 4 | … | 1 |
| … | … | … | … | … | … | … |
| $u_m$ | 2 | 2 | 1 | 2 | … | 2 |

Step 2: Definition 1: The item set of target user u is $I$u={$i|r_{ui}$>=1}, the item's attributive set A={unknown, action, adventure, animation, children, comedy, crime, record, drama, fantasy, horror, black, musical drama, love, mystery, science fiction, thriller, war, the West}. The number of different attributes in the statistical item set $I_u$ is $N_a$, and the degree of preference of user u to different attributes is calculated according to the formula (1).

$$P_{ua} = N_a / \sum_{a \epsilon A} N_a \tag{1}$$

Step 3: Generating an item attribute scoring matrix for user preference, and calculates the sum of the user's scores on different attributes according to the formula (2). Among it, the $I_{ua}$ suggests that the attribute a' set is contained in the user u scoring item, $P_{ua}$ represents the proportion of the attribute a' set to the user u scoring item, $r_{ui}$ represents the score of user u to item i, and $\overline{r_u}$ suggests the average score of user u to items that already rated. The item attribute scoring matrix for user preference is shown in Table 2.

$$R_{ua} = \sum_{i \epsilon I_{ua}} P_{ua} * (r_{ui} - \overline{r_u}) \tag{2}$$

**TABLE 2.** The Item Attribute Scoring Matrix for User Preference

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | … | $A_a$ |
|---|---|---|---|---|---|---|
| $u_1$ | 0 | 0.875 | 0 | 1.42 | … | 1.23 |
| $u_2$ | 3 | 2 | 1.45 | 6.34 | … | 0 |
| $u_3$ | 1 | 0.12 | 0.421 | 0.341 | … | 1.67 |
| … | … | … | … | … | … | … |
| $u_m$ | 0.562 | 2 | 3.65 | 0.452 | … | 7.45 |

Step 4: Using the min-max standardization method to normalize the item attribute scoring matrix of user preference.

## User preference clustering

In the clustering algorithm, the k-means clustering algorithm is the most common and the easiest to understand. The k-means clustering algorithm is mainly divided m objects into k clusters. First, k elements are randomly selected from D as the centers of k clusters respectively. The Euclidean distance formula (3) is used to calculate the distance between the remaining elements and the k clusters centers, and the elements are classified into the nearest cluster. According to the results of clustering, the respective centers of k clusters are recalculated. The computing method is to adopt the arithmetic average of the respective dimensions of all the elements in the cluster. Then all the elements in the D are clustered again according to the new center until the clustering results are no longer changed.

$$D(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2} \tag{3}$$

## Similar clusters of target users and the selection of nearest neighbors

Using cosine similarity formula (4) computes the similarity between the target user and each cluster center, defining the first α clusters of highest similarity as the similar cluster set of target user $\mathrm{clust_a} = \{c_1, c_2, c_3, \dots, c_\alpha\}$. Then calculating the similarity between the target users and the users in the similar clusters, and selecting N users of the most similar degree to the target users. Finally get the nearest set of each cluster and the target user $N_{c_l} = \{u_1, u_2, \dots, u_N, c_l \in \mathrm{clust}_\alpha\}$.

$$sim(u_i, u_j) = \cos(\overrightarrow{u_i}, \overrightarrow{u_j}) = \frac{\overrightarrow{u_i} \cdot \overrightarrow{u_j}}{|\overrightarrow{u_i}| * |\overrightarrow{u_j}|} \tag{4}$$

Among them, $\overrightarrow{u_i}$ and $\overrightarrow{u_j}$ represent the score of user $u_i$ and $u_j$ to item attribute respectively.

# IMPROVED LATENT FACTOR MODEL

## Definition of Latent Factor Model

Funk-SVD is a matrix dimension reduction method commonly used in the recommended system, which decomposes the user's scoring matrix $R_{UI}$ into two low-dimension matrix $P_{U\times F}$ and $Q^T_{I\times F}$. Among them, the parameter F is the number of latent feature factors, and $P_{U\times F}$ represents the vector matrix of the user' latent feature factors, and $Q^T_{I\times F}$ represents the vector matrix of things' latent feature factors.

$$R_{UI} = P_{U\times F} Q^T_{I\times F} \tag{5}$$

Using matrix $P_{U\times F}$ and $Q^T_{I\times F}$ to obtain the predicted score of user u for the item i is:

$$\hat{r}_{ui} = \sum_f^F p_{uf} q_{if} \tag{6}$$

Among them, $p_{uf}$ represents the degree of association between the user u and latent feature factor f, $q_{if}$ represents the degree of association between the item i and latent feature factor f; in order to get the result of matrix $P_{U\times F}$ and $Q_{I\times F}$, loss function was introduced:

$$C(p, q) = \sum_{(u,i)\in \mathrm{Train}} (r_{ui} - \hat{r}_{ui})^2 = \sum_{(u,i)\in \mathrm{Train}} \left(r_{ui} - \sum_f^F p_{uf} q_{if}\right)^2 \tag{7}$$

Among them, $r_{ui}$ represents the real score of user u to item I; Train represents the training dataset. In the training process, regular terms $\lambda(\|p_u\|^2 + \|q_i\|^2)$ were added to avoid overfitting, λ is the regular term parameter, then the loss function converted into:

$$C(p, q) = \sum_{(u,i)\in \mathrm{Train}} \left(r_{ui} - \sum_f^F p_{uf} q_{if}\right)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2) \tag{8}$$

the parameters $p_{uf}$ and $q_{if}$ are derived respectively, and advance toward the direction of the quickest gradient descend. The recurrence formula is obtained as follow:

$$p_{uf} = p_{uf} + a(q_{if}(r_{ui} - \hat{r}_{ui}) - \lambda p_{uf}) \tag{9}$$

$$q_{if} = q_{if} + a(p_{uf}(r_{ui} - \hat{r}_{ui}) - \lambda q_{if}) \tag{10}$$

## Improvement of Latent Factor Model

Due to the loss of partial feature information in the process of matrix decomposition, the nearest user corrects the predictive score to get a more accurate one.

$$\hat{r}_{ui,c_l} = \hat{r}_{ui,c_l} + \frac{\sum_{k\epsilon N_{c_l}} (r_{ki} - \hat{r}_{ki,c_l}) \cdot \mathrm{sim}(u,k)}{\sum_{k\epsilon N_{c_l}} \mathrm{sim}(u,k)} \tag{11}$$

Among them, $N_{c_l}$ represents the most similar N users with user u in the similar cluster $c_l$, $r_{ki}$ represents the actual score of user k to the item i, $\hat{r}_{ki,c_l}$ represents the predictive score of user k to item i in the similar cluster $c_l$, $\mathrm{sim}(u, k)$ represents the similarity between the user u and the user k.

## Final Prediction Score

According to formula (11), the predictive score set of the similar cluster of target users was calculated. And weighted summation was conducted by use of the similarity between target user and similar cluster and predictive score set, then final predictive score of target user to item i was obtained.

$$\hat{r}_{ui} = \frac{\sum_{c_l \epsilon clust_\alpha} sim(u,c_l) * \hat{r}_{ui,c_l}}{\sum_{c_i \epsilon clust_\alpha} sim(u,c_l)} \tag{12}$$

Among them, $\hat{r}_{ui,c_l}$ represents the prediction score of the target user u in the cluster $c_l$ for project i. $sim(u, c_l)$ represents the similarity between the target user u and the similar cluster.

# EXPERIMENT AND RESULT ANALYSIS

## Experimental Data

The experimental dataset in this article uses the Movielens dataset established by the GroupLens item group in the United States. The dataset includes 943 users' information, 1682 movies information, and 100 thousand scoring data. In the experiment, the dataset is divided according to the ratio of 80% and 20%. Among them, 80% of the data as a training sample to train the model, and 20% of the data are to test the model as the test data, and the corresponding test results are statistically analyzed.

## Measure Standards

The accuracy of the recommended system prediction is an important indicator of evaluating the recommended system. Mean absolute error (MAE) and root mean square error (RMSE) are usually used in the field of recommended system to measure the accuracy of the predictive score. The smaller the value of the two methods, the smaller the error and the higher the accuracy of the prediction. Since RMSE is more punitive for error prediction, this paper uses RMSE as a measure standard.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in T}(r_{ui} - \hat{r}_{ui})^2}{|T|}} \tag{13}$$

Among them, T represents the test set, $r_{ui}$ represents the actual score of the user u to the item i, $\hat{r}_{ui}$ represents the predictive score of the user u to the item i.

## Analysis of Results

To verify the performance of the improved algorithm proposed in this paper in the predictive score, we have compared it with the following two algorithms: user-based collaborative filtering algorithm (UserCF) and latent factor model (LFM). In the improved algorithm, there are 6 main parameters: the number of user clustering k, the implicit feature F, the learning rate a, the regularization parameterλ, the iterated numbers m and the neighboring number N. Through experiments, it is found that the number of implied feature F, iterations m and neighboring numbers N are the most important parameters that affect the experimental results.

Therefore, under the condition of fixed learning rate a=0.01, user clustering number k=10, regularization parameter=0.025, neighboring number N=20, similar cluster number α=5 and iterated number m=100, we studied the influence of the number of implicit feature F on the recommended accuracy of the improved algorithm. The results of the experiment are shown in the following figure:

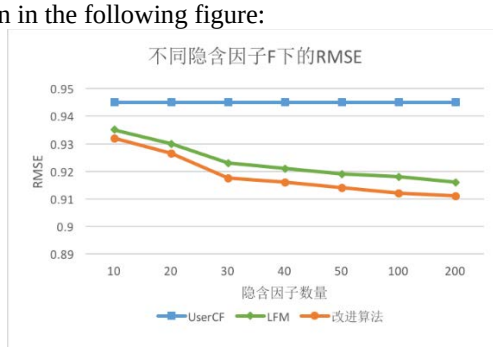


**FIGURE 1.** RMSE Value under Different F Values.

Since UserCF does not introduce an implicit factor, the accuracy of the UserCF algorithm will not change with the change of implied factors. As can be seen from the figure, the root mean square error of the LFM and the improved algorithm (RMSE) decreases with the increase of the number of implied factors. The improved algorithm

effectively compensates for the loss of the implicit feature information caused by matrix decomposition, and has a better recommendation accuracy than the LFM.

In order to study the effect of the selected neighboring number N on the accuracy of the recommendation, we have conduct an experiment. Consuming fixed learning rate a=0.01, user clustering number k=10, regularization parameter λ=0.025, implicit feature F is 40, similar cluster number α=5, iterated number m=100. The results of the experiment are shown in Figure 2.

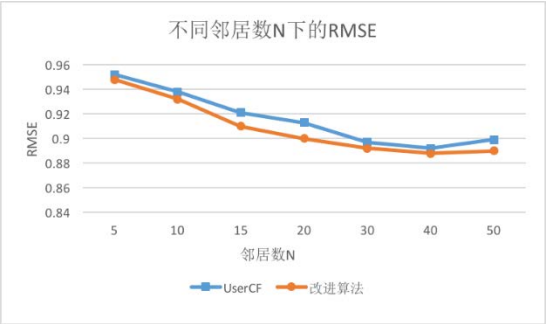


**FIGURE 2.** RMSE Value under Different N Values.

The results of Figure 2 show that the recommended accuracy of the user-based collaborative filtering algorithm and the improved algorithm will increase with the increase of the neighboring number N. When the range of neighboring number N is between 30 and 40, the recommended quality of the algorithm is best.

At the same time, this paper also considers the comparison between the LFM algorithm and the improved algorithm in the running time. In order to obtain a better predictive score matrix, it usually requires many iterations to obtain an ideal one. Under the condition of the fixed learning rate a=0.01, user clustering number k=10, regularization parameter $\lambda$=0.025, similar cluster number α=5, neighboring number N=35 and implicit factor number F=40, we conducted the experiment by adjusting the number of iterations m.

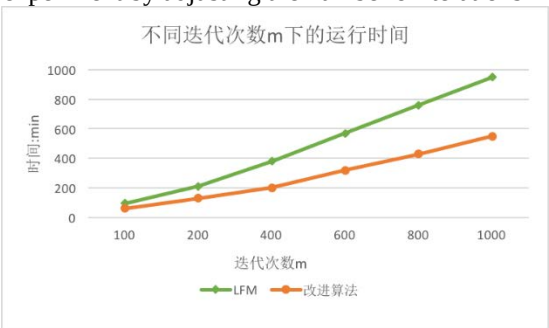


**FIGURE 3.** Model Running Time.

The results in figure 3 show that the training time of the LFM model is mainly affected by the number of iterations, and the improved model training time is mainly affected by the user clustering and the number of iterations. Although user clustering has consumed some time, the scale of the user rating matrix was reduced, which saves the total running time of the model. With the increasing number of iterations, the running time of the improved model is obviously lower than the the LFM model.

## CONCLUSION

Aiming at the problem of long running time and the loss of user information in the LFM-based collaborative filtering algorithm, this paper proposes a recommended algorithm of Latent Factor Model Fused with User Clustering. To a certain extent, the problem of user information loss is alleviated, and the training time of the model is greatly reduced. However, the model also has some defects, which has not effectively solved the cold boot of the recommended system, and also pointed out the direction for the follow-up study.

## REFERENCES


1. Rong Huigui, Huo Shengxu, Hu Chunhua, Mo Jinxia. Collaborative Filtering Recommended Algorithm Based on User Similarity [J]. Journal of China Institute of communications, 2014, 35 (02): 16-24.
2. Sarwar B，Karypis G，Konstan J，et al. Item-based collaborative filtering recommendation algorithms[C]. Proceedings of the 10th International Conference on World Wide Web，2001:285-295.
3. Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann. *Machines for Collaborative Filtering[C]*. Proc of the 24th International Conference on Machine Learning. 2007: 791-798.
4. Huang Liwei, Liu Yanbo, Li Deyi. The Recommended System Based on Deep Learning [J/OL]. Computer journal, 2017:1-29.
5. Yang Yang, Xiang Yang, Xiong Lei. Collaborative Filtering Recommended Algorithm Based on Matrix Decomposition and User's Neighboring Model [J]. Application of Computer, 2012,32 (2): 395 - 398.
6. Shi Yue, Martha L, Alan H. Mining Contextual Movie Similarity with Matrix Factorization for Context-aware Recommendation [J]. ACM Transactions on Intelligent Systems and Technology, 2010, 4( 1) : 385-388.
7. Wu Ke, Zhan Yinwei, Li Ying. Recommended Algorithm of Latent Factor Model Fused with User' Attribute [J]. Computer Engineering, 2016,42 (12): 171-175.
8. Ming Xiaohong. Research on Collaborative Filtering Recommended Algorithm Based on User Clustering [D]. Beijing Jiaotong University, 2017.