An Improved Slope One Algorithm Based on User Similarity Weights

Hang ZHANG¹, Sheng LIU², and Wencheng LIU³

¹School of software, Central South University, 410075 Changsha, China

²School of software, Central South University, 410075 Changsha, China

³School of software, Central South University, 410075 Changsha, China

ABSTRACT. This paper presents a Slope one improved algorithm based on user similarity and user interest forgetting function. Aiming at the problem of large number of users and a lot of noise data, the inactive users are filtered out by setting the threshold of user activity, and then the neighbors of the target users are obtained through the calculation of user similarity. According to interest forgetting function, and then filter out items that have less effect on current users to reduce the noise data to improve the accuracy of the algorithm. Experimental comparison shows that the improved algorithm has better accuracy than the commonly used weighted Slope one and two-pole Slope one.

1 Preface

Collaborative Filtering (CF) is the most widely used and most successful personalized recommendation algorithm, which is divided into two categories: User-based collaborative filtering algorithm and Item-based collaborative filtering algorithm. Slope one algorithm is a special form of Item-based collaborative filtering algorithm, which has the characteristics of easy to use, accurate recommendation and high computational efficiency. However, the problems such as cold start and data sparse restrict its development. To solve these problems, Proposed an improvement suggestion^[1], in which, for the data sparsity problem, the most common is to compress the original data by using the dimension reduction technique^[3], in addition, it can also directly remove irrelevant users or projects to reduce users - For the cold-start problem, some researchers point out that their neighbors can be searched initially according to the user attributes or project attributes, that is, the content-based recommendation^[4,5]. For the Slope one algorithm, the change of user interest is not considered And user similarity, this paper proposes an improved Slope One based on user interest forgetting function and user nearest-neighbor filtering strategy in order to provide more accurate recommendations for target users.

2. Slope One Introduction

When collaborative filtering is recommended, people need to build user-item scoring matrices first. For m users and n items in the recommendation system, the user set is represented by $U=\{u_1, \dots, u_m\}$, the set of items is represented by $P=\{p_1, \dots, p_n\}$, the scoring matrix is represented by R . The element r_{ij} in R is the score of the item p_j by the user u_i . If a user i did not score for item j , then the corresponding r_{ij} is empty. After that, the current user's non-graded item in the grading matrix is scored.

2.1 The basic slope one algorithm

The basic method of the algorithm is to use a simple regression expression $w=f(v)=v+b$, where v is the prediction score of the user u for the project, $R_{u,j}$ is the average deviation of item i from the item j score, denoted as $Dev_{i,j}$, U is defined as a user set, $S_{i,j}(U)$ is a set of users scoring item i and item j , $Num()$ is the number of elements contained in the set, then the formula of $Dev_{i,j}$ is

$$Dev_{i,j} = \sum_{u \in S_{i,j}(U)} \frac{R_{u,j} - R_{u,i}}{Num(S_{i,j}(U))} \quad (1)$$

Therefore, we can use $Dev_{i,j}+R_{u,j}$ to obtain the predictive score of item i from user u and get the predicted value after all possible predictions are averaged:

$$P(u)_i = \frac{\sum_{j \in R_i} (Dev_{i,j} + R_{u,j})}{Num(R_i)} \quad (2)$$

Where: R_i denotes that all users have given item sets whose scores satisfy the criteria ($i \neq j$ and $S_{i,j}$ are non-empty).

2.2 Improved Slope One algorithm

The current Slope One algorithm mainly includes Weighted Slope One algorithm and Bi-Polar Slope One algorithm.

Weighted Slope One algorithm takes into account the average number of different users $Dev_{j,i}$ credibility is different, the number of users as the weight, assuming that 100 users simultaneously rated the project j and i , while only 10 users at the same time scoring Projects j and k , then apparently obtained $Dev_{j,i}$ is more persuasive than $Dev_{j,k}$, and therefore weighted to the final predicted value. The weighted algorithm is

$$P^{wS}(u)_i = \frac{\sum_{i \in S(U) - \{i\}} (Dev_{i,j} + R_{u,j}) n_{i,j}}{\sum_{i \in S(U) - \{i\}} n_{i,j}} \quad (3)$$

Bi-Solar Slope One divides the scoring matrix into two categories according to "like" and "dislike". The dividing scheme determines whether the score of the user is greater than the average score of the user, "like" if the score of the item is greater than "like", and "dislike", see formula (4) and formula (5). When predicting for current user u , the predictor will focus only on those scores that the user has agreed on the positive or negative score.

$$S^{like}(u) = \{i \mid i \in S(u), r_{u,i} > \bar{r}_u\} \quad (4)$$

$$S^{dislike}(u) = \{i \mid i \in S(u), r_{u,i} < \bar{r}_u\} \quad (5)$$

among (4) and (5), $S(u)$ represents a set of items for which user u has a score, and $\bar{r}_u$ represents the average score of user u .

Finally, "like" and "dislike" are combined to get the comprehensive forecast score.

$$P^{bpSO}(u, j) = \frac{\sum_{i \in S^{like}(u) - \{j\}} P_{j,i}^{like} card(S_{j,i}^{like}) + \sum_{i \in S^{dislike}(u) - \{j\}} P_{j,i}^{dislike} card(S_{j,i}^{dislike})}{\sum_{i \in S^{like}(u) - \{j\}} card(S_{j,i}^{like}) + \sum_{i \in S^{dislike}(u) - \{j\}} card(S_{j,i}^{dislike})} \quad (6)$$

3. Slope One Method Of Weighting Based On User Similarity And Forgetting Function

Aiming at the existing problems of Slope One algorithm, this paper proposes a weighted Slope One algorithm (SAF) based on user similarity and forgetting function modification, that is, before the user item score is predicted, by removing extremely inactive users, To reduce the interference of the noise samples, and then calculate the similarity between users to get the user's nearest neighbor set, use the similarity between users as the weight of the score prediction, correct the item scores in the nearest neighbor set by the interest forgetting function, and finally, The nearest neighbor set performs a predictive analysis of the sample.

3.1 User activity rating

When the user similarity is calculated, the complexity of the algorithm is directly affected by the score item. All the users who remove the extremely inactive before calculating the similarity can reduce the time consumption and also can improve the prediction accuracy to a certain extent. In addition extremely inactive users score very little on the project, most likely new users, for new users because of the lack of data support, often recommend popular products will have better results. First set a threshold for activity, this article is defined as U_{inact} for users who score less than 0.01% of the number of users, and the user whose rating item is greater than or equal to 0.01%, defined as U_{act} , User Set $U = U_{inact} \cup U_{act}$. Define the total number of users for the project score $Num(item_u)$, the total number of items $Num(item)$, then

$$U^{inact} = \{item^u \in item \mid \frac{Num(item^u)}{Num(item)} < 0.01\% \} \quad (7)$$

$$U^{act} = \{item^u \in item \mid \frac{Num(item^u)}{Num(item)} \geq 0.01\% \} \quad (8)$$

3.2 User Similarity Measures

The similarity measure is very important to the recommendation system. The traditional similarity measure methods mainly include cosine similarity, modified cosine similarity and Pearson correlation coefficient.

Compared with the traditional similarity measure method, similarity algorithm based on Euclidean distance is simple and suitable for the case of scoring data sparseness, Euclidian similarity calculation is only needed when there is at least one user to score together, In the case of similarity, users who do not have a common score are considered as having similarities of 0. The Euclidean distance is

$$d(item_i, item_j) = \sqrt{\left(\sum_{i,j \in U} (item_i + item_j)^2\right)}$$

based on the Euclidean distance similarity is

$$\text{sim}(item_i, item_j) = 1/(1 + d(item_i, item_j)) \quad (9)$$

When selecting the similarity calculation method, the similarity of Euclidean distance only needs more than one common score between users, which is more suitable for sparse matrix, and its complexity is lower than the traditional similarity measure method. Therefore, Similarity calculation, this article uses formula (9) calculation.

3.3 User interest forgetting function

The famous German psychologist Ebbinghaus discovered a lot of experiments on the basis of human nature's forgetting laws: forgetting began immediately after memory, and the entire process of forgetting was uneven. Ebbinghaus plotted his experimental results as a curve, the famous Ebbinghaus forgetting curve, shown in Fig 1:

Fig 1. Ebbinghaus forgetting curve

Ebbinghaus forgetting curve mathematical function formula is as follows:

$$b = \frac{100k}{(\lg t)^c + k} \quad (10)$$

Where b is the memory retention (in percent), t is the time interval from memory awareness to the current time (in minutes), and c and k are two control constants. Ebbinghaus through the experiment repeatedly demonstrated that when $c=1.25, k=1.84$, the forgetting function and the most similar to the normal rule of forgetting.

User's remaining interest in the project in real time, that is, the remaining score. User interest forgetting function formula is as follows:

$$r'_{i,j} = r_{i,j} \bullet \frac{1.84}{(\lg t)^{1.25} + 1.84} \quad (11)$$

Where $r_{i,j}$ represents the score of user i on item j , $r'_{i,j}$ represents the remaining score of user i on item j after

decaying over time, and t represents the difference between the current time and the user's rating time (unit: minute).

3.4 Algorithm specific steps

1). Define user set $U_{act} = \{U_{act1}, U_{act2}, \dots, U_{acti}\}$, where $U_{acti} \subset U_{act}$, User $u \notin U_{acti}$ and $u \in U_{act}$. Set user activity threshold, through the formula (8) filter out the user set, used to calculate the user similarity.

2). The nearest neighbor set defining user u is $NU_n = \{n \in i \mid a \leq |\text{sim}(u, U_{actn})|\}$, the value of a is (0,1), and the similarity between users u and v is given by the formula (9), we define $\text{sim}_{u,v} = |\text{sim}(u, v)|$.

3). Calculate $r'_{i,j}$ from equation (11) based on the nearest neighbor set NU_n obtained in step 2, and stabilize when $r'_{i,j}$ is less than 20% of $r_{i,j}$. The fixed value is no longer calculated. In the nearest neighbor set NU_n , replace $r'_{i,j}$ with the original $r_{i,j}$ to get a new nearest neighbor set NNU_n .

4). Define $\text{sim}_{i,j} = \{j \in k \mid a \leq \text{sim}(item_i, item_j)\}$, use $\text{sim}_{i,j}$ as the weight, and use the formula (3) to calculate the predictive score.

$$P(u)_i = \frac{\sum_{v \in NNU_n} (Dev_{i,j} + R_{u,j}) \text{sim}_{i,j}}{\sum_{v \in NNU_n} \text{sim}_{i,j}} \quad (12)$$

4. Experimental Results And Analysis

4.1 Experimental data set

This paper uses MovieLens data set to evaluate the algorithm. The data for this dataset is the user's rating of the movie, from 1 to 5, respectively, the higher the score indicates that the user prefers the movie and records the time of the user's rating, at least 20 movies per user Scored. MovieLens is made up of three databases of different sizes. This article uses a small-scale library containing 100,000 scans of 1682 movies by 943 independent users. The data sparseness for this dataset is: $1 - 100000 / (943 * 1682) = 0.9369$.

The whole experiment divides the dataset into training and test datasets. Firstly, the dataset is divided into five equal parts randomly, among which four are selected as the training set and one is divided into the test set, and 200 of them are removed from the training set Record the history score and then join the test set as a forecast for new users.

4.2 Evaluation index

In this paper, the mean absolute error (MAE) is used as the metric, which is also the most commonly used recommended quality measure. MAE measures the

accuracy of the algorithm's recommended results by the degree of deviation between the project's forecast score and the user's actual rating. It needs to divide the dataset into two parts: the training set and the test set. The algorithm works on the training set and predicts the test set. MAE is calculated as follows:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N}$$

Where p_i is the forecast score of item i , q_i is the actual score of item i , and N is the predicted number of user-item scores in the test set. It can be seen from the formula that the smaller the MAE value, the higher the recommended quality of the algorithm.

4.3 Experimental results analysis

Experiment 1 In order to verify whether the accuracy of the proposed SAF weighted Slope One algorithm (SWSO) predictive score is more accurate than that based on user-based collaborative filtering (UBCF) and the project-based collaborative filtering (IBCF) algorithm, The data sets are respectively estimated by the algorithm SWSO, UBCF and IBCF, and then the MAE of the predictive score is respectively calculated. The similarity calculation adopts formula (5), the value of a is (0,1), the frequency is 0.1, the experimental results shown in Figure 2. a different value for the above three algorithms is not only the similarity of different, but also the number of different samples. For the algorithm IBCF, the value of a represents the value of the item similarity, and for the algorithm BWSO and UBCF, the value of a represents the value of the similarity between users. As can be seen from Fig 2, the prediction accuracy changes relatively gently until the value of a is less than or equal to 0.4. The MAE value of $a = 0.4$ is only 0.263% higher than that of $a = 0.1$, and the number of samples is significantly reduced. Therefore, considering the complexity of the algorithm and the prediction accuracy, a value of a is reasonable. It can also be seen that the prediction accuracy of the SAF weighted Slope One algorithm (SWSO algorithm) proposed in this paper is higher than the prediction accuracy of the UBCF and IBCF algorithms.

Fig 2. MAE Comparison of Slope One Algorithms at Different Thresholds

Experiment 2 In order to evaluate the algorithm better, we compared the proposed algorithm with other improved algorithms based on original Slope One, which are weighted Slope One algorithm, Bi-Polar Slope One algorithm, and the comparison between the interest forgetting function Corrections before and after correction. As can be seen from the experimental results in Fig. 3, the proposed algorithm can achieve smaller MAE than other original Slope One algorithms, and the magnitude of the improvement is also great. After adding the interest forgetting function, the MAE also has reduce. This shows that the proposed algorithm can greatly improve the quality of recommendation when the data is sparse by introducing user interest forgetting function to modify user similarity calculation to select and not score items.

Fig 3. MAE Comparison of Slope One Algorithm in Optimal Case

Experiment 3 In order to verify whether the prediction accuracy of the scoring data will be reduced after removing extremely inactive users from the sample data, we use weighted Slope One algorithm, Bi-Polar Slope One algorithm and SAF weighted Slope One algorithm (SWSO) of the samples and the user set U_{act} samples for predictive score. The sample data is extremely inactive users, the data sparsity is 0.8574, the sparseness is reduced by 7.970%. However, it is less than 0.2% that the MAE value measured by the user set U_{act} is less than the sample data set U by the user set U . Although removal of inactive users does not show much benefit in reducing MAE, computational efficiency is improved for the search of subsequent neighbors.

Table 1. MAE values for each algorithm under full sample and active sample

Algorithm	U sample	U_{act} sample
SAF weighted Slope One	0.7213	0.7081
Bi-Polar Slope One	0.7547	0.7352
weighted Slope One	0.7753	0.7445

5.Conclusion

In this paper, the current Slope One algorithm does not consider the similarity and lead to the low accuracy of personalized recommendation, by investigating the user similarity and user interest forgetting function to amend its similarity, and then according to the revised user similarity Determine the set of its neighbors, and use the similarity as a weight to improve the slope one algorithm. Three groups of experiments show that the removal of inactive user samples mainly improves the computational efficiency of the algorithm, selects the nearest neighbor by computing the similarity, and then modifies the nearest user's score by the interest forgetting function, which makes the sample data more streamlined and more significant Improve the accuracy of the score data prediction, but also greatly reduce the computational workload of the entire algorithm.

References

1. Xu H L,WU X,LI X D.Comparison Study of Internet Recommendation System[J]. Journal of Software, 2009, 20(2):350-362.
2. LIU J G,ZHOU T,WANG B H.Research Progress of Personalized Recommendation System[J].Progress in Natural Science,2009(1):1-15
3. POLATH,DUW.SVD-based collaborative filtering with privacy[C]//Proceedings of the 2005 ACM Symposium on Applied Computing .ACM,2005:791-795.
4. Li B,Zhu X Q,Li R J,et al.Cross-Domain collaborative filtering overtime[C] // Proc.of the 22nd Int'l Joint Conf.on Artificial Intelligence.IJCAI /AAAI Press, 2011.2292-2298.
5. MELVILLE,MOONEYRJ,NAGARAJANR.Content-tboosted collaborative filtering for improved recommendations[C]//AAAI/IAAI.Edmonton,Canda, 2002:187-192.
6. Han Min,Liu Xiaoxin.Feature selection techniques with class separability for multivariate time series[J]. Neurocomputing, 2013,110(1) : 29-34.