Weighted Slope One Algorithm with Integrated User Trust Factor

Mei Yangyang[1, a, *], Xiao Zhenghong[1, b], Ouyang Jia[1, c], Yan Yiting [1, d], and Xu Shengdong[1, e]

[1]*College of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

[a] *843558246@qq.com,* [b] *750735160@qq.com,* [c] *ouyangjia1@163.com,* [d] *779084229@qq.com,* [e] *949998604@qq.com*

Keywords: collaborative filtering, Slope One, trust factor, user similarity

Abstract: In view of the low accuracy of the Slope One personalized recommendation algorithm because of ignoring user trust and project similarity, a weighted Slope One algorithm that integrates the user trust factor is proposed in this work. This study considers the proportion of users' common-score items to the number of items scored by the target users, develops user trust factor model and algorithms, uses the Pearson correlation coefficient to calculate user similarity, introduces the trust factor to modify user similarity and obtain the target users' top-K nearest neighbor sets, and uses a modified weighted Slope One algorithm for the predictive analysis of a sample. Experiments are conducted using the MovieLens data set. Results show that the proposed method improves the accuracy of prediction and effectively improves recommendation accuracy.

1. Introduction

With the rapid development of e-commerce, the numbers of product categories and users in large-scale e-commerce systems have increased dramatically. The number of users is often much higher than that of products, but the users' rated products generally do not exceed 1% of the total number of products [2], so the user–item rating matrix is extremely sparse. Data sparsity results in the low efficiency of recommendation algorithms and inaccurate recommendations, which are typical problems currently faced by CF algorithms [3]. In real life, two people who have similar interests may have varying trust in an item; consequently, their acceptance of a recommendation will differ. In other words, besides similarity, trust is an important factor influencing a person's decision-making process. The effective integration of user trust relationships into personalized recommendations is essential to improving the quality of recommendations. In fact, the effective integration of user trust relationships into personalized recommendations has become a hot topic in the field of recommendation systems [4], and many methods can be used for reference.

In 2016, Lu et al. proposed an implied trust-aware CF algorithm that can realize accurate personalized recommendation by mining potential trust relationships, such as user preferences and activities [5]. Li et al. proposed a CF recommendation algorithm combined with user trust; the algorithm can effectively improve the accuracy of recommendation by combining the rating trust and preference trust between users [6]. A CF algorithm based on a trust factor was proposed by Guo et al.,

who established a trust model on the basis of the number of users evaluated and the number of times recommended for others [7]. Although these methods improve recommendation accuracy to a certain extent, they cannot overcome problems well in real-time systems of massive data, and the time complexity is high. To solve problems in real time, Lemire and Maclachlan proposed the Slope One algorithm in 2005[8]. This algorithm is thus far the most concise form of CF algorithm that is based on item evaluation. It has the advantages of easy implementation, high efficiency, good expansibility, and low algorithmic complexity. However, it considers neither the similarity and the mutual trust relationship between users nor the possible internal relationship between items. It considers only the average deviation of items, thereby resulting in low recommendation accuracy. To solve these problems, a weighted Slope One algorithm with integrated user trust factor is proposed in this study. The proposed algorithm considers the proportion of the number of users' common-score items to the number of items scored by the target users, designs the user trust factor model, calculates the user similarity by using the Pearson correlation coefficient, and introduces the trust factor to modify the user similarity. The top-K nearest neighbor set of the target user is obtained, and the improved weighted Slope One algorithm is used to predict and analyze the sample to improve the accuracy of prediction.

2. Related Work

2.1 Introduction to the Slope One Algorithm

The Slope One algorithm considers that a linear relationship exists between the user rating and the item and uses the linear regression method to predict the score. The prediction formula is expressed as $w = f(v) = v + b$, where the parameter v is the historical score generated by the target user, and the parameter b is the average difference between the different items' ratings. For the user–item rating matrix, the mean deviation for the different items i and j is defined as follows:

$$dev_{ij} = \frac{1}{card(U_{ij})} \sum_{u \in U_{ij}} \left(r_{ui} - r_{uj} \right) \tag{1}$$

Where dev_{ij} represents the average deviation of items i and j in the rating matrix; U_{ij} represents the set of users who have scored items i and j; r_{ui} represents the rating of item i by user u; and r_{uj} represents the rating of item j by user u; and $card(U_{ij})$ represents the number of users in the set U_{ij}.

The Slope One algorithm uses $r_{vi} - dev_{ij}$ to predict the score of item j by user v. In general, a user may have more than one rated item; thus, all the forecasts are averaged. The final prediction value can be obtained as follows:

$$p_{vj} = \frac{1}{card(R_j)} \sum_{i \in R_j} \left(r_{vi} - dev_{ij} \right) \tag{2}$$

Where p_{vj} denotes the prediction score of user v for unrated item j; r_{vi} denotes the rating of item i by user v; dev_{ij} denotes the average score deviation of items i and j; R_j denotes the user's set of

v rated items; and $card(R_j)$ denotes the number of users in the set R_j.

2.2 Weighted Slope One Algorithm

The Slope One algorithm does not consider the number of items scored by the user; consequently, the more ratings are available, the more accurate the prediction will be. For example, 1000 users rate items i and j, and only 10 users rate items i and k. Thus, the average score deviation dev_{ij} is more convincing than dev_{ik}. To address this issue, the weighted Slope One algorithm was proposed in Reference [8]. This algorithm uses the following prediction formula (3):

$$p_{vj} = \frac{\sum_{i \in R_j} (r_{vi} - dev_{ij}) \cdot c_{ij}}{\sum_{i \in R_j} c_{ij}} \tag{3}$$

Where c_{ij} is the weight, and $c_{ij} = card(R_{ij})$ is the number of users who jointly evaluate items i and j $(i \neq j)$.

2.3 Similarity Measure

Many measures have been used to determine similarity, including Pearson's correlation coefficient, Euclidean distance, and cosine similarity. Considering the complexity of the algorithm and the size and characteristics of the data, the present study uses the Pearson correlation coefficient and cosine similarity to design a similarity measure model.

2.3.1 Pearson Correlation Coefficient

The Pearson correlation coefficient is a basic measure for calculating the similarity of vectors; the linear correlation between two involved vectors is computed to measure the degree of similarity between the two[9]. The Pearson correlation coefficient between two users is defined as follows:

$$PCC\_sim(u,v) = \frac{\sum_{i \in T_{uv}} (r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in T_{uv}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{i \in T_{uv}} (r_{vi} - \overline{r_v})^2}} \tag{4}$$

Where $PCC\_sim(u,v)$ represents the Pearson correlation coefficient between users u and v, and its range of values is [−1, 1]. The closer the value is to 1, the higher the similarity; conversely, the closer the value is to −1, the lower the similarity. In formula (4), T_{uv} represents a set of items that users u and v score together; $\overline{r_u}$ represents the average of the item scores or ratings of user u; and $\overline{r_v}$ represents the average of the item scores of user v.

2.3.2 Cosine Similarity

The cosine similarity is based on the user's score vector. The cosine of the included angle between the two vectors is calculated to measure the degree of similarity between the two vectors. The cosine similarity between two users is defined as follows:

$$COS\_sim(u,v)=\frac{u\times v}{\|u\|\times\|v\|}=\frac{\sum_{i\in T_{uv}} r_{ui}\times r_{vi}}{\sqrt{\sum_{i\in T_{uv}} r_{ui}^{\ 2}}\sqrt{\sum_{i\in T_{uv}} r_{vi}^{\ 2}}} \tag{5}$$

However, both similarity measures have several drawbacks.

(1) Misjudgment may easily occur when two users jointly score items that are small and close;

(2) These traditional metrics measure similarity between two users. This process is inadequate and should be differentiated;

(3) Trust is also an important factor influencing a person's decision-making process and should thus be reflected in the measurement method.

3. Weighted Slope One Algorithm with Integrated User Trust Factor

To solve the three problems mentioned in the previous section, this study improves the weighted Slope One algorithm with use of the Pearson correlation coefficient, designs an improved model of the user trust factor, and proposes a weighted Slope One algorithm with integrated user trust factor.

3.1 Trust Factor

The traditional user trust relationship is equal, that is, user u trusts user v, and user v also trusts user u. However, in real life, user u has a high degree of trust to user v, and often, user v also trusts user u; but user v does not necessarily have a high trust degree in user u. As shown in Table 1, for user v, user u has the same rating information for items 2 and 5 (I_2 and I_5 in the table, respectively), and user v can be considered to have a high degree of trust in user u, but thinking that user u has a high degree of trust in user v is not necessarily reasonable. The fact that users u and user w rate similarly means they can be regarded as having the same level of trust.

Table 1 User–item rating matrix

	I_1	I_2	I_4	I_5	I_6
u	2	3		3	2
v		3		3	
w	2	3		3	2

In this study, for the two rating users u and v, the trust degree of user u in user v is measured on the basis of the proportion of their common-score items to the number of items rated by the target users. This degree of trust is called the trust factor and has the range [0, 1]. The trust factor is defined as follows:

$$w(u,v)=1-\exp\left(-\frac{|T_u\cap T_v|}{|T_u|}\right) \tag{6}$$

Where $w(u,v)$ indicates the degree of trust that user u has in user v; T_u and T_v represent the number of items evaluated by users u and v, respectively.

The number of users' common-score items is one of the important variables affecting the trust between users. The larger the number of the users' common-score items, the higher the degree of trust between the users. Similarly, the participation of users is also an important factor that affects the degree of trust between users. The higher the participation of a user is, the more easily he/she can

obtain the trust of other users. Therefore, in this study, the number of users' common-score items is taken into account in the proportion of the two users' scored items to improve the model for user trust factor, which is defined as follows:

$$asy_w(u,v)=\left(1-\exp\left(-\frac{|T_u\cap T_v|}{|T_u|}\right)\right)\cdot\frac{2\cdot|T_u\cap T_v|}{|T_u|+|T_v|} \tag{7}$$

The improved user trust factor is introduced to modify the traditional user similarity measures.

1) Pearson correlation coefficient after incorporating the user trust factor

$$\begin{aligned} AWPCC\_sim(u,v)&=PCC\_sim(u,v)\cdot asy_w(u,v)\\ &=\frac{\sum_{i\in T_{uv}}(r_{ui}-\overline{r}_u)(r_{vi}-\overline{r}_v)}{\sqrt{\sum_{i\in T_{uv}}(r_{ui}-\overline{r}_u)^2}\sqrt{\sum_{i\in T_{uv}}(r_{vi}-\overline{r}_v)^2}}\cdot\left(1-\exp\left(-\frac{|T_u\cap T_v|}{|T_u|}\right)\right)\cdot\frac{2\cdot|T_u\cap T_v|}{|T_u|+|T_v|} \end{aligned} \tag{8}$$

2) Cosine similarity after incorporating the user trust factor

$$\begin{aligned} AWCOS\_sim(u,v)&=COS\_sim(u,v)\cdot asy_w(u,v)\\ &=\frac{\sum_{i\in T_{uv}}r_{ui}\times r_{vi}}{\sqrt{\sum_{i\in T_{uv}}r_{ui}^{\,2}}\sqrt{\sum_{i\in T_{uv}}r_{vi}^{\,2}}}\cdot\left(1-\exp\left(-\frac{|T_u\cap T_v|}{|T_u|}\right)\right)\cdot\frac{2\cdot|T_u\cap T_v|}{|T_u|+|T_v|} \end{aligned} \tag{9}$$

3.2 Description of Weighted Slope One Algorithm with Integrated User Trust Factor

In this study, methods based on the similarity threshold and K value are used to validate the similarity measures with the user trust factor and the prediction performance of the proposed algorithm. The traditional threshold-based method is insufficiently flexible in repeatedly adjusting the threshold value of similarity when the nearest neighbor is selected. A dynamic threshold optimization scheme comprising two algorithms is proposed to solve this problem. The scheme dynamically calculates the average similarity of all the users whose similarity is greater than 0 in the nearest neighbor set of the target users (K-value algorithm) and selects the mean as the threshold (dynamic threshold algorithm). The algorithms are as follows.

Input: User–item rating matrix $R_{m\times n}$, target user u, target item i

Output: Predicted score p_{ui} of target user u for item i

3.2.1 K-value Algorithm

Step 1. Initialize the user–item rating matrix, target user, and target item.

Step 2. If target user u has evaluated at least one item and evaluated item j with other users, then set the number of nearest neighbor users K, calculate the similarity $AWPCC\_sim(u,v)$ between target users u and v using formula (8), and perform reverse order processing with the first K as the nearest neighbor user set $T_{sim(u,v)}$. Otherwise, output user u evaluates the average value of the item score and ends.

Step 3. Use formula (3) to calculate the predicted score p_{ui} of target user u for item i.

Step 4. Repeat Step 2, with K assuming the values of 5, 10, 20, 40, 80, and 160 chronologically.

Step 5. End.

3.2.2 Dynamic Threshold Algorithm

Step 1. Initialize the user–item rating matrix, target user, and target item.

Step 2. If target user u has evaluated at least one item and evaluated item j with other users, then use formula (8) to calculate the average value $\overline{sim(u,v)}$ of all the users whose similarity with target user u is higher than 0.

Step 3. Calculate the similarity $AWPCC\_sim(u,v)$ between target users u and user v by using Formula (8), taking the users whose similarity is not less than $\overline{sim(u,v)}$ as the nearest neighbor user set $T_{sim(u,v)}$. If the neighbor user set $T_{sim(u,v)}$ is an empty set, then output user u evaluates the average of the project score and ends.

Step 4. Use Formula (3) to calculate the predicted score p_{ui} of target user u for item i.

Step 5. End.

4. Test and Analysis

4.1 Data Set

In this study, the common data set MovieLens, which contains 100000 rating records by 943 users for 1682 movie items, with each user having rated more than 20 movie items. The rating range is 1–5, with 1 meaning “very poor” and 5 meaning “very good.” All the algorithms are tested five times to reduce the effect of randomness on the experimental results. At each time, 80% is selected as the training set, whereas the remaining 20% is assigned to the test set. The final experimental results are the average values of the five tests.

4.2 Evaluation Index

Statistical accuracy is used to evaluate the advantages and disadvantages of each algorithm. Two indices are commonly used to indicate statistical accuracy: mean absolute error (MAE)[10] and root mean square error (RMSE). MAE is the average of the deviation between the single real value and the predicted value, and the RMSE reflects the degree of deviation of the predicted data from the real value. The smaller the MAE and RMSE, the higher the prediction accuracy. These indices are calculated as follows:

$$MAE = \frac{\Sigma_{u,i\in T}\left|r_{ui} - pre_{ui}\right|}{\left|T\right|} \tag{10}$$

$$RMSE = \sqrt{\frac{\sum_{u,i\in T}(r_{ui} - pre_{ui})^2}{T}} \tag{11}$$

4.3 Experimental Results and Analysis

For this study, two groups of experiments are designed in the Python 3.6 environment using the tool PyCharm. The proposed algorithm is verified using the MovieLens data set.

Experiment 1. For improved experimental results, the following algorithms are based on the *K*-value algorithm:(1) Weighted Slope One algorithm based on Pearson correlation coefficient (PCC Slope One);(2) Weighted Slope One algorithm based on Pearson correlation coefficient with the user trust factor (AWPCC Slope One);(3) Weighted Slope One algorithm based on cosine similarity with

user trust factor (AWCOS Slope One). By contrast, the dynamic threshold algorithm is used in the weighted Slope One algorithm that is based on the Pearson correlation coefficient with the user trust factor (DTPCC Slope One). The numbers of users' neighbors (*K*) are 5, 10, 20, 40, 80, and 160. After the predicted values are obtained, the MAE (Fig. 1) and RMSE (Fig. 2) are calculated and compared.

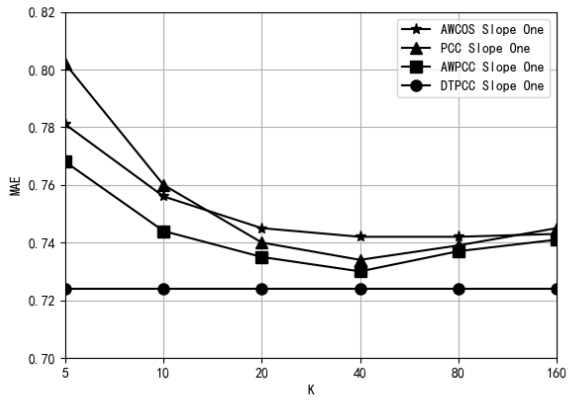

Fig. 1 MAE-K

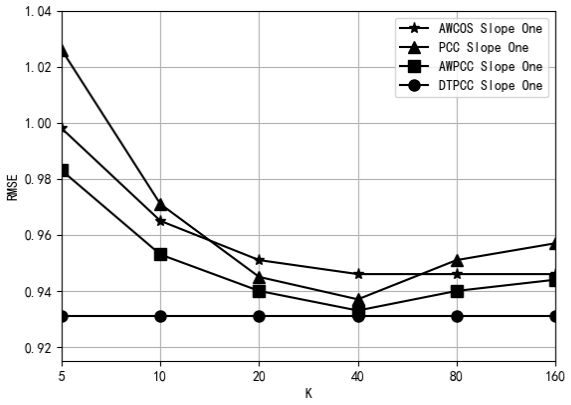

Fig. 2 RMSE-K

As shown in Figs. 1 and 2, with an increase in the number of nearest neighbors (*K*), the prediction results initially increase and then stabilize after *K*=40. In view of the sparse data and the distribution of the user's neighborhood, *K* increases to a certain extent, and the accuracy tends to stabilize or decrease. Apparently, the performances of the AWPCC Slope One and DTPCC Slope One algorithms are improved, indicating that the introduction of the user trust factor has a significant effect on the improvement of prediction accuracy. The DTPCC Slope One algorithm has the best performance, and the AWPCC Slope One algorithm is better than the PCC Slope One and AWCOS Slope One algorithms. On the basis of the results in Experiment 1, the DTPCC Slope One algorithm and the AWPCC Slope One algorithm are selected for Experiment 2.

Experiment 2. The two selected versions of the weighted Slope One algorithm with integrated user trust factor are compared with the following existing Slope One algorithms to evaluate the prediction performance of the proposed algorithm:

(1) Integrating User Similarity and Item Similarity into Weighted Slope One Algorithm[11] (Algorithm 1);

(2) Integrating Item Relevance into Weighted Slope One Algorithm[12] (Algorithm 2)

The comparison of the prediction accuracy levels of the four algorithms is shown in Figs. 3 and 4.

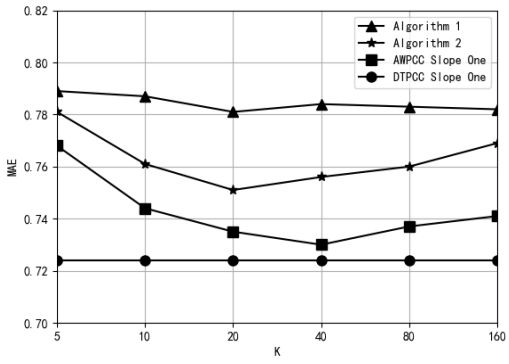

Fig. 3 MAE-K

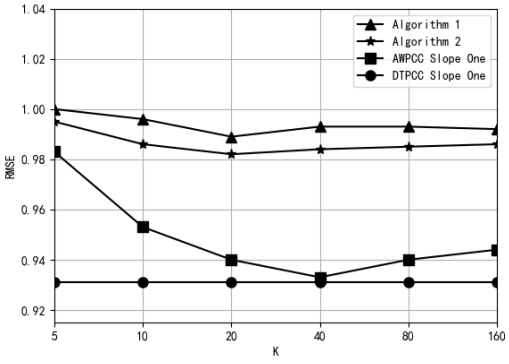

Fig. 4 RMSE-K

As shown in Figs. 3 and 4, the two versions of the weighted Slope One algorithm with integrated user trust factor are superior to the Algorithm 1 and the Algorithm 2. In particular, the DTPCC Slope One algorithm proposed in this work is highly superior to the three other algorithms. Nevertheless, the AWPCC Slope One algorithm still performs better than do the two other algorithms, obtaining an MAE of only 0.734 when K=40. The MAE of the DTPCC Slope One algorithm is 0.724, which is the lowest among the MAE of all compared algorithms. For a 5-point evaluation system, MAE=0.73 is generally a remarkable score that cannot be easily surpassed[13].

5. Conclusion

To address the problems of the Slope One algorithm, this work proposes a weighted Slope One algorithm integrated with a trust factor model. In this study, a user trust factor model is designed, the Pearson correlation coefficient is used to calculate user similarity, the trust factor is introduced to modify the user similarity, and the established weighted Slope One algorithm is used to predict and analyze a sample through experiments. The experimental results show that the improved trust factor algorithm is feasible, and the recommendation quality of this algorithm is remarkably improved in the case of sparse data. In the future, we will integrate the proposed algorithm with machine learning algorithms, incorporate the concept of artificial intelligence, and further improve the algorithm to obtain increasingly accurate and efficient recommendation performance.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China under Grant No.61702119, Supported by the Guangdong Provincial Science and Technology Program (No. 2016A01010101029), Supported by the Guangzhou Science and Technology Program (No. 201607010152), and Supported by Young Creative Talents Project of Guangdong Provincial Education Department (Natural Science) under Grant No.57/572020507

References

[1] Xu Hai-ling, Wu Xiao, Li Xiao-dong, et al. Cnnic L O, Beijing, Beijing. Comparison Study of Internet Recommendation System [J]. Journal of Software, 2009, 20(2):350-362.

[2] Castro-Schez J J, Miguel R, Vallejo D, et al. A highly adaptive recommender system based on fuzzy logic for B2C e-commerce portals [J]. Expert Systems with Applications An International Journal, 2011, 38(3):2441-2454.

[3] Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering[J]. Acm Transactions on Information Systems, 2004, 22(1): 116-142.

[4] Zhang Fu-guo,Xu Sheng-hua. Research on recommendation diversification in trust based e-commerce recommender systems[J]. Journal of the China Society for Scientific & Technical Information, 2010, 29(2): 350-355.

[5] Kun LU, Xie L, Ming-Chu LI, et al. Research on Implied-trust Aware Collaborative Filtering Recommendation Algorithm [J]. Journal of Chinese Computer Systems, 2016.

[6] Li Liang, Dong Yuxin, Zhao Chunhui, Cheng Weijie. Collaborative Filtering Recommendation Algorithm Combined with User Trust[J].Journal of Chinese Computer Systems,2017,38(05):951-955.

[7] Guo Yanhong, Deng Guishi, Yu Chunyu. Collaborative Filtering Recommendation Algorithm Based on Factor of Trust [J].Computer Engineering,2008(20):1-3.

[8] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering [J]. Computer Science, 2007:21--23.

[9] Guo Hao. Research on Weighted Slope One Algorithm Based on Likelihood Ration Similarity and Genre Correlation [D]. Liaoning University, 2017.

[10] Park Y J, Tuzhilin A. The long tail of recommender systems and how to leverage it[C] ACM Conference on Recommend System. 2008:11-18.

[11] Zhang Yulian, Pei Sisi, Liang Shunpan. Integrating User Similarity and Item Similarity into Weighted Slope One Algorithm[J].Journal of Chinese Computer Systems, 2016, 37(06): 1174-1178.

[12] Feng Yong, Xu Hongyan, Wang Yubing, Guo Hao. Research on Weighted Slope One Algorithm Incorporating Item Relevance[J/OL].Computer Science and Exploration:1-10.
[13] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluation Collaborative Filtering Recommender Systems [J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53.