Improved Slope One Algorithm Using Multi-weight and Auxiliary Items

Jiaxin Han, Yang Yu
School of Computer Science, Xi’an Shinyou University Shaanxi, Xi’an 710065, China
yuyangxsys@gmail.com

Abstract. Slope one algorithm is widely used in recommendation systems. Although slope one algorithm is simple and efficient, the correlation between items and data sparsity issue are still the main issues. This paper proposes a multi-weight slope one algorithm, which obtains the correlation between items from multiple aspects, and introduces auxiliary items in relatively sparse scoring data to improve the recommendation effect. Experimental results on the MovieLens show that the recommended effect of multiple weight is better than the single weight. In the case of the sparseness of the data, the MAE value is reducing by 3% when using auxiliary items.

1. Introduction
The large amount of increase in information available over the Internet has created a greatest challenge in searching useful information[1]. The recommendation system emerges as a solution based on collective intelligence for recommending favorite projects to users. The slope one algorithm is an item-based collaborative filtering proposed by Lemire[2]. Unlike traditional project-based collaborative filtering, the slope one algorithm predicts the rating of an item by using the difference of the items and using the user's existing ratings. Compared with the collaborative filtering algorithm, slope one algorithm has the advantages of simplicity, high-speed, and efficiency[3].

Although slope one algorithm has many advantages, the correlation between items and data sparsity issue are still the main issues. This paper proposes a multi-weight slope one algorithm. The correlation between items is obtained from the scoring data, the preference for the items of users and other information. Facing data sparsity issue, auxiliary items are added into sparse scoring data to improve the recommendation effect, that are obtained from the scoring data and the attribute information of items.

2. Related Work
At present, the improvements of slope one algorithm are mainly concentrated on two aspects. The first is to improve the similarities between items. The second is to solve the problem of sparsity data.

Improvements in similarity mainly use a variety of information to improve the degree of correlation between items. Jiang[4] proposed an algorithm, which could assign different weights for items at different time. Li[5] considered item characteristics and common item information and proposed that the user’s comprehensive rating to the item was predicted mainly by experts and item characteristics. Chen and Zheng[6] proposed a recommendation algorithm combining SVD and slope one. They use the similarities with projects rather than the number of items that are commonly scored to weight the deviations from items.
Sparsity is a problem occurring frequently in recommender systems when many users have provided ratings to a limited number of items, or many items have received only a few ratings [7]. Lin [8] proposed a collaborative filtering recommendation algorithm combining k-means and slope one. The method was simple and easy to implement, and tried to solve the data sparsity problem. However, it is not scalable. Saeed [9] proposed a slope one algorithm to solve sparsity problem. The algorithm proposes introducing virtual items in the case of sparse data to improve the sparse scoring matrix. The influence of attributes of items is not considered in Virtual items and the process of generating scores of virtual items is too simple, ignoring the impact on the users.

3. Slope One Algorithm Improvement

3.1. Basic Slope One Algorithm

Slope one algorithm uses a linear regression method of scoring prediction. The linear regression formula is expressed as \( f(x) = x + b \), where \( x \) is the historical score scored by the target user in the recommendation system, and \( b \) is average score deviation between the items. For the different items \( i \) and \( j \) in the scoring matrix, their average score deviation \( \text{dev}_{ij} \) as follows:

\[
dev_{ij} = \frac{\sum_{u \in U_{ij}} (r_{ui} - r_{uj})}{|U_{ij}|}
\]

Where \( r_{ui} \) represents the rating of user \( u \) to item \( i \), and \( U_{ij} \) represents the set of users that scored both items \( i \) and \( j \), \(|U_{ij}|\) indicates the number of users in the set \( U_{ij} \).

After using the formula to obtain the average score deviation from items, the slope one algorithm is used to predict the score. slope one algorithm does not consider the number of users that have scored items to adjust weight of the scoring. Lemire proposed a weighted slope one algorithm. The calculation formula is as follows:

\[
P_{ui} = \frac{\sum_{i \in I_{ui}} (r_{ui} - \text{dev}_{ij}) |U_{ij}|}{\sum_{i \in I_{ui}} |U_{ij}|}
\]

3.2. Multi-weight Slope One Algorithm

The basic slope one uses the number of items that the user scored together as the weight of the user's rating, without considering the effect of similarity between the items on the weight of score. Using the Pearson coefficient to measure the difference between items, which is expressed as follows:

\[
sim_{R}(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_{i}) (r_{uj} - \bar{r}_{j})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_{i})^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_{j})^2}}
\]

Where \( \bar{r}_{i} \) represents the average rating of users who scored on item \( i \)

Considering that each user has different degrees of loving for different project attributes, we combine the attributes of the project to get the user's preference for the project. Expressed as follows:

\[
sim_{m}(i, j) = \frac{\sum_{p \in A_{i}} \sum_{a \in A_{p}} |\mu_{p}|}{\sum_{a \in A_{p}} |\mu_{p}|}
\]

Where \( A_{i} \) represents a collection of all attributes of item \( i \), \( \mu_{p} \) represents item \( \mu \) containing attribute \( p \).

The number of people share the project \( i \) and the project \( j \) had an impact on the similarity with the projects. Combining all factors, the comprehensive similarity between item \( i \) and \( j \) are expressed as \( \text{sim}(i, j) \)

\[
\text{sim}(i, j) = \text{sim}_{R}(i, j) \cdot \text{sim}_{m}(i, j) \cdot |U_{ij}|
\]

After getting the similarity, we get the user's rating of the item according to the weighted slope one algorithm

3.3. Multi-weight Slope One Algorithm with Auxiliary Items

When the number of users' scored items is much smaller than the total number of items, the quality of recommendation is degraded, because the fewer effective items are obtained by the slope one
algorithm and the predicted results are greatly affected by the scored items, resulting in a decrease in the recommended quality. In order to improve this situation, this paper proposes to use the auxiliary items to augment the user's scoring matrix. When the items scored by user are less than the threshold \( T \), the auxiliary information is generated by the user's information. The similarity of the auxiliary item is measured by the improved cosine similarity, which is expressed as:

\[
sim_{s\text{a}}(i,j) = \frac{\sum_{u \in U_{ij}} p_{fu}(r_u - \mu_u)(r_{u,j} - \mu_{u,j})}{\sqrt{\sum_{u \in U_{ij}} (r_u - \mu_u)^2} \cdot \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \mu_{u,j})^2}}
\]

\( p_{fu} \) is the weight of user \( u \). The greater the number of user's scoring items, the less impact user's ratings are

\[
p_{fu} = \log \frac{I}{I_u + 1}
\]

Where \( I \) is the number of items.

The properties of item have an impact on the similarity with items, using the following formula to represent

\[
sim_{s\text{b}}(i,j) = \frac{|A_i \cap A_j| + \alpha}{|A_i \cup A_j| + N \alpha}
\]

\( \alpha \) is Laplace smoothing parameter, \( N \) is the maximum value of the union set of the properties of item \( i \) and \( j \).

The number of users that had scored auxiliary item and predicted item, and the numbers of users that had scored auxiliary items and some items that user \( u \) scored, have an impact on the similarity.

\[
Count_{i,j} = U_{ij} \cdot (1 - rate) + C_{u,i} \cdot rate
\]

\[
C_{u,i} = U_{\theta \gamma} + t
\]

Where \( rate \) represents the proportion of auxiliary items in all items, \( C_{u,i} \) represents the number of users who have scored \( i \) and items that user \( u \) scored, \( \theta \) represents the proportion of the scored items.

The weight of the auxiliary is expressed as:

\[
sim_{s}(i,j) = \frac{\sum_{u \in U_{ij}} p_{fu}(r_u - \mu_u)(r_{u,j} - \mu_{u,j})}{\sqrt{\sum_{u \in U_{ij}} (r_u - \mu_u)^2} \cdot \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \mu_{u,j})^2}} \cdot \frac{|A_i \cap A_j| + \alpha}{|A_i \cup A_j| + N \alpha} \cdot Count_{i,j}
\]

The score of the auxiliary item is combined the user’s rating and other users’ ratings to the auxiliary item, as follows:

\[
r_{s_{u,i,j}} = \frac{r_u, C_{u,i} + r_{s,i} U_{ij}}{C_{u,i} + U_{ij}} + c
\]

Where \( c \) is an adjustable constant.

Through multiple weights and auxiliary items, we organize the improved algorithm. The following is the flow of the entire algorithm.

**Algorithm 1: Prediction \((u, j, T, \alpha, \theta, c)\)**

**Inputs:** \( u \) : active user; \( j \) : active item; \( T \) : threshold is used to control auxiliary items; \( \alpha \) : Laplace smoothing parameter; \( \theta \) : the proportion of the scoring item; \( c \) : the parameter of \( r_s \)

**Outputs:** \( P_{u,j} \) : predicted rating of user \( u \) to item \( j \)

1. let Predicate_items = \{ \( i \in I_u \mid |U_{i,j}| > 0 \) \}
2. let \( S = {} \)
3. for each \( i \) in Predicate_items:
   4.   add \((i, r_{u,i}, dev_{i,j}, sim(i,j))\) to \( S \)
5. end for
6. let \( k = T - |\text{Predicate_items}| \)
7. if \( k > 0 \) then
   8.   let Items = \{ \( i \notin I_u \}, S1 = {} \)
9.   for each \( i \) in Items:
   10.      add \((i, r_{s_{u,i,j}}, dev_{i,j}, sim_{s}(i,j))\) to \( S1 \)
11. end for
12. sort \( S_1 \) in descending order of \( sim_s(i, j) \)
13. add first \( k \) items of \( S_1 \) to \( S \)
14. end if
15. let \( P_{u,j} = \text{Predicted rating by applying Weighted Slope One On } S \)
16. return \( P_{u,j} \)

4. Experiments

The MovieLens dataset is used in this paper, which provides a ml-100k dataset for users to score movies at different time periods. The dataset records the historical scoring information of 943 users on 1682 films, providing movie attributes such as movie name, movie type, release time, and so on. The data sparsity is 93.7%. The project attributes selected in this paper are movie types. There are 18 types of movie types, such as sci-fi, adventure, action, comedy and so on and a movie can have more than one type. The Mean Absolute Error (MAE) is used as an indicator to measure the accuracy of the recommendation algorithm[10].

4.1. Experiment Design

In the experimental part, in order to verify the recommended effect of the multi-weight slope one algorithm, the data is divided into 9 groups according to the number of users. In different numbers of users, we use the multi-weight slope one algorithm and the basic slope one algorithm to predict the score of the items and calculate the MAE value. The experimental results obtained are shown in Figure 1.

In order to verify the influence of the auxiliary item on the multi-weight slope one algorithm, the experiment takes 100 pieces of data that the user does not satisfy the threshold \( T \). In different numbers of users, we use the multi-weight slope one algorithm and the multi-weight slope one algorithm with auxiliary items to predict the score of the items and calculate the MAE value. We set \( \alpha = 0.01, \theta = 0.5, c = 1.0 \). The experimental results obtained are shown in Figure 2.

4.2. Result Analysis

We mark the multi-weight slope one algorithm as MWSO, the multi-weight slope one algorithm with auxiliary items is recorded as AMWSO, and the original slope one algorithm is recorded as SO. Analysis of Figure 1, Figure 2, we can get the following conclusions:

(1) In Figure 1, it can be seen that the MAE of the MWSO algorithm is significantly lower than the SO algorithm in different numbers of users. It shows that the prediction result of MWSO algorithm is better than SO algorithm. MWSO algorithm can get better recommendation effect.

(2) In Figure 2, we set different T values. It can be seen from the comparison experiments that the AMWSO algorithm has the lowest MAE when T=30 and MAE is reduced by 3%. It shows that the auxiliary item can improve the recommendation effect of the multi-weight slope one algorithm.
5. Conclusion
In this paper, a multi-weight slope one algorithm is proposed. The correlation between items is obtained from the scoring data, the preference for the items of users and other information. In the case of sparse data, auxiliary items are added into sparse scoring data to improve the recommendation effect. Experiments on the MovieLens dataset show that the multi-weight slope one algorithm has a better recommendation than the basic slope one algorithm and the MAE value of the multi-weight Slope One algorithm with auxiliary items is 3% lower than the multi-weight Slope One algorithm.

Acknowledgments
This work was financially supported by the Shaanxi Provincial International Co-operation and Exchanges in Science and Technology Plan Project (2016kw-047).

References
[1] Gandhi S, Gandhi M. Hybrid Recommendation System with Collaborative Filtering and Association Rule Mining Using Big Data[C]//2018 3rd International Conference for Convergence in Technology (I2CT). IEEE, 2018: 1-5.
[2] Lemire D, Maclachlan A. Slope one predictors for online rating-based collaborative filtering[C]//Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2005: 471-475.
[3] Wang Q X, Luo X, Li Y, et al. Incremental Slope-one recommenders[J]. Neurocomputing, 2018, 272: 606-618. Jiang T Q, Lu W. Improved slope one algorithm based on time weight[C]//Applied Mechanics and Materials. Trans Tech Publications, 2013, 347: 2365-2368.
[4] Jiang T Q, Lu W. Improved slope one algorithm based on time weight[C]//Applied Mechanics and Materials. Trans Tech Publications, 2013, 347: 2365-2368.
[5] Li W, Gao L, Wang H, et al. Improved Weight Slope One Algorithm of Integrating Expert Similarity and Item Characteristics[C]//2018 Sixth International Conference on Advanced Cloud and Big Data (CBD). IEEE, 2018: 316-321.
[6] Chen M, Hu W, Zheng J. A complementary predictor for collaborative filtering[C]//Proceedings of the 4th International Conference on Computer Engineering and Networks. Springer, Cham, 2015: 443-450.
[7] Grčar M, Mladenič D, Fortuna B, et al. Data sparsity issues in the collaborative filtering framework[C]//International Workshop on Knowledge Discovery on the Web. Springer, Berlin, Heidelberg, 2005: 58-76.
[8] Lin K, Wang J, Wang M. A hybrid recommendation algorithm based on hadoop[C]//Computer Science & Education (ICCSE), 2014 9th International Conference on. IEEE, 2014: 540-543.
[9] Saeed M, Mansoori E G. A new slope one based recommendation algorithm using virtual predictive items[J]. Journal of Intelligent Information Systems, 2018, 50(3): 527-547.
[10] Zhu Y X, Lu L Y. Evaluation metrics for recommender systems[J], Journal of University of Electronic Science and Technology of China, 2012, 41(2): 163-175.