Collaborative filtering recommendation algorithm based on KNN and Xgboost hybrid

Yingxian Li*, Junwu Xu†, Min Yang‡

College of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China
*21705010016@stu.wit.edu.cn

Abstract: In the traditional collaborative filtering recommendation algorithm, it is easy to fall into the dilemma of local optimization due to single classification, which affects the recommendation effect of the algorithm. A hybrid collaborative filtering recommendation algorithm based on KNN-Xgboost is proposed. The algorithm uses KNN to fill in the user's predictive evaluation with less project evaluation, reducing the sparseness of the matrix. Then the Xgboost algorithm is used to implement multi-classifiers to predict the data, and the multi-class calculation results are calculated to form the recommended results. The experimental results show that the sparseness of the matrix is solved to a certain extent, and the recommendation accuracy is improved.

1. Introduction
With the advent of the 5G era and rapid changes in network resources, problems such as information overload and the inability to quickly obtain useful information are accompanied by problems. Promote the birth of the recommendation system and become an indispensable tool for major e-commerce companies to promote their products. The recommendation system with personalized recommendation effects [1] has brought huge benefits to major e-commerce online merchandise sales. Collaborative filtering [2] is currently the most used and most widely used personalized recommendation technology in recommendation systems. Collaborative filtering is usually divided into two mainstreams: user-based collaborative filtering[3] and item-based collaborative filtering[4]. The former calculates the correlation between the user and the product, and the latter calculates the user's history to recommend high-scoring products to the target user. The recommendation results obtained by these two algorithms will fall into the best effect of a single attribute, but more than one attribute affects the recommendation effect. For example, the user's gender, age, role movie category, etc. will affect the final recommendation result. Not only that, as the amount of data increases, the dimensionality of the data matrix will also increase, and matrix sparseness will also increase, resulting in a decrease in recommendation effect.

2. Related work
KNN recommendation algorithm:
The working principle of the KNN recommendation algorithm [5] is to find k nearest neighbor sets by calculating the similarity between (commodity) users, and recommend the product Top-N with the highest score to the target user.
2.1. Similarity calculation

The most commonly used methods for calculating user and product similarity are Pearson similarity and cosine similarity calculation methods. Assuming that there are m users and n types of products in the sample data set, the data set of product ratings is expressed as a "user-item" rating matrix $R_{m \times n}$.

- **Cosine similarity**:
  \[
  \text{sim}(u, v) = \cos(u, v) = \frac{u \cdot v}{\|u\| \|v\|}
  \]

- **Pearson correlation**:
  \[
  \text{sim} = \frac{\sum_{i,j} (r_{ui} - \overline{r_u})(r_{vj} - \overline{r_v})}{\sqrt{\sum_{i,j} (r_{ui} - \overline{r_u})^2 (r_{vj} - \overline{r_v})^2}}
  \]

  Among them $r_{ui}, r_{vj}$ are the respective ratings of user u and user v for item i; $\overline{r_u}, \overline{r_v}$ are the mean values of user u and user v for their respective rating items.

2.2. $k$ nearest neighbors selection

Through similarity calculation, the k users with the highest similarity to the target user are selected and combined into a set, which is the k nearest neighbor choices.

2.3. Forecast recommendation

Suppose that the nearest neighbor set of user a is $H_a$, and the user score of the nearest neighbor set is used to predict the target user’s score for the item not participating in the score.

\[
  p_{a,i} = \overline{r_a} + \frac{\sum_{j \in H_a} \text{sim}(a,j)(r_{ji} - \overline{r_j})}{\sum_{j \in H_a} |\text{sim}(a,j)|}
  \]

Finally, the item set N with the highest predicted score is used as the recommendation set Top-N set of the target user a.

3. the algorithm of this paper

3.1. Improved user similarity calculation

Suppose the user set is $U = U_1, U_2, U_3 \cdots U_{m-1}, U_m$, the product set is $I = I_1, I_2, I_3 \cdots I_{n-1}, I_n$, the "user-item" equal diversity set is $R$, and $R_{ui}$ represents the rating of user u on item i.

The traditional KNN similarity calculation method is only calculated for two users who rated the same item. If the two users have few items evaluated together, the calculated similarity is higher. However, in reality, the two products may have very different preferences. Therefore, predicting and calculating the items that users have not evaluated can reduce the sparseness of the matrix to a certain extent. The improved user similarity calculation formula:

\[
  \text{sim} = \frac{\sum_{i \in l_{u-v}} (r_{ui} - \overline{r_u})(r_{vj} - \overline{r_v}) + \sum_{l \in I_{u-v}} (r_{ui} - A)(r_{vi} - A)}{\sqrt{\sum_{i \in l_{u-v}} (r_{ui} - \overline{r_u})^2 (r_{vj} - \overline{r_v})^2 + \sum_{l \in I_{u-v}} (r_{ui} - A)^2 (r_{vi} - A)^2}}
  \]

Where $l_{u-v}$ represents the set of user u rating item i and user v not rating item i, $A$ represents the rating of item i by user u.

3.2. Modified cosine similarity

The original cosine similarity calculation formula can not complete the calculation of different scoring scales of items by different users, and the modified cosine similarity can eliminate the above problems.

\[
  \text{sim}(a,b) = \frac{\sum_{i \in l} (r_{ai} - \overline{r_u})(r_{bi} - \overline{r_v})}{\sqrt{\sum_{i \in l} (r_{ai} - \overline{r_u})^2} \cdot \sqrt{\sum_{i \in l} (r_{bi} - \overline{r_v})^2}}
  \]

Among them, a and b respectively correspond to the ratings of users u and v.

3.3. Xgboost algorithm description

The Xgboost algorithm [6] uses a boosting method, The boosting method [7] regards the results obtained
by multiple weak classifiers as continuous values, and these continuous values can be regarded as the value of the loss function, so that weak classification can be used for iterative training to achieve the optimization model effect. The objective function of the Xgboost algorithm, that is, the algorithm loss function, constructs an optimization model by constructing a minimized loss function. The Xgboost model contains multiple CART trees, so the objective function of the model:

$$O_j(\theta) = \sum_{i=1}^{n} L(y_i, \tilde{y}_i) + \sum_{k=1}^{K} \omega(f_k)$$  \hspace{1cm} (6)

- Regularization of the objective function:
  $$L(\emptyset) = \sum_{i=1}^{n} L(y_i, \tilde{y}_i) + \sum_{k=1}^{K} \omega(f_k)$$  \hspace{1cm} (7)
  $$\omega(f) = \alpha T \frac{1}{2} \beta \|\omega\|^2$$  \hspace{1cm} (8)

$T$ represents the number of nodes in the Xgboost Tree, and $\omega$ represents the evaluation results of each node on the product.

- The final loss function is:
  $$L(\emptyset) = \sum_{i=1}^{n} L(y_i, \tilde{y}_i) + \alpha T \frac{1}{2} \beta \sum_{t=1}^{T} \omega_{t,i}^2$$  \hspace{1cm} (9)

- Hybrid algorithm implementation steps:
  In the algorithm design, $U$ represents the user set, $I$ represents the item set, and $R$ represents the user's rating set for the item.
  Step1: Given data set$(U_1, I_1, R_1) \ldots (U_n, I_n, R_n)$, $\{U_i \in U, I_i \in I, 0 \leq R_i \leq 5\}$;
  Step2: The given data set adopts improved similarity calculation method to calculate its similarity;
  Step3: According to the improved KNN algorithm, obtain $k$ neighbor sets as the input of the Xgboost algorithm;
  Step4: Calculate the smallest loss function according to the final $L(\emptyset)$;
  Step5: Results calculated by multiple weak classifiers $F(x) = \text{sim}(\Sigma_{i=1}^{K} W_i, R_i)$;
  Step6: The sorted set is sorted in descending order of score to form a set;
  Step7: Recommend Top-N products from the classification set to target users.

4. Experimental results and analysis

4.1. Experimental data

The experimental data in this article comes from the ml-1m data set provided by the Movielens site, in which 6,040 users rated 3,883 movies with 1,000,209 reviews. The 8:2 training set and test set are randomly divided 5 times to form 5 different training sets and test sets. Finally, the average of the training and test results of all groups is used as the result of the evaluation algorithm in this paper, and the Top-N product with the highest score is recommended to the target user.

4.2. Evaluation algorithm index

The most commonly used indicators for recommendation algorithms are RMSE, MAE, Recall and accuracy (Precision). They can more intuitively reflect the accuracy of the recommendation algorithm. Where $P$ is the number of positive cases in the sample, $N$ is the number of negative cases in the sample, $TP$ is the number of data in the predicted data that are true positives, and $TN$ is the number of data in the predicted data that are true negatives.

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (p_i - q_i)^2}$$  \hspace{1cm} (10)

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^{m} |p_i - q_i|$$  \hspace{1cm} (11)

$$\text{recall} = \frac{TP}{P}$$  \hspace{1cm} (12)

$$\text{precision} = \frac{TP + TN}{P + n}$$  \hspace{1cm} (13)

The settings of each parameter in the algorithm are shown in Table-1, where Max_depth represents the depth of the spanning tree. Too small or too large will easily result in data overfitting and affect the
recommendation accuracy. After several debugging results comparison, when the depth of the tree is set to 3, the recall rate and accuracy rate of the algorithm in this paper reach the highest.

| Seed | Learning_rate | Max_depth | Num_boost_round | Random_state | Eta | Nthread |
|------|---------------|-----------|-----------------|--------------|-----|---------|
| 7    | 0.05          | 3         | 700             | 7            | 0.3 | 4       |

4.3. Experimental result set analysis

In order to verify the recommendation effect of the recommended algorithm in this paper, the root mean square error (RMSE) evaluation index is used to compare the algorithm in this paper with the traditional KNN recommendation algorithm on the same data set. The result is shown in Figure 1:

In order to further verify the recommended accuracy of the algorithm in this paper, the average absolute error (MAE) evaluation index is also used to compare the algorithm in this paper with the traditional KNN collaborative filtering algorithm CMA algorithm. The result is shown in Figure 2:
Experiments show that the value of K affects the recommendation result. Too small K leads to the model being too simple and does not achieve the expected effect; too large K value is prone to fitting, which affects the recommendation effect. According to the comparison of Figure 1 and Figure 2, it can be seen that when the K value is 150, the ideal K value is reached. The length of the recommendation result Top-N sequence will also affect the accuracy of the recommendation. Therefore, in the experiment, Top-N is set to \{5, 10, 15, 20, 25\}. According to actual needs, if the recommendation result Top-N is too long, it will not achieve the effect of accurate recommendation for the target user. Instead, it becomes the user’s burden. Therefore, the Top-N setting should not be too long.

![Figure 3: Comparison of recall rates of several algorithms](image)

![Figure 4: Comparison of accuracy of multiple algorithms](image)

The recall rate and accuracy rate of the recommended results of different algorithms are compared. According to Figure 3 and Figure 4, the results show that the results obtained in the Recall and Precision of the recommended results are all due to several other algorithms. The accuracy rate also increases with the length of the Top-N sequence, so collaborative filtering recommendation based on Xgboost multi-classifier is better than traditional collaborative filtering recommendation.

5. Conclusion
The algorithm in this paper analyzes the traditional collaborative filtering recommendation algorithm, based on the fact that a single classifier is easy to form a local optimal problem. The user similarity
calculation method is improved to make the recognition of the K nearest neighbor set users more accurate, and then the Xgboost multi-classifier is used to average the results, so that the recommendation result can reach the global optimum. Experimental results show that the improved hybrid recommendation algorithm is effective for local optimization problems and obtains more accurate recommendation results.

Acknowledgments

This work was supported by the Wuhan Institute of Technology Software engineering laboratory and the eleventh Graduate Innovation Fund of Wuhan Institute of Technology (CX2019235, CX2019236,).

References

[1] Lee Gun Ho, Lee Dong Hun. Recommending System of Products on e-shopping malls based on CBR and RBR[J]. The KIPS Transactions:PartD, 2004, 11D(5).
[2] Dong-Seop Lee, Gye-Young Kim, Hyung-Il Choi. A Web-based collaborative filtering system[J]. Pattern Recognition, 2003, 36(2).
[3] Jianrui Chen, Chunxia Zhao, Ulji, Lifang Chen. Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering[J]. Complex & Intelligent Systems, 2020, 6(6).
[4] Anand Shanker Tewari. Generating Items Recommendations by Fusing Content and User-Item based Collaborative Filtering[J]. Procedia Computer Science, 2020, 167.
[5] Li Luo, Xiao Ling Gao, De Jun Wu. The Research of Cloud Platform Based on KNN Collaborative Filtering Algorithm in Fruits Marketing System[J]. Applied Mechanics and Materials, 2013, 2594.
[6] Jinxiang Chen, Feng Zhao, Yanguang Sun, Yilan Yin. Improved XGBoost model based on genetic algorithm[J]. International Journal of Computer Applications in Technology, 2020, 62(3).
[7] Peter B ü hlmann, Bin Yu. Boosting[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(1).