ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

A Recommendation Algorithm Uses Contribution Factor for Selecting Influential Neighbor

Yuan Xie [a)] and Lei Ding

School of Computer, Guangdong University of Technology, Guangdong 510006, China.

[a)] Corresponding author: 944738579@qq.com

Abstract. Collaborative filtering recommendation algorithm based on KNN neighbor selection does not consider the blind follower of neighbors when selecting neighbors, which causes some neighbor users to play a minor role in predicting the target user's scoring of unknown items. In response to this problem, a contribution factor is proposed. Jointly evaluate the item set from this perspective, consider the neighboring user's recommendation capability, calculate the neighboring user's recommendation contribution, combine the traditional user similarity to jointly select the neighbors, and recalculate the neighbor user's weight of the unknown project to improve the recommendation performance. Experimental results show that this improved algorithm improves recommendation accuracy.

Key words: KNN; contribution factor; neighbors; algorithm; recommendation performance.

INTRODUCTION

With the rapid development of the Internet and the skyrocketing amount of information, the traditional keyword-based search cannot meet the needs of users' personalized preferences. The recommendation system emerged at the historic moment to help users select items that meet user preferences from a large amount of data. Collaborative filtering is the most widely used and most successful recommendation algorithm in the proposed system. It can guess the user's preferences based on the user's historical behavior, and then make personalized recommendations for the user. However, with the explosive expansion of the Internet, data sparsity has become the most prominent problem in the recommendation system [1], and the accuracy of recommendation results has been reduced. Many scholars use matrix decomposition, padding, and cloud model calculations from the perspective of similarity calculations. Other methods to improve the accuracy of similarity calculation [2], and then improve the accuracy of the recommended results. There are also some scholars who consider from another point of view that the use of clustering, the introduction of weights, etc. make neighborhood choice more reasonable.

This paper starts with the main link of neighbor selection, based on the theory of leader nodes and follow-up nodes in social networks. It considers the role of neighboring users that have been evaluated but whose target users have not evaluated the recommendation. Calculate the contribution factor of neighboring users. According to the size of the factor, classify the follow-up size of neighboring users. Then combine the traditional similarity to calculate the contribution weight of the recommendation and make a more reasonable recommendation for the target user.

TRADITIONAL COLLABORATIVE FILTERING ALGORITHM

The basic flow of the traditional collaborative filtering recommendation algorithm is mainly divided into four steps, as follows:

1. According to the rating of n items in the system by m users, build a user-item scoring matrix(m×n)

2. Calculation of vector similarity between users, commonly used similarity calculations include cosine similarity, modified cosine similarity, and Pearson correlation similarity [3-4]. The specific formulas are as follows:

cosine similarity: The user's rating of items in the system can be seen as a multi-dimensional vector describing the user's interest features. This method measures the similarity between users by using the cosine angles between vectors. The calculation formula is shown in equation (1). Among them: Rui represents the rating value of the user u on the item i, Rvi represents the rating value of the user v on the item i, and Iuv represents the collection of the common rating items of the user u and the user v.

$$sim(u, v) = cos(u, v) = \frac{\sum_{i \in I_{uv}} R_{ui} \times R_{vi}}{\sqrt{\sum_{i \in I} R_{ui}^2} \times \sqrt{\sum_{i \in I} R_{vi}^2}} \tag{1}$$

Modified cosine similarity: the cosine similarity calculation does not take into account the difference in the user's rating scale whether the user prefers to score high or low. The modified cosine similarity improves the cosine similarity by subtracting the user's mean score of the item. Defects, the formula is shown in equation (2). Iu and Iv represent the collection of scoring items for user u and v, respectively. Other symbols are the same as above.

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} \left(R_{ui} - \overline{R_u}\right) \times \left(R_{vi} - \overline{R_v}\right)}{\sqrt{\sum_{i \in I_u} \left(R_{ui} - \overline{R_u}\right)^2} \times \sqrt{\sum_{i \in I_v} \left(R_{vi} - \overline{R_v}\right)^2}} \tag{2}$$

Pearson correlation similarity: this algorithm is different from the denominator scope of modified cosine similarity. When two user's scoring items are the same, these two algorithms are equivalent, and the symbolic meaning is the same as above, and the formula is shown in formula (3).

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} \left(R_{ui} - \overline{R_u}\right) \times \left(R_{vi} - \overline{R_v}\right)}{\sqrt{\sum_{i \in I_{uv}} \left(R_{ui} - \overline{R_u}\right)^2} \times \sqrt{\sum_{i \in I_{uv}} \left(R_{vi} - \overline{R_v}\right)^2}} \tag{3}$$

3.Choose the target based on similarity size, generally use KNN [5] as neighbor selection.

4. The target user is predicted to score the unknown item and provide a recommendation list for it. According to the weight of the similarity between the neighbor user and the target user calculated in the third step, the score for the unknown item is predicted in a weighted manner, and a formula can be used. As shown in equation (4). sim(a,j) represents the similarity between users a and j, n is the number of users in user a's neighbor group, R_j (i) represents the user j's rating of item i, and $\overline{R_J}$ represents user j's the average score of the items, $P_{a,j}$, represents the predicted score of item i for user a.

$$P_{a,i} = \overline{R_a} + \frac{\sum_{j=1}^{n} \left(R_j(i) - \overline{R_j}\right) \times sim(a, j)}{\sum_{j=1}^{n} sim(a, j)} \tag{4}$$

ALGORITHM DESIGN

Introduce the Concept of Contribution Factor

The amount of user B's contribution when making a recommendation to the target user A, that is, the amount of recommended information contained. It can be seen in the prediction that if the user B evaluates but the user A does not evaluate few items, the number of items is very large. Or, his role in the prediction is small. One extreme example is when user B evaluates a subset of user A that is a subset of user A. At this time, they have high similarity. However, when participating in the prediction, user B has no value. For this problem, the contribution factor of B is defined as (5) shows. The parameter φb is the ratio of the items evaluated by user B to the items in the system.

$$\chi_b = \frac{min(h, \varphi_b \times (n - a))}{n - a}, \chi \in [0,1] \tag{5}$$

$$\varphi_b = \frac{b}{n} \tag{6}$$

It can be seen that the larger the value of χ is, the more the number of items evaluated by B but not evaluated by A, and the wider the knowledge of neighboring user B, the more the new item can be recommended to the target user A. According to the calculated χ size, Neighbor users are classified. In the actual recommendation system, the number of items evaluated by the user is very small. According to the statistics, the average is only 3%, usually less than 1%.

Neighbor Selection and Recommendation to Join Contribution Factors

When the neighbor chooses, it uses the contribution factor χ and the similarity sim (i, j) weight to define the new recommendation contribution simc(i,j). If the user is blindly following the user, simc=0, if not, as (7) follows:

$$simc(i, j) = sim(i, j) \times \chi(i, j) \tag{7}$$

Using equation (7) to calculate simc(i,j) between the target user and other users, select k users with the highest degree of targeted user contribution as neighboring users. According to the neighbor user's rating of the project to the target user The recommended formula is shown in equation (8), where: simc(a,j) represents the weight of the recommended contribution between users a and j, and k is the number of users in the user group of user a, $R_j(i)$.) indicates that user j scores item i, $\overline{R_j}$ indicates the mean of user j's rating of the item, and $P_{a,j}$ indicate that user a's predicted score for item i.

$$P_{a,i} = \overline{R_a} + \frac{\sum_{j=1}^{k} \left(R_j(i) \times \overline{R_j}\right) \times simc(a, j)}{\sum_{j=1}^{k} simc(a, j)} \tag{8}$$

Collaborative Filtering Recommendation Algorithm Flow Based on Contribution Factor

Input: User item scoring matrix, target user A, index i.

Output: Provides a list of recommendations for target user A.

Calculate the similarity between target user A and other user B (index number j) according to equation (3) and user rating matrix sim(i,j)

Use equation (6) to calculate other user B parameters φj

Substituting parameter p into equation (5) to calculate the contribution factor χ(i,j) of other users B to target user A

Substituting sim(i,j) and χ(i,j) into equation (7) calculates user B's recommended contribution simc(i,j); according to the simc(i,j) sorting, selects the k nearest users to target user A’s set of neighbors Nk={n1,n2,…,nk},and use equation (8) to predict user A's rating of the item.

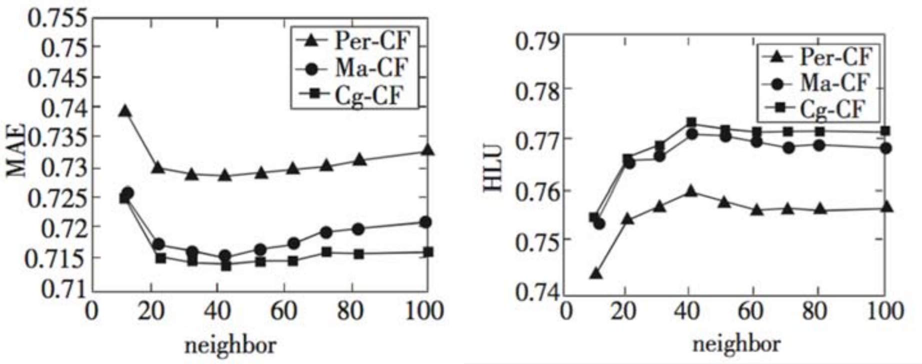
Sort p to provide target user A with a list of recommendations

EXPERIMENT

This experiment compares three algorithms, the traditional Person-rule-based Collaborative Filtering Recommendation Algorithm (Per-CF), the set threshold reduction similarity contingency recommendation algorithm (Ma-CF) proposed in [16], and the paper's Collaborative Filtering Recommendation Algorithm for Contribution Factor (Cg-CF) Experiments were conducted in a Java-based Eclipse development environment. To verify the validity of the algorithm in this paper, the Movielens-100k dataset provided by Group lens was used in the experiment. 100,000 rating records, 943 users rated 1682 movies, scored 1 to 5, sparsity was 94.3%. Through random 2/8 segmentation of the data set, 80% for the training data, 20% for the test data, conduct the simulation experiment of this algorithm. This paper evaluates the performance of the recommendation system from two aspects: 1) MAE, which is easy to understand, and can intuitively measure the recommended quality. It is the most commonly used recommended quality measurement method; 2) HLU, which reflects the level of the user's favorite items in the recommendation list provided by the recommendation system to the user.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|p_i - r_i\right| \tag{9}$$

$$HLU = \sum_{i \in V(u,K)} \frac{max(r_{ui} - d, 0)}{2^{\frac{(l_{ui} - d, 0)}{(h-1)}}} \tag{10}$$



(a) (b)

FIGURE 1. (a) Comparison of the recommended accuracy of the three algorithms;(b) Comparison of the HLU of the three algorithms

The MAE of the Cg-CF algorithm in this paper has been very small, which shows that the algorithm in this paper does improve the recommendation accuracy compared to the traditional recommendation algorithm. The HLU of Cg-CF has been kept high. It indicates that the user's favorite items in the user-supplied list are more reasonably sorted and the possibility of successful recommendation is greater, which also reflects the accuracy of the algorithm's recommendation.

CONCLUSION

This paper studies the follow-up characteristics of neighboring users and proposes a collaborative filtering recommendation algorithm based on the contribution factor. This algorithm proposes the user's contribution factor through the weighted operation of the items that have been evaluated by neighbor users but not evaluated by the target users. This factor can effectively kill neighboring users who blindly follow the trend and make neighbor users contribute more when predicting the target user's rating of unknown items. Finally, simulation experiments are performed to compare with other two algorithms, which proves that the algorithm does indeed make neighbor selection more effective. Rationalization improves the accuracy of recommendations.

ACKNOWLEDGMENTS

This work is supported by the national natural science foundation of China under grant (No. 61202267), natural science foundation of Guangdong province under grant (No.2016A030313713).

REFERENCES

1. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE transactions on knowledge and data engineering, 2005, 17(6): 734-749.
2. Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook[M]//Recommender systems handbook. springer US, 2011: 1-35.
3. Kohrs A, Merialdo B. Creating user-adapted Websites by the use of collaborative filtering[J]. Interacting with Computers, 2001, 13(6): 695-716.
4. Sarwar B M. Sparsity, scalability, and distribution in recommender systems[M]. University of Minnesota, 2001.
5. Chen Muchen, Chen Longsheng, Hsu F H, et al. A profitability-based recommender system [C]//Proc of IEEE International Conference on Industrial Engineering Management.2007:219-223.