Design of Hybrid Recommendation Algorithm Based on User Dynamic Behavior and Static Attributes

Xiaoxiao Dai*
Computer School, Beijing Information Science & Technology University (BISTU), Beijing City 100085, China
*Corresponding author e-mail: 2018021001@mail.bistu.edu.cn

Abstract. Collaborative filtering recommendation algorithm is one of the most successful recommendation algorithms, but the traditional collaborative filtering recommendation algorithm is proved to have a series of problems in data sparsity, cold boot and scalability. Based on the problems mentioned above, this paper analyses user dynamic behavior and a hybrid recommendation algorithm based on user dynamic behavior and static attributes (UDBSA) was proposed in this paper. By determining the optimal value of BP critical point for the number of ratings, the recommendation strategy was dynamically selected according to the number of ratings. This method can comprehensively alleviate the influence of problems of new users and concept drift related problem on recommendation results.

Keywords: Collaborative Filtering, Recommendation Algorithm, User Attributes

1. Introduction
In the period of rapid development of Internet in China, the e-commerce industry represented by Taobao and Jingdong.COM and new media industry represented by Youku, iQIYI.COM and Tencent video, etc. bring convenience to users, but also make it impossible for network users to find useful information from massive information within a reasonable time. In view of the above situation, the emergence and application of recommendation system plays an important role in solving the problem of "information overload" encountered by users facing massive data on the Internet [1].

1.1. Traditional Collaborative Filtering Recommendation Algorithm
The traditional filtering algorithm can find congenial user groups by analyzing the previous feedback of users, exploring the preferences of users. Besides, the scores on prediction can be made to intuitively show the interest level that the users displayed so as to make corresponding recommendations, just like user-based collaborative filtering (UBCF) [2].

There is also item-based collaborative filtering (IBCF) [3]. This kind of algorithm is applicable to semi-structured or unstructured data that is difficult to extract the attributes of items [4]. With the massive increase in the number of users and network information, this kind of recommendation algorithm gradually shows "fatigue", and "cold start", "concept drift" and other problems gradually become urgent problems to be solved in the recommendation field.
1.2. Problem Description

1.2.1. Problems in "cold boot": This problem is mainly caused by the incomplete establishment of user interest model due to sparse user information data in the face of new users of the system. In addition, when a new project of the system is established, the behavior record is too small due to the weak correlation between the user and the project, which will be a cause for this problem.

1.2.2. "Concept drift": This concept first appeared in the field of machine learning. In recent years, with the introduction of machine learning algorithms in recommendation algorithm for further improvement, and with the passage of time, the user's interests will undergo corresponding changes with the generation of implicit condition. Other implicit variables are hard to be captured by the system to meet the emands in non-inductive experience, and the temporal threshold can be recognized by the system. As a result, the result is largely derivated from the practical interests [5].

2. Introduction to Relevant Algorithms and Evaluation Indexes

The evaluation index of the result of the recommendation algorithm needs to be considered through multiple dimensions. Two evaluation methods in this paper were selected for comprehensive consideration: prediction accuracy and classification accuracy.

2.1. Precision on Prediction

The commonly used evaluation criteria for prediction precision are Mean Absolute Error (MAE) [6] and Normalized Mean Absolute Error (NMAE) [7]. Supposing that the recommendation system conducts prediction scores for n times for the users \( A = \{a_1, a_2, a_3, \ldots, a_n\} \) and the real scores for users are \( R = \{r_1, r_2, r_3, \ldots, r_n\} \). Then the evaluation equation of MAE based on the scores of users for N times can be expressed as:

\[
\epsilon_a = \frac{\sum_{i=1}^{n} |A_i - R_i|}{n}
\]  

NMAE standardized and optimized the score on the basis of MAE, and the evaluation equation can be expressed as:

\[
\epsilon_a = \frac{\text{MAX}}{r_{\text{max}} - r_{\text{min}}} \left( \frac{\sum_{i=1}^{n} |A_i - R_i|}{n \times (r_{\text{max}} - r_{\text{min}})} \right)
\]

2.2. Precision on Classification

Classification accuracy is a method to measure the accuracy score of dichotomy. The classification algorithm is used to determine whether the recommendation algorithm recommends items that meet the user's interests or filters out items that are disliked. The commonly used classification accuracy has the accuracy rate, recall rate. The calculation formula of accuracy is as follows:

\[
P = \frac{N_{rs}}{N_s} = \frac{N_{rs}}{N_{rs} + N_{ri}}
\]

The formula for recall rate is as:
3. Hybrid Recommendation Algorithm based on User Attribute Extraction

The traditional recommendation is based on the dynamic users to the static project. In this case, the recommendation system is based on the static user to the dynamic project for implementation. The static nature of the item causes it to be unable to establish a user-project relationship, which makes new user based issues difficult. Therefore, it can be considered that user attributes and project labels are added to the algorithm in improvement as portraits of users and projects respectively, and the relationship between users and projects can be established through clustering of both sides.

3.1. Recommendation Algorithm (UDTB) Based on User Dynamic Timing Behavior

This section proposes a recommendation algorithm based on user timing. Based on the time characteristics of user behavior, the behavior time attribute is added to the core user-item matrix in the IBCF algorithm, and the scoring time label is added to the scoring record value of each user behavior. The fitting function [8] is selected to simulate the forgetting process of users, and an improved method based on the time weight of the forgetting curve is proposed. The calculation formula is as follows:

\[ w(i, j) = e^{-\log_2 \varepsilon (t - t_{ij})} \]

\( w(i, j) \) represents the time weight of user I's scoring record of item J in the scoring matrix. \( t_{ij} \) denotes the time of the score, \( t \) denotes the current time, and \( \varepsilon \) denotes the weight parameter.

Two methods of similarity calculation in IBCF algorithm were selected for improvement: cosine similarity and Pearson correlation coefficient method. The corresponding optimization equation are shown in equations 6 and 7.

\[
Sim(i, j) = \frac{\sum_{u \in U_i} \omega(u, i) \cdot (s_{ui} - \bar{s}_i) \cdot \omega(u, j) \cdot (s_{uj} - \bar{s}_j)}{\sqrt{\sum_{u \in U_i} \omega(u, i) \cdot (s_{ui} - \bar{s}_i)^2} \sqrt{\sum_{u \in U_j} \omega(u, j) \cdot (s_{uj} - \bar{s}_j)^2}}
\]

The optimization of equation of Pearson's coefficient is as follows:

\[
Sim(i, j) = \frac{\sum_{u \in U_i} \omega(u, i) \cdot (s_{ui} - \bar{s}_i) \cdot \omega(u, j) \cdot (s_{uj} - \bar{s}_j)}{\sqrt{\sum_{u \in U_i} \omega(u, i) \cdot (s_{ui} - \bar{s}_i)^2} \sqrt{\sum_{u \in U_j} \omega(u, j) \cdot (s_{uj} - \bar{s}_j)^2}}
\]

The time weight function is used to optimize the aggregation function formula. According to the items that the user has scored, the score is weighted to predict the user's score for unknown items. The calculation equation is shown as follows.

\[
\hat{r}_{c,j} = \frac{\sum_{j \in I_c} Sim(i, j) \cdot \omega(c, j) \cdot s_{c,j}}{\sum_{j \in I_c} Sim(i, j) \cdot \omega(c, j)}
\]
\( \hat{r}_{c,i} \) represents user C's predicted score for project I. \( I^c \) represents the collection of historical scoring items of user C, and J represents one of the items. \( \text{Sim}(i, j) \) denotes the similarity between project I and Project J, and \( s_{c,i} \) denotes the true score of J given by user C. \( \omega(c, j) \) represents the time weight of scores by user C to project J.

### 3.2. Hybrid Recommendation Algorithm based on Static Attribute (USAC) of Users

The attribute set of multiple dimensions can be used to describe different character objects. The attribute sequence of individual portrait in the attribute set is as follows:

\[
h_i = \{c_1, c_2, \ldots, c_n\}
\]

User information is collected to enable the platform to provide targeted personal services based on it. In addition to user information and item content, item categories are typically represented by short tags.

The specific optimization scheme of the hybrid recommendation algorithm (USAC algorithm) based on user static attribute is as follows.

The Jaccard coefficient method [9] was used to propose a formula for the improvement of calculation of user attribute similarity:

\[
\text{Res}(u, v) = \frac{|I'_u \cap I'_v| + \sum_{k=1}^{n} w_k}{|I'_u \cup I'_v|}
\]

\( I'_u, I'_v \) respectively represent subsets of discrete attributes in the attribute sets of user U and user V.

Supposing that there are n number of user attribute dimensions as continuous values, \( I'^u_u \) and \( I'^v_v \) respectively represent the set of continuous attributes of user U and V.

For the KTH continuous attribute owned by two users, it constitutes the calculation in which the attribute value pair is used and participated. The calculation equation is expressed as:

\[
w_k = 1 \cdot \frac{|x_k - y_k|}{E_k}
\]

Then, after clustering the user groups through similarity calculation [10], it is necessary to find the common interests in the set of neighbor users where the target user is located. By calculating the label frequency of all neighboring users who have visited the project, the labels are arranged in descending order of frequency:

\[
R = \{(l_1, r_1), (l_2, r_2), \ldots, (l_n, r_n)\}|r_1 \geq r_2 \geq \ldots \geq r_n|
\]

\( l_k \) represents the Kth tag corresponding to the item visited by the neighbor user, and \( r_k \) represents the frequency of the Kth tag. The weight score of the k tag is calculated by the frequency of the k tag in the proportion of all tags.

### 3.3. Hybrid Recommendation Algorithm based on User Dynamic Behavior and Static Attributes

In this section, two algorithms, USAC and UDTB, are combined to construct a hybrid recommendation algorithm (UDBSA) based on user dynamic behavior and static attributes.
The UDBSA algorithm dynamically switches to select the recommendation method based on the user rating state. The number of user ratings is the key point for algorithm selection. Therefore, this paper sets the boundary point BP for the number of scores as the basis to distinguish whether the target user is a new user or not: when the number of scores is lower than the BP point, it is a new user; otherwise, it is not. The optimized formula of the combined recommendation algorithm based on user attributes can be expressed as:

\[
I_{i,\text{rec}} = \begin{cases} 
I_{i,\text{UDTB}} & |S_i| < BP \\
I_{i,\text{USAC}} & |S_i| \geq BP
\end{cases}
\]

(13)

\(I_{i,\text{rec}}\) represents the collection of items recommended by the comprehensive recommendation method to user \(I\), \(I_{i,\text{UDTB}}\) represents the collection of items recommended by UDTB algorithm to user \(I\), \(I_{i,\text{USAC}}\) represents the collection of items recommended by USAC algorithm to user \(I\), and \(S_i\) represents the collection of scoring behaviors of user \(I\).

When new users arrive, the status of ratings by users should firstly judged. If the number of user ratings is lower than BP, USAC strategy will be adopted to recommend items to users. The user rating status will still be checked in the next recommendation. Until the number of end-user ratings exceeds BP value, UDTB strategy can be used to recommend users. After that, the user rating status will not be concerned and UDTB algorithm will be used all the time.

4. Experiment Section

This experiment used the Movie Lens rating dataset published by the Social Computing Research Group at the University of Minnesota. Experimental methods: The five-fold cross-validation method was adopted, and the above experimental sets were divided and recommended for 10 times. The Precision and Recall of the experimental results were averaged as the final experimental results.

![Figure 1. Influence of the number of historical scores on precision](image)

As can be seen from Figure 3, the number of intersection point BP of the two algorithms is equal to 9 in term of precision result. As shown in Figure 4, for recall results, the number of intersection points BP of the two algorithms is equal to 2. It can be seen from the above experimental results that the USAC method is not based on user ratings, so the number and order of ratings have no influence on the algorithm with no change in the experimental results. When the number of users decreases, the performance of UDTB is inferior to that of USAC in both verification methods. The precision of experimental results and recall rate of UDTB increase with the increase of the number of users’
historical scores, and UDTB will be better than USAC when the number of users reaches the point BP where the curves intersect.

**Figure 2.** Influence of the number of historical scores on recall rate

The UDBSA algorithm is compared with IBCF algorithm and UBCF algorithm, and the results are shown in Figure 3. The performance of UDBSA is superior to UBCF and IBCF. It is particularly true for the number of scores in stages 1-9, and during which the number of scores is judged to be new users. The recommendation strategy based on static attributes of users is used in UDBSA algorithm, which is suitable for the recommendation for users at this stage. Therefore, the precision is much higher than that of UBCF and IBCF. When the number of scores reaches the stage 9-15, UDBSA will be switch to UDTB algorithm. This strategy will also improve the precision with the increase of scores, and the recommendation results are still better than the other two algorithms. Therefore, the validity of UDBSA algorithm on precision is verified.

The influence of the number of user ratings on recall is shown in Figure 4. USAC has no obvious effect on recall rate in the period of improving new users. Only when the number of ratings is at stage 1-2, the recommended results of USAC algorithm can be compared with other UBCF and IBCF, and the recall will not change with the increase of the number of ratings. Therefore, UDTB algorithm will perform better than UBCF and IBCF in the stage of scoring 2-15. In summary, the experimental results verify the effectiveness of UDBSA algorithm.

**Figure 3.** Comparison of effectiveness of classification precision index
5. Conclusion
This paper introduces a hybrid recommendation algorithm based on dynamic behavior and static attributes of users, which recommendation strategies are dynamically selected according to the number of user ratings. This method can comprehensively alleviate the influence of new user problem and concept drift problem on recommendation, and at the same time, the effectiveness of the algorithm is verified by experiments.

In the future development of recommendation system, the key direction of algorithm for updating should be on the dimensional division and in-depth exploitation of user information, and on building a multi-dimensional information model for users in interest, so as to make the results recommended for users more precise.

References
[1] Feng Yuanyuan. E-commerce long tail item recommendation methods. Chinese Academy of Sciences. 2015, 2015 (how a02): 151-154.
[2] Daniel Valcarce, Javier Parpar, Alvaro Barreiro. Item-based relevance modelling of getting rid of long tail products. Knowledge-based systems, 2019, 103(C): 41-51.
[3] Wang Xiaojun. Using fuzzy segmentation to improve the expansibility and precision of collaborative filtering. Journal of Beijing University of Posts and Telecommunications, 2017, 40(1):74-78.
[4] Wang Jing. An optimized Item-based collaborative filtering recommendation algorithm. Journal of Chinese Mini-Micro Computer Systems, 19, 12(12):2337-2342.
[5] Rong Huigui. Collaborative filtering recommendation algorithm based on user similarity. Journal of Communications, 2019(2):16-24.
[6] Zhang Ziyang. Collaborative filtering recommendation algorithm based on the nearest neighbor optimization and selection method. Acta Scientiarum Naturalium Universitatis Nankaiensis, 2017(3):27-32.
[7] Fan Yongquan. An enhanced collaborative filtering recommendation method integrating user preference and trust. Journal of Xihua University (Natural Science Edition), 2015, 35(4):8-12.
[8] Huang Xianying. Personalized news recommendation technology based on improved collaborative filtering algorithm. Journal of Sichuan University (Natural Science edition), 2008, 55(1):49-55.
[9] Gao Qian, He Juhou. Improved collaborative filtering recommendation algorithm for data sparsity. Computer Technology and Development, 2016, 26(03):63-66.
[10] Wang Wei. Improvement of collaborative filtering algorithm based on user similarity.
of East China Normal University (Natural Science edition), 2016, 3:60-66.