Collaborative Filtering Algorithm Based on Improved Time Function and User Similarity

Weiguo Zhang, Xiran Zhou*, Weixuan Yuan
*Xi'an University of Science and Technology, Xi'an 710054, China

*305747503@qq.com

Abstract. As a typical representative of information filtering technology in the era of big data, a recommendation system is an important means to solve the problem of information overload. A collaborative filtering recommendation algorithm is one of the important technologies to realize the recommendation system, but the traditional collaborative filtering algorithm only considers the similarity of ratings between users. As the number of users and the number of items increases, it faces user interest drift and reduced recommendation Precision, and other issues. In this regard, a collaborative filtering algorithm based on improved time function and user similarity is proposed. First, considering that user interests will dynamically change over time, this paper introduces an improved time function in the traditional scoring similarity; secondly, considering the impact of the number of items evaluated by users on the calculation of similarity measurement, this paper takes the Pearson correlation coefficient weighted scoring into account; finally, the fusion recommendation is based on the improved time function and the weighted Pearson correlation coefficient to improve the Precision of recommendation prediction. This paper conducts simulation experiments on MovieLens 100K and 1M data sets respectively. The results show that, compared with the traditional collaborative filtering recommendation algorithm, combining the improved time function and the weighted Pearson correlation coefficient can effectively improve the recommendation Precision.

Keywords: Collaborative Filtering, Pearson Correlation Coefficient, Time Function, Similarity Algorithm.

1. Introduction
With the rapid development of the era of big data, the amount of data around people is increasing at an exponential rate. How to quickly and accurately find the required information in the huge data has become a current problem. In order to solve the above problems, a recommendation system was proposed. The recommendation system is a predictive platform based on the mining of massive data. It uses machine learning and data to provide suggestions to help users make decisions and provide personalized recommendations [1].

In recent years, collaborative filtering algorithms have become one of the most widely used recommendation algorithms. The theoretical basis is people’s herd mentality, which uses the preferences of groups with similar interests and common experiences to recommend information that users are interested in. Accurate acquisition of user interests is a very important research content of recommendation systems.
However, due to the influence of the environment, age, and other factors, everyone’s interests are constantly changing. The traditional collaborative filtering recommendation algorithm also has certain limitations. It believes that user rating time does not affect the measurement of similarity, but in fact, user interest changes are affecting the quality of system recommendations. At the same time, in the collaborative filtering recommendation algorithm, the calculation result of the similarity between users directly affects the recommendation quality of the system [2]. There are three traditional similarity calculation methods, but these three methods do not consider the scoring time, which does not affect the calculation of similarity between users or items. Although the initial impact on the system is small, as the time the user uses the system increases, the user's interests may change greatly [3]. At this time, using traditional methods will make the calculated similarity results inaccurate. In addition, in the calculation of prediction scores, the importance of each score is only distinguished by the magnitude of similarity, and the importance of user rating time is not distinguished, resulting in inaccurate prediction scores and reducing the Precision of recommendation results [4].

In order to solve these problems, this paper proposes a collaborative filtering algorithm based on an improved time function and user similarity. The algorithm takes into account the special changes of user interest over time and calculates the Pearson similarity to improve the recommendation results of user interest changes. So as to achieve the purpose of improving the recommendation Precision.

2. Process description of the improved algorithm

2.1. Application of time function

In the 19th century, the famous German psychologist Ebbinghaus proposed the Ebbinghaus forgetting curve, which is called a time function here, as shown in the figure below:

![Ebbinghaus forgetting curve](image)

Ebbinghaus found through experiments that people start forgetting immediately after learning, and the speed of forgetting is different. When forgetting begins, the speed is the fastest, dropping sharply, then gradually slowing down until it stabilizes, and then forgetting stops.

The change in user interest is roughly the same as the time function, but there are subtle differences. Therefore, in the early work, Ding et al. were inspired by the Ebbinghaus forgetting curve and designed a monotonically decreasing time function [5]. The time function formula is shown in the following:

\[
T(t) = e^{-\alpha t}
\]  

(1)
In (1), $t$ represents time; $\alpha$ is a parameter that controls the degree of information attenuation. Therefore, this paper follows the theoretical calculation principle, based on the traditional collaborative filtering recommendation algorithm, and describes the changes in user interest over time by improving the time function.

2.2. Improved time function
The user's interest in the project is not fixed but will change over time. For example, changes in the scene, the occurrence of major events, and the increase in age will affect the user's interests and preferences, and the user may be inclined to pursue new things and explore the world when he is young, and when he is older, he may prefer to keep stable living environment, prefer healthy activities [6]. However, in reality, the user's interest does not change all the time but may change every certain period of time. Since the traditional time function, as shown in (1), expresses the user's interest too dense and does not conform to real life. The fact that users use time periods as the basis for changes in interest [7]. Therefore, this section proposes an improved time function. Since the user's behavior in the most recent time period can best express the user's current interest, the time function in the collaborative filtering recommendation in this paper is shown in the following:

$$F_i = e^{-\frac{T_{u,i} - T_0}{T_{\text{max}} - T_0}}$$

(2)

In (2), $F_i$ the improved time function; $T_{u,i}$ represents the behavior time of the user $u$ on the item $i$; $T_0$ represents the user's initial rating time; $T_{\text{max}}$ indicates the time of the user's last rating. $T_{u,i} - T_0$ represents the time difference of users on the project, and the improved time function can scientifically reflect the relationship between user interest and time.

2.3. Similarity calculation
Similarity calculation has been widely used in data mining and recommendation systems. Commonly used similarity calculation methods mainly include cosine similarity, adjusted cosine similarity, Pearson correlation coefficient, Jaccard coefficient, and Tanimoto coefficient. Different calculation methods have different use environments, advantages, and disadvantages [8]. In the paper, based collaborative filtering, Euclidean, Pearson correlation coefficient, and cosine similarity algorithm are mainly used to calculate user similarity. The Euclidean algorithm is generally used in data-intensive situations, that is, almost all data has attribute values, and the magnitude of the attribute values is important [9]. The Pearson correlation coefficient algorithm is generally used when the data is affected by level expansion, that is, different users use different scoring standards. The cosine similarity algorithm is generally used when the data is sparse.

In real life, each user has different scoring standards, so this paper uses the Pearson correlation coefficient algorithm to calculate user similarity. Pearson correlation coefficient formula is shown in the following:

$$\text{sim}_u(a,b) = \frac{\sum_{i \in U} [R(a,i) - \overline{R(a)}][R(b,i) - \overline{R(b)}]}{\sqrt{\sum_{i \in U} [R(a,i) - \overline{R(a)}]^2 \sum_{i \in U} [R(b,i) - \overline{R(b)}]^2}}$$

(3)

In (3), $\text{sim}_u(a,b)$ represents Pearson correlation coefficient; $U$ is the set of items scored by a and b; $i$ is the item; $R(a,i)$ represents the rating of the item $i$ by current user $a$; $R(b,i)$ represents the
rating of the item i by current user b; \(\overline{R(a)}\) represents the average value of the items that user a has rated; \(\overline{R(b)}\) represents the average value of the items that user b has rated.

2.4. Pearson's correlation coefficient with weights introduced
Consider the influence of the total number of items evaluated by users on the calculation of similarity measure. For any two users, the number of items they scored together is fixed. The smaller the total number of user-rated items, the closer the similarity; on the contrary, the more the total number of user-rated items, the smaller the similarity. In the calculation of similarity, the number of common scores between users increases proportionally, and the total number of user rating items decreases inversely [10]. Therefore, this paper will define a new weight formula (4), and then enter the new weight into the Pearson correlation coefficient to calculate the relationship between users. The new weights are shown in the following:

\[
W = \frac{\delta}{2} \left( \frac{p_a + p_b}{p_a p_b} \right)
\]

(4)

\[
\text{Sim}(a, b) = \frac{\delta}{2} \left( \frac{p_a + p_b}{p_a p_b} \right) \text{sim}_2(a, b)
\]

(5)

In (4) and (5), \(W\) represents new weights; \(\text{Sim}(a, b)\) is the similarity coefficient with weight added; a is the current user; b is other users, and \(p_a\) are the total number of items rated by users a and b respectively; \(\delta\) is the total number of items with the same rating of users a and b.

The Pearson correlation coefficient after introducing the new weights is shown in the following:

\[
\text{Sim}(a, b) = \frac{\delta}{2} \left( \frac{p_a + p_b}{p_a p_b} \right) \frac{\sum_{i \in U} (R(a,i) - \overline{R(a)}) [R(b,i) - \overline{R(b)}]}{\sqrt{\sum_{i \in U} [R(a,i) - \overline{R(a)}]^2} \sqrt{\sum_{i \in U} [R(b,i) - \overline{R(b)}]^2}}
\]

(6)

Collaborative filtering recommendation integrating improved time function and user similarity
The improved time function and the weighted Pearson correlation coefficient are combined and recommended. The specific method is as follows: Combine formula (2) and formula (6), and propose a new user rating similarity formula, as is shown in the following:

\[
\text{Sim}_{\text{mix}}(a, b) = \frac{\delta}{2} \left( \frac{p_a + p_b}{p_a p_b} \right) \frac{\sum_{i \in U} [F_i R(a,i) - \overline{R(a)}) [F_i R(b,i) - \overline{R(b)}]}{\sqrt{\sum_{i \in U} [F_i R(a,i) - \overline{R(a)}]^2} \sqrt{\sum_{i \in U} [F_i R(b,i) - \overline{R(b)}]^2}}
\]

(7)

In (7), \(F_i\) is an improved time function, which is integrated into the weighted Pearson correlation coefficient to obtain the final formula \(\text{Sim}_{\text{mix}}(a, b)\). This formula is divided into two parts, the first part is the improved time function, and the second part is the weighted Pearson correlation coefficient.

3. Experimental results and analysis

3.1. Experimental data and environment
GroupLens is the earliest recommendation system to help users find movies they are interested in. MovieLens web site is a free experimental site for research purposes. The website has tens of thousands of registered users[11]. The creator GroupLens group collects the user's movie ratings in the
system as a MovieLens data set, which was later used by researchers, and it is widely used to measure the Precision of recommendation algorithms.

The experimental data of this experiment uses the 100K and 1M data sets in the MovieLens data set to conduct experiments. The rating points for movies are between 0 and 5 points, and the user’s preference for movies is directly proportional to the rating points. The data in the data set is divided into the training set and test set according to the ratio of 80% and 20%.

3.2. Recommended results inspection index

3.2.1. Precision

Precision is the most important indicator of the recommendation system algorithm. It marks the ability of the recommendation system algorithm to predict the Precision of user behavior. Simply put, it refers to the degree of overlap between the predicted behavior calculated on the training set in the experiment and the actual behavior of the user on the test set. The correct rate is calculated as the proportion of the top-N items that the target user recommended by the user has scored. The higher the ratio, the better the recommendation result. The Precision is positively correlated with the recommendation effect. The higher the Precision, the better the recommendation effect [12]. The Precision formula used for the recommendation algorithm is as follows:

\[
\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}
\]

(8)

In (8), \(U\) represents user collection; \(u\) represents a user in \(U\); \(R(u)\) represents a list of recommendations made to users based on the training set data, and \(T(u)\) represents a list of recommendations made to users based on the test set data.

3.2.2. Recall

Recall refers to the ratio of the number of related documents retrieved to the number of all related documents in the literature library, which measures the Recall of the retrieval system. The Recall used for the recommendation algorithm is as follows:

\[
\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}
\]

(9)

In (9), the \(R(u)\) and \(T(u)\) in the Recall are the same as those defined in the Precision.

3.3. Experimental analysis

In this paper, experiment one and experiment two are carried out on MovieLens 100K and MovieLens 1M data sets respectively.

**Experiment 1**: In the collaborative filtering recommendation algorithm, under the same similarity, choosing different neighbors will also produce different prediction Precision. Therefore, this paper compares the user-based collaborative filtering algorithm (UserCF), the collaborative filtering algorithm with penalty terms (UserCF-IIF), and the collaborative filtering algorithm based on improved time function and user similarity (UserTS-CF).

In the MovieLens 100K data set, UserCF, UserCF-IIF and UserTS-CF are compared. The Precision is shown in the following figure:
According to the above figure, the UserTS-CF proposed in this paper has a significant improvement in recommendation Precision on the MovieLens 100K data set.

In the MovieLens 100K data set, the Recall of UserCF, UserCF-IIF, and UserTS-CF in different neighborhoods are compared, and the Recall is shown in the following figure:

According to the above figure, the UserTS-CF proposed in this paper has a significant improvement in recommendation Recall on the MovieLens 100K data set.

**Experiment 2**: Precision and Recall take values between 0 and 1. The closer the value is to 1, the higher the Precision or Recall rate.

In the MovieLens 1M data set, the Precision of UserCF, UserCF-IIF, and UserTS-CF in different neighborhoods are compared. The Precision is shown in the following figure:
According to the above figure, the UserTS-CF proposed in this paper has a significant improvement in recommendation Precision on the MovieLens 1M data set.

In the MovieLens 1M data set, the Recall of UserCF, UserCF-IIF, and UserTS-CF in different neighborhoods are compared. The Recall is shown in the following figure:

### Fig. 5 Recall comparison under MovieLens 1M data set

According to the above figure, the UserTS-CF proposed in this paper has a significant improvement in recommendation Recall on the MovieLens 1M data set.

### 4. Conclusion

In order to solve the problems of user interest drift and reduction of recommendation Precision, this paper proposes a collaborative filtering algorithm based on improved time function and user similarity. The improved time function algorithm is introduced into the traditional scoring similarity, and it is Weighted scoring based on the inferior correlation coefficient that can improve the recommendation Precision. Finally, through the MovieLens data set, the superiority of the improved recommendation algorithm compared with the traditional collaborative filtering recommendation algorithm is verified and compared with the traditional collaborative filtering recommendation algorithm and the collaborative filtering algorithm with penalty items. Next, we apply the improved algorithm to the community
and realize real-time optimization of the recommendation results based on user attribute preference feedback.

References
[1] Jiang W, Yang L 2016 Research of improved recommendation algorithm based on collaborative filtering and content prediction Int. Conf. on Computer Science & Education 23-25 Okayama Japan (Piscataway: IEEE)
[2] Ferdaous H, Bouchra F and Brahim O 2018 Recommendation using a clustering algorithm based on a hybrid features selection method J. Intelligent Information Systems 51 p183–205
[3] Ar Y, Bostanci E 2016 A genetic algorithm solution to the collaborative filtering problem J. Expert Systems with Applications 61 pp122–128
[4] Dong L 2017 Dynamic Model Adaptive to User Interest Drift Based on Cluster and Nearest Neighbors (Piscataway: IEEE) p99
[5] Liang X 2012 J. Recommended system practice (Beijing: people's posts and telecommunications press) pp57-59
[6] Akermi I, Boughanem M and Faiz R 2017 Just-In-Time recommendation approach within a mobile context Int. Conf. on Web Intelligence (Piscataway: IEEE)
[7] Wang X 2017 A Collaborative Filtering Algorithm of Weighted Information Entropy and UserAttributes Int. Conf. on Intelligent Human-machine Systems & Cybernetics
[8] Azizi M, Do H 2018 A collaborative filtering recommender system for test case prioritization in web applications 33rd Annual ACM Symp. on Applied Computing pp1560–1567
[9] Zijian L 2020 An Improved Similarity Measurement Method in Collaborative Filtering Algorithm J. Pure Mathematics 10(05) pp404-413 DOI: 10.12677/PM.2020.105050
[10] Koren Y 2010 Factor in the neighbors scalable and accurate collaborative filtering J. ACM Transactions on Knowledge Discovery from Data 4(1) pp1-24
[11] Mohammadpour T, Bidgoli A M, Enayatifar Rasul and Javadi H H S 2019 Efficient clustering in collaborative filtering recommender system Hybrid method based on genetic algorithm and gravitational emulation local search algorithm J. Genomics (Amsterdam: Elsevier Science) p1016
[12] Vafa M, Danesh Z and Kasra Naderi 2019 Proposing a novel method for improving the performance of collaborative filtering systems regarding the priority of similar users J. Phys. A: Statistical Mechanics and its Applications