

# Optimization And Implementation Of Item-based Collaborative Filtering Algorithm Based on Attributes and Penalty Factors

Lukun Zhu.

School of Beijing University of Posts and Telecommunications, Beijing 100876, China

**Abstract.** A new item-based collaborative filtering algorithm based on attribute similarity and Penalty was proposed by analyzing the drawbacks of traditional item-based collaborative filtering algorithm according to the similarity between items to select the nearest neighbor[1].The new item-based collaborative filtering algorithm uses the similarity of the item attributes to modify the original item similarity calculation method, and dynamically generates the punish factors according to the item's heat. It comprehensively considers the influence of the item attributes and item heat on the recommendation system, and improves the traditional item similarity measure method. The experimental results on the Movie Lens dataset show that the proposed algorithm can effectively solve the problem of sparse evaluation data and inaccurate recommendation results [2].

**Keywords:** Collaborative Filtering; Attribute similarity; Penalty factor; Recommended system.

## 1. Introduction

At present, many techniques have been proposed for the recommendation system, such as data mining, content filtering, collaborative filtering, integrated use of content filtering and collaborative filtering and so on. One of the hottest recommendation technology is based on collaborative filtering recommendation[3]. There are mainly two types of collaborative filtering recommendation algorithms: user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. The basic idea of the user-based collaborative filtering algorithm is to use the similarity between the target user and other users or user groups to recommend. When the system finds a group of users and preferences similar to the target user, then that they are for a certain item's score is similar. When the system finds that the preferences of a group of users are similar to those of the target user, the user-based collaborative filtering algorithm considers them to be similar. With the outbreak of information on the Internet, how to quickly select the information of interest in the mass information, gradually become a hot research. Recommendation system came into being, and play an increasingly important role in e-commerce systems, blog systems, and so on. user-based collaborative filtering algorithm consider that the nearest neighbors of the target user rating for a item similar to the target user rating the item. With increasing data, the problem of how to find the nearest neighbor from Large-scale users becomes the bottleneck of user-based collaborative filtering algorithm[4]. The main idea of item-based collaborative filtering algorithm is to calculate the similarity between items instead of calculating the similarity between users, effectively solve this problem.

By analyzing sparseness and new item cold-start issues of the traditional item-based collaborative filtering algorithm, A new collaborative filtering algorithm is proposed based on attributes and item-based collaborative filtering algorithm incentive factor, to improve the traditional item similarity measure, and in a way to solve the problem in new items recommended cold start system[5]. we propose a property and incentive factor based on item-based collaborative filtering algorithm, to improve the traditional similarity measurement and solve the problem of cold start in the recommended system.

## 2. Traditional Item-based Collaborative Filtering Technology Principle Analysis

The item-based collaborative filtering algorithm predicts the user's score for the item based on the user's score for the similarity items [6]. The item-based collaborative filtering algorithm can find the nearest neighbors to the target item. Since the current user's score of the nearest neighbor is similar to the score of the target item, therefore the current user's score for the nearest neighbor can be used to

predict the current target's Score, and then select the highest score forecast TOP-N item recommended to the user.

### 2.1 Item Similarity Calculation.

A key step in the item-based collaborative filtering algorithm is to calculate the similarity between items and to filter out several neighbors that are similar to the target item. The basic idea is to calculate the similarity  $sim(i, j)$  of item  $i$  and item  $j$  according to the scores of different users on item  $i$  and item  $j$ .

There are now some mature algorithms to compute the similarity between items. Here we introduce three most popular formulas for item similarity calculation [7].

#### 2.1.1 Cosine-based Similarity.

The similarity calculation for the two items is based on the user-item scoring matrix  $A(m, n)$ , which contains  $m$  users and  $n$  items [8]. And  $R_{ui}$  is the score of the user  $i$  for item  $u$ . If user  $u$  does not score item  $i$ , it is written as  $R_{ui} = 0$ . Let  $i$  be the scoring vector of item  $i$  for  $m$  users and  $j$  be the scoring vector of item  $j$  for  $m$  users. According to its definition, we can denote the prediction  $sim(i, j)$  as

$$sim(i, j) = \frac{\sum_{u=1}^m R_{ui} * R_{uj}}{\sqrt{\sum_{u=1}^m (R_{ui})^2 * \sum_{u=1}^m (R_{uj})^2}} \quad (1)$$

#### 2.1.2 Correlation-based Similarity.

The similarity between the two items  $i$  and  $j$  is calculated by computing the Pearson correlation to make the correlation computation accurate we should isolate the co-rated cases. The formula can be described as

$$sim(i, j) = \frac{\sum_{u=1}^m (R_{ui} - \bar{R}_i) * (R_{uj} - \bar{R}_j)}{\sqrt{\sum_{u=1}^m (R_{ui} - \bar{R}_i)^2} \sqrt{\sum_{u=1}^m (R_{uj} - \bar{R}_j)^2}} \quad (2)$$

#### 2.1.3 Adjusted Cosine Similarity.

In the calculation of item similarity, we should take full account of the impact of the user rating scale on the calculation results, and take the user rating minus the user average score, so as to reduce the deviation caused by different scoring standards. we can denote the prediction  $sim(i, j)$  as

$$sim(i, j) = \frac{\sum_{u=1}^m (R_{ui} - \bar{R}_u) * (R_{uj} - \bar{R}_u)}{\sqrt{\sum_{u=1}^m (R_{ui} - \bar{R}_u)^2} \sqrt{\sum_{u=1}^m (R_{uj} - \bar{R}_u)^2}} \quad (3)$$

### 2.2 Prediction Computation.

The most important step in the collaborative filtering system is to generate the recommended TOP N itemsets based on the similarity calculation, usually using the following two calculation methods[9].

#### 2.2.1 Weighted Sum

as the name implies, this method computes the prediction on an item  $i$  for a user  $u$  by computing the sum of the ratings given by the user on the items similar to  $i$ [10]. each ratings is weighted by the corresponding similarity  $sim(i, j)$  between items  $i$  and  $j$ . Formally, using the notion we can denote the prediction  $P_{ui}$  as

$$P_{ui} = \frac{\sum_{j \in kNNi} sim(i, j) R_{uj}}{\sum_{j \in kNNi} |sim(i, j)|} \quad (4)$$

### 2.2.2 Based on the weighted average of the item

In order to eliminate a single user for a item scoring deviation, using the average score of the item to predict the score[11].  $R_i$ ,  $R_j$  represent the average of the score of item  $i$  and  $j$  by all users. The formula can be described as

$$P_{ui} = \bar{R}_i + \frac{\sum_{j \in kNNi} \bar{a} \text{sim}(i, j)(R_{uj} - \bar{R}_j)}{\sum_{j \in kNNi} \bar{a} |\text{sim}(i, j)|} \quad (5)$$

Formula (5) is superior to the formula (4) in the accuracy of the recommendation, so this paper uses the formula (5) to predict score.

### 2.3 Performance Bottleneck Problem Analysis In Traditional Item-based Collaborative Filtering Algorithm.

The traditional item-based collaborative filtering algorithm provides a theoretical basis for the recommendation system, and derives many commercial recommendation systems.

However, with the outbreak of data, the traditional item-based collaborative filtering algorithm is more and more difficult to support the development of recommendation service on the accuracy and efficiency [12]. By analyzing the performance bottleneck of the traditional item-based collaborative filtering algorithm, based on the item-based collaborative filtering algorithm of attribute and reward and punishment factors, the improved algorithm reduces the impact of high-heat items on recommendation results by introducing bonus factors. At the same time, it solves the problem of user-item sparsity and the cold-start problem of newly added items according to the attribute of product as a basis of recommendation.

## 3. The Penalty Factor Of The Item-based Coordination Filtering Algorithm Based On The Property

Based on the above analysis of the traditional item-based collaborative filtering algorithm, this paper proposes an item-based collaborative filtering algorithm based on attributes and penalties for the accuracy and performance bottlenecks encountered [13]. The core of the algorithm is to improve the search of the nearest neighbor item set of the traditional target item based on the item attributes and the heat of the item [13].

### 3.1 Item Heat Calculation Strategy

In the recommendation system, the interference of the popularity item to the recommendation result is an important factor restricting the development of the recommendation system. For example, the user purchases the macbook and the toothbrush in one purchase. As the toothbrush is a popular commodity, both appear in a purchase behavior [14], cannot explain the two are close neighbors. This paper puts forward the strategy of rewards and punishment factors, according to the heat of the popular item to make the appropriate punishment.

The item heat has a time attribute, and the popularity of the item is different at different times. In order to accurately show the popularity of the item  $i$ . We denote the whole time as  $T$ ,  $T = \{t_1, t_2, t_3, \dots, t_n\}$ , and calculate the pop ( $i$ ) of item  $i$  at different time.

### 3.2 Comprehensive Item Attributes And Item Heat To Improved Item Similarity Calculation

The improved algorithm combines item attribute similarity  $\text{sim}_s(i, j)$  and item scoring similarity  $\text{sim}_t(i, j)$  in linear combination, then the item similarity is obtained as the final similarity between item [15].

$A(m, n)$  is the user-item evaluation matrix, the similarity between items is measured according to the correlation coefficient proposed by Pearson.  $\text{sim}_t(i, j)$  is the item attribute matrix, we assume that the attribute vector of item  $i$  and item  $j$  in  $k$ -dimensional attribute space is  $i = \{i_1, i_2, \dots, i_k\}$ ,  $j = \{j_1, j_2, \dots, j_k\}$ , where the attribute value is 1 or 0, and the similarity  $\text{sim}_s(i, j)$  between item  $i$  and item  $j$  is:

$$sim(i, j) = \frac{i \cdot j}{\|i\|_2 * \|j\|_2} \tag{6}$$

In summary, this paper presents a new formula for computing the similarity of items:

$$sim(i, j) = sim_s(i, j)(1 - a) + asim_t(i, j)q, q = \frac{1}{1 + pop(i)} \tag{7}$$

#### 4. Experimental Results And Analysis

The experiment uses a public data set provided by Movie Lens (<https://movielens.org/>), which evaluates 4,000 movies from 6,000 users, with each user evaluating at least 20 movie items. The 150,000 scoring data of 4600 users were selected as the initial data set, and the sparse level was 98.1%. For the sparse scoring matrix, the SVD prediction scoring data set is used to fill the scoring matrix (unscored item scoring is estimated by the SVD algorithm).

The experiment use mean absolute deviation (MAE) which is used to measure the similarity. The accuracy of the prediction is calculated by calculating the deviation between the predicted user rating and the actual user rating [16]. The smaller the MAE, the better the measure of the similarity method. Assuming that the predicted set of user ratings is  $\{p_1, p_2, \dots, p_n\}$ , the corresponding actual score set is  $\{q_1, q_2, \dots, q_n\}$ . MAE is given by[17]

$$MAE = \frac{\sum |p_i - q_i|}{n} \tag{8}$$

**Experiment 1**  $\alpha$  is an adjustable balance factor, so the value of  $\alpha$  may have an impact on the recommended accuracy. In experiments, changes in MAE were observed by increasing  $\alpha$  from 0 to 1 by 0.1. The experiment is shown in Fig.

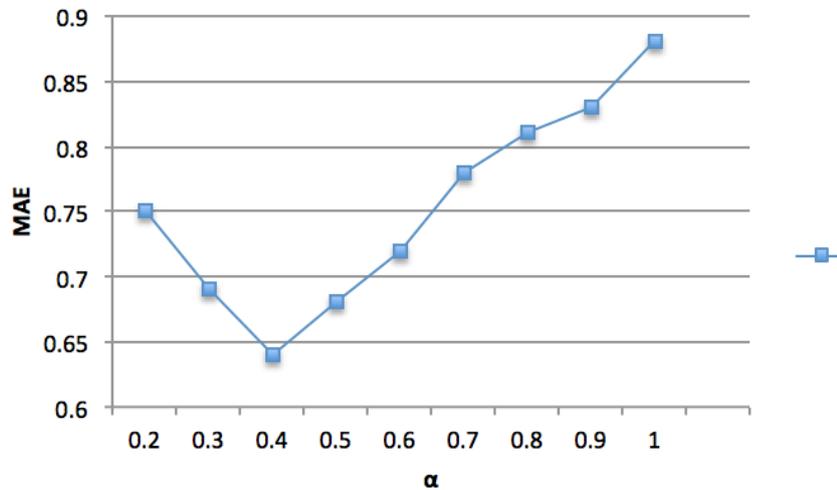


Fig. 1 The MAE And  $\alpha$  Reference

It can be seen from the figure, the value of MAE changes with the change of  $\alpha$ . When the value of  $\alpha$  is 0.4, the value of MAE is the smallest, the best effect is recommended.

**Experiment 2** Since data sparsity can affect the recommendation accuracy. In this paper, we compare the traditional item-based algorithm with different data sparsity and improve the performance of the proposed algorithm.

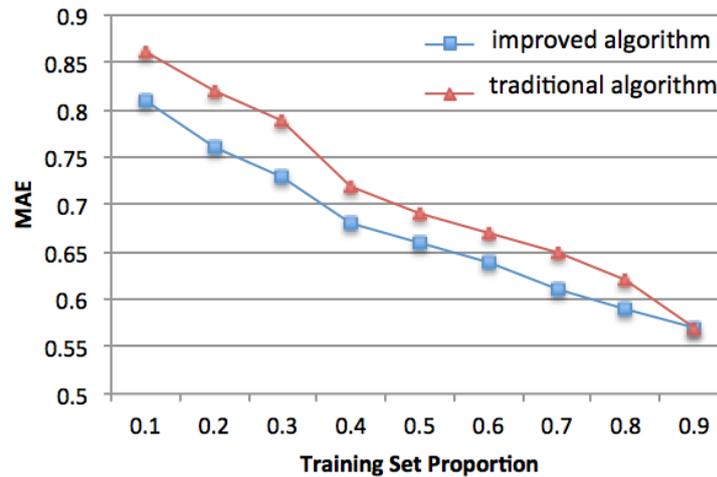


Fig. 2 The MAE And Training Set Proportion Reference

It can be seen from above Figure that the improved algorithm is better than the traditional Item-Based collaborative filtering algorithm in recommending performance.

## 5. Conclusion

Aiming at the shortcomings of the traditional item-based collaborative filtering algorithm, this paper proposes an item-based collaborative filtering algorithm which combines attribute similarity and penalty factor. This algorithm corrects the traditional similarity calculation by the similarity of the item attributes, and introduces the penalty factor to eliminate the interference of the popular item to the recommendation system. Experiments show that the improved algorithm can improve the accuracy of the proposed method to a certain extent.

## References

- [1] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* (2004), p. 143-177.
- [2] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference*. (2001), p. 285-295.
- [3] H. Chai and J.Y. Liu, "Research on improved Slope one recommendation algorithm", *Netinfo Security*, vol. 2. (2015), p. 77-81.
- [4] M. K. Du, M. Liu, S. H. Li and Q. Pu, "Slope one collaborative filtering recommendation algorithm based on neighbor", *Journal of Chongqing University of Posts and Telecommunications*, vol. 26. (2014), no. 3, p. 421-426.
- [5] C.G. Huang, J. Yin and J. Wang, "Uncertain neighbors' collaborative filtering recommendation algorithm", *Chinese Journal of Computers*, vol. 33. (2010), no. 8, p. 1369-1377.
- [6] M S Shang, L Lü and Y C Zhang, "Empirical analysis of web-based user-object bipartite networks [J]", *EPL (Euro physics Letters)*, vol. 90. (2010), no. 4, p. 48006.
- [7] J G Liu, T Zhou and Q Guo, "Review on the evaluation method of the personalized recommender systems [J]", *Complex Systems and Complexity Science*, vol. 6. (2009), no. 3, p. 1-10.
- [8] D. Anand and K.K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities [J]", *Expert Systems with Applications*, vol. 38. (2011), no. 5, p. 5101-5109.

- [9] W. Wang and J. Yang, "An improved collaborative filtering based on item similarity modified and common ratings [c]", 2012 International Conference on Cyber worlds. (2012), p. 231-235.
- [10] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems", *Lecture Notes in Computer Science*, vol. 3290. (2004), p. 492-508.
- [11] S.Z. Vincent and F. Boi, "Using hierarchical clustering for learning the ontologies used in recommendation systems", *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ( 2005) , p. 599-608.
- [12] K. Goldberg, T. Roeder and D. Gupta, "Eigentaste: A constant time collaborative filtering algorithm", *Information Retrieval*, vol. 4. (2001), no. 2, p. 133-151.
- [13] M. Papagelis, D. Plexousakis and T. Kutsuras, "Alleviating the sparsity problem of collaborative filtering using trust inferences", *Proceedings of the 3rd International Conference on ITrust 2005.* (2005), p. 224-239.
- [14] C. Zeng, C.X. Xing and L.Z. Zhou, "Similarity measure and instance selection for collaborative filtering", *Proceedings of the 12th International Conference on World Wide Web.* (2003), p. 652-658.
- [15] J G Liu, T Zhou and Q Guo, "Review on the evaluation method of the personalized recommender systems [J]", *Complex Systems and Complexity Science.* vol. 6 (2009), no. 3, p. 1-10.
- [16] G Linden, B Smith and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering [J]", *Internet Computing IEEE.* vol. 7 (2003), no. 1, p. 76-80.
- [17] C. Jayawardhena, L. T. Wright and C. Dennis, "Consumers online: intentions orientations and segmentation", *International Journal of Retail & Distribution Management.* vol. 35 (2007), p. 515-526.