A Collaborative Filtering Algorithm Based on Relative Similarity for Enhancing Overall Recommendation Diversity (Postprint)

Authors: Jiang Shuhao, Zhang Liyi, Zhang Zhixin

Date: 2017-11-08T00:00:00+00:00

Abstract

Purpose: Aiming to improve the overall diversity of recommendation systems, this study addresses the problem of errors caused by imbalanced and sparse user rating data distribution that affect recommendation accuracy and diversity. Methods: Based on the number of co-rated items between users, a relative similarity index is derived through weighted calculation to modify the similarity computation method, and consequently optimize the rating prediction algorithm. This approach improves overall diversity while ensuring recommendation accuracy, thereby enhancing the long-tail marketing effectiveness for enterprises. Results: Experimental results demonstrate that when the rating threshold is 3.5 and the number of nearest neighbors is 20, the proposed method on the MovieLens dataset improved overall diversity by 114 and accuracy by 6.5% compared with the results using traditional cosine similarity computation. Limitations: The method is only applicable to nearest neighbor-based collaborative filtering algorithms and does not involve other recommendation techniques. Conclusion: The method effectively improves the overall diversity of recommendations, yielding recommendation results with relatively high user satisfaction in both recommendation accuracy and overall diversity.

Full Text

A Collaborative Filtering Algorithm Based on Relative Similarity for Improving Aggregate Recommendation Diversity

Jiang Shuhao[1,2], **Zhang Liyi**[1,2], **Zhang Zhixin**[2]
[1]School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China

[2]Information Engineering College, Tianjin University of Commerce, Tianjin 300134, China

Abstract

[**Objective**] This study aims to improve the overall diversity of recommendation results. The proposed algorithm reduces errors caused by the uneven distribution and sparsity of user rating data, thereby enhancing both recommendation accuracy and diversity. [**Methods**] We first generated a relative similarity index based on the number of common ratings through weighted calculation. Second, we modified the similarity computation method and optimized the rating prediction algorithm. The proposed model improves aggregate diversity while maintaining recommendation accuracy, thereby enhancing the effectiveness of long-tail marketing for enterprises. [**Results**] Experimental results demonstrate that when the rating threshold is 3.5 and the number of nearest neighbors is 20, the proposed method on the MovieLens dataset increases aggregate diversity by 114 and improves accuracy by 6.5% compared to results obtained using traditional cosine similarity calculations. [**Limitations**] This method is only applicable to collaborative filtering algorithms based on nearest neighbors and does not encompass other recommendation techniques. [**Conclusions**] The proposed method effectively improves the aggregate diversity of recommendations while achieving high user satisfaction with both recommendation accuracy and overall diversity.

Keywords: Aggregate diversity; Relative similarity; Collaborative filtering

1. Introduction

For every user, identifying useful information from massive datasets is extremely difficult yet crucial. Personalized recommendation systems represent an important solution to this problem, helping users select the most appropriate information from vast data collections. Recommendation systems identify user preferences to recommend the most suitable or interesting items to specific users. Recommendation algorithms primarily include content-based filtering, collaborative filtering, and hybrid approaches. These systems are widely applied in movies, music, books, tourism, e-commerce, social networks, and web search. While accuracy is a critical metric for evaluating recommendation systems—it assesses whether recommended items are truly suitable for users—such recommendations often contain information users have already obtained through other channels, making them unnecessary in many cases. Consequently, diversity, another important metric for evaluating recommendation systems, has attracted increasing attention from researchers and users. Diversity reflects the differences in the types of recommended items, with some scholars arguing that diversity can, in certain contexts, provide better user satisfaction than accuracy.

Recommendation accuracy and diversity are fundamentally different aspects. A good recommendation system should balance both criteria, yet they are mutually constraining. Significantly improving recommendation diversity inevitably impacts accuracy, resulting in less relevant recommendations. Conversely, high accuracy reduces diversity, causing recommendations to become overly similar and monotonous. Numerous studies have focused on improving recommendation diversity, but most have concentrated on enhancing individual diversity —the diversity within a specific user' s recommendation list—rather than aggregate diversity, which refers to the number of different items recommended to different users. Effectively improving aggregate diversity not only satisfies users' personalized experience requirements but also enhances long-tail marketing effectiveness, helping enterprises maximize profits. Aggregate diversity is not directly related to individual diversity.

Collaborative filtering recommendation systems make recommendations based on user rating data, which often suffers from uneven distribution and sparsity issues that cause recommendation errors. These errors significantly impact both recommendation accuracy and diversity. While much research has focused on improving accuracy, few have addressed diversity, particularly aggregate diversity. This paper focuses on aggregate diversity, striving to improve the system' s overall recommendation diversity while ensuring recommendation accuracy.

Current research defines diversity in two categories: individual diversity and aggregate diversity. Individual diversity is a metric from a single user's perspective, aiming to recommend items with low mutual similarity yet high personal relevance to a particular user. Diversity-focused recommendation has become a very popular research area, with researchers proposing various methods to improve recommendation diversity, though most sacrifice some accuracy and focus primarily on individual diversity.

Aggregate diversity reflects a recommendation system' s ability to recommend different types of items to different users. Unlike individual diversity, aggregate diversity evaluation requires consideration of all users. Although aggregate diversity is not directly related to individual diversity, it represents a broader concept. Some research has targeted aggregate diversity: Lacerda et al. proposed a user interest modeling approach that designs recommendation systems from perspectives of accuracy, novelty, and diversity, improving aggregate diversity by recommending long-tail items with fewer ratings. Park proposed a clustering method based on known rating values or rating frequencies to increase predicted ratings for long-tail items, thereby improving aggregate diversity. Adomavicius et al. proposed an improved item ranking technique to enhance system aggregate diversity, with ranking methods such as item popularity ranking, inverse predicted rating, and neighborhood rating variance providing system designers with greater flexibility and compatibility with different rating prediction algorithms to achieve better aggregate diversity. Fleder et al. studied the impact of recommendation systems on sales diversity, finding that even well-known recommendation systems could reduce sales diversity because they recommend products based on sales and ratings. Bobadilla et al. proposed optimization-based methods to improve aggregate diversity, including greedy algorithms, maximum flow-based methods, and integer programming approaches. The greedy algorithm is an iterative process that replaces already-recommended items with items above a threshold. The maximum flow-based method is a graph algorithm that improves recommendation diversity by formulating a maximum flow problem between users and items. The integer programming method solves multi-criteria optimization problems using accuracy and diversity. While these methods improve aggregate diversity, they significantly impact accuracy. Wang Sen proposed a recommendation method to improve system aggregate diversity and long-tail item recommendation rates by comprehensively considering multiple criteria including predicted item value, item popularity, and item preference. This paper proposes a method to improve recommendation aggregate diversity while ensuring accuracy under conditions of uneven and sparse user rating data.

---

3. Collaborative Filtering Recommendation Based on Relative Similarity

This study employs a user-based collaborative filtering recommendation algorithm. Similarity calculation is a critical step in collaborative filtering, with results decisively affecting the generation of K nearest neighbors and subsequently influencing rating prediction. Commonly used similarity calculation methods include cosine similarity, Pearson correlation coefficient, and Jaccard similarity. This paper uses cosine similarity to calculate similarity between user *u* and other users, and the optimization of similarity specifically targets cosine similarity.

3.1 Cosine Similarity and Rating Prediction

Most collaborative filtering recommendation systems adopt cosine similarity, which has proven highly successful in many studies. Assume *U* is the user set of the recommendation system, *I* is the item set to be recommended, *R(u, i)* is the actual rating of user *u* for item *i*, and *R*(u, i)* is the predicted rating of user *u* for item *i*, as shown in equation (1) [**?**]:

$$Sim(u, u)R(u, i)R(u, i)iI(u, u)R(u, i)iI(u, u)iI(u, u)R(u, i)$$

where *I(u, u')* represents the set of items rated by both users *u* and *u'* . After similarity calculation, the nearest neighbor set *S(u)* can be obtained. Let *R(u)* be the average rating of user *u*, then the predicted rating *R*(u, i)* is calculated as in equation (2) [**?**]:

$$uS(u)R*(u, i)R(u)Sim(u, u)(R(u, i)R(u))|Sim(u, u)|uS(u)$$

Research has found that in many cases, using cosine similarity to improve recommendation accuracy inadvertently reduces aggregate diversity. These situations represent the shortcomings of cosine similarity when dealing with uneven and sparse user rating data.

Example 1. Assume four users u_1, u_2, u_3, u_4 and their rated items: $Ln(u_1) = \{i_1, i_3, i_5\}$, $Ln(u_2) = \{i_3, i_5, i_7\}$, $Ln(u_3) = \{i_3\}$, $Ln(u_4) = \{i_2, i_3, i_7\}$. Clearly, u_3 is the nearest neighbor of the other three users because in this case they only share one commonly rated item i_3. Therefore, the cosine similarity between user u_3 and other users is 1, as cosine similarity typically considers two users with one common rated item to have a similarity of 1 regardless of individual rating differences.

Example 2. Assume four users with ratings: $u_1 = \{2, 2, 2\}$, $u_2 = \{3, 3, 3\}$, $u_3 = \{5, 5, 5\}$, $u_4 = \{2, 5, 3\}$. The similarity between user u_4 and the other three users is identical at 0.9366. This problem becomes more severe when the number of ratings is very limited because insufficient valid ratings increase the probability of computational errors. If this occurs in user *u*' s nearest neighbors, user *u*' s predicted rating *R*(u, i)* will equal the average rating since *R(u*' *, i)* equals *R(u*' *)* (see equation (2)). Therefore, the fewer ratings a user has, the greater the prediction error, subsequently affecting both recommendation accuracy and aggregate diversity.

3.2 Optimized Similarity Algorithm

The solution to this problem involves weakening users with few ratings (whether total ratings or individual item ratings) and strengthening users with abundant rating data. The specific approach considers the number of commonly rated items between two users. In reality, the more items two users have rated in common, the higher their relative similarity should be. Therefore, the number of commonly rated items should be an important factor in similarity calculation. Based on this principle, we designed the Relative Similarity (RS) algorithm as follows:

CR: Number of common rated items
MCR: Maximum number of common rated items

Input: User set *U*, item set *I*, similarity between any two users *simuser,any_{user}*
Output: Relative similarity *RS*

```
CR=0, MCR=0
for user = 1 to |U|-1 do
    MCR=0
    for any_{user}=user+1 to |U| do
        CR=|I(user, any_{user})|
        if CR > MCR then
            MCR=CR
        end
```

```
      end
  end
for user=1 to |U| do
    for any_{user}=user+1 to |U| do
        CR=|I(user, any_{user})|
        W=(CR/MCR)
        RS=W*simuser,any_{user}
```

This algorithm modifies the similarity calculation results between users after initial similarity computation. It has three input parameters: user set, item set, and calculated similarity between users. Lines define two variables: *CR* (the number of common rated items between current user and any user) and *MCR* (the maximum number of common rated items between current user and all users). The inner loop from lines to calculates the maximum number of common rated items between the current user and other users, while the outer loop from lines to determines the maximum number of common rated items among all user pairs, preparing data for weight usage in the algorithm' s second part. The second part (lines -) applies weighted correction to previously calculated similarity results. Line sets the ratio of the number of common rated items between *any_{user}* and current user *user* to the maximum number of common rated items for the current user as the weight *W*. Line uses this weight to modify the similarity calculation result, yielding relative similarity *RS*. When two users have a relatively large *CR*, *W* approaches 1, indicating higher similarity, while a smaller *CR* makes *W* approach 0, indicating lower similarity, consistent with the algorithm' s design. Nearest neighbors calculated using the modified similarity values are more accurate.

For instance, in Example 1 from Section 3.1 where $Ln(u_1) = \{i_1, i_3, i_5\}$, $Ln(u_2) = \{i_3, i_5, i_7\}$, $Ln(u_3) = \{i_3\}$, $Ln(u_4) = \{i_2, i_3, i_7\}$, assuming only these four users exist in the user set, the maximum number of common rated items for user u_1 with other users is 2. Therefore, the relative similarity *RS* between u_1 and u_3 is $1/2 = 0.5$. With more users, the similarity between u_1 and u_3 could be even lower. In Example 2, where four users collectively rated three items, the relative similarity *RS* between u_4 and the other three users should be $0.9366/3 = 0.3122$. Comparing these values, the optimized relative similarity better reflects true user similarity.

Using relative similarity *RS* in rating prediction adjusts misleading similarity through weighting common rated items, giving greater influence to items with more common ratings in similarity calculation while reducing the weight of items with fewer common ratings. This prevents misleading similarities from being considered as nearest neighbors. The modified predicted rating calculation is shown in equation (3):

$$R*(u,i)R(u)uS(u)u,u*(R(u,i)R(u))uS(u)$$

Using this formula for predicted rating calculation yields recommendation results with relatively high user satisfaction in both accuracy and aggregate diversity.

---

4. Experiments and Results

4.1 Dataset

The experiments used a subset of the publicly available MovieLens dataset. The MovieLens dataset contains 100,000 ratings from 943 users on 1,682 movies, with rating values ranging from 1 to 5. The data subset was partitioned into 80% training set and 20% test set. The following operations were implemented in both datasets: creating a user-item matrix, calculating nearest neighbors in collaborative filtering using the modified similarity equation, and performing rating prediction after determining nearest neighbors. Finally, recommended items were determined based on the criterion whether the predicted rating *R*(u, i)* for user *u* on item *i* exceeded the rating threshold.

4.2 Evaluation Metrics

The algorithm design aims to improve aggregate diversity while maintaining accuracy; therefore, evaluation should consider both metrics. After predicting ratings for target items, the algorithm generates final recommendation lists by setting a rating threshold. The associated threshold is defined as *Tr*. For each predicted rating, if *R*(u, i) ≥ Tr*, recommendation accuracy is calculated as follows [**?**]:

$$accuracy\frac{|result(Ln(u))|}{|(L(u))|}$$

where *Ln(u) = {*i_1*,* i_2*, ···, i }* represents the top *n* recommended items, and *result(Ln(u)) = {i Ln(u) / R(u, i) ≥ Tr}* represents recommended items whose actual ratings exceed the specified threshold.

Even highly accurate recommendation systems cannot guarantee user satisfaction with recommendation results. Another critical aspect is the variety of recommended items, evaluated through recommendation system diversity metrics. However, different researchers employ varying diversity metrics. Diversity comprises individual diversity and aggregate diversity: individual diversity measures intra-user variety—the diversity of items recommended to a single user—while aggregate diversity measures a system' s ability to recommend different items to different users. This paper employs the following aggregate diversity calculation formula [**?**]:

$$diversity\frac{|L(u)|}{|U|}uUn$$

4.3 Experimental Results Analysis

Figure 1 [**Figure 1: see original paper**] compares accuracy before and after using relative similarity with three dashed lines showing accuracy variation with rating threshold using traditional similarity calculation for nearest neighbor counts of 10, 20, and 50, and three solid lines showing accuracy variation under the same conditions using relative similarity.

The data in Figure 1 shows that with the same rating threshold, recommendation accuracy improves as the number of nearest neighbors increases—a similar trend in both datasets. This indicates that the number of nearest neighbors positively influences recommendation accuracy, though more neighbors simultaneously increase prediction rating computational complexity. The comparison between dashed and solid lines reveals that relative similarity not only fails to reduce recommendation accuracy but actually improves it, with noticeable enhancement when the rating threshold is 3.5. These results demonstrate that the algorithm successfully maintains recommendation accuracy as designed.

Figure 2 [**Figure 2: see original paper**] similarly uses three dashed and three solid lines to compare aggregate diversity before and after applying relative similarity. The results show that aggregate diversity significantly improves when using relative similarity. Experimental data indicates that at a rating threshold of 3.5, diversity increases from 98 to 221 when nearest neighbors is 10, from 87 to 201 when nearest neighbors is 20, and from 79 to 127 when nearest neighbors is 50.

Additionally, the solid lines in Figure 2 show that when the threshold is small, aggregate diversity values significantly improve compared to not using relative similarity. However, as the threshold gradually increases, especially reaching 4, aggregate diversity values approach those of the original method. Figures 1 and 2 also reveal that when the threshold is 4, although accuracy is relatively high, diversity values decrease substantially, leading to the conclusion that higher rating thresholds result in more concentrated recommendation varieties. The comparison of solid line data further shows that at a threshold of 3.5, recommendation accuracy is lowest while diversity is highest, demonstrating their mutually constraining relationship.

This paper proposes an effective method for improving recommendation aggregate diversity and introduces a relative similarity-based approach for analyzing recommendation diversity. Experimental results demonstrate that compared to previous research methods, the proposed recommendation model effectively optimizes both aggregate diversity and accuracy as the threshold increases. Moreover, at relatively high thresholds (e.g., 3, 3.5), the system' s recommendations maintain high accuracy while achieving good diversity optimization.

The proposed method primarily focuses on improving recommendation system aggregate diversity. Previous research has addressed individual diversity optimization, and investigating the correlation between these two diversity types and comprehensive improvement methods represents a direction for future work.

References

[1] Adomavicius G, Kwon Y. Optimization-based Approaches for Maximizing Aggregate Recommendation Diversity [J]. *Informs Journal on Computing*, 2014, 26(2): 351-369.

[2] Shambour Q, Lu J. An Effective Recommender System by Unifying User and Item Trust Information for B2B Applications [J]. *Journal of Computer and System Sciences*, 2015, 81(7): 1110-1126.

[3] Yigit M, Bilgin B E, Karahoca A. Extended Topology Based Recommendation System for Unidirectional Social Networks [J]. *Expert Systems with Applications*, 2015, 42(7): 1342-1350.

[4] Adomavicius G, Kwon Y. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2012, 24(5): 896-911.

[5] Núñez-Valdez E R, Lovelle J M C, Martínez O S, et al. Implicit Feedback Techniques on Recommender Systems Applied to Electronic Books [J]. *Computers in Human Behavior*, 2012, 28(4): 1186-1193.

[6] Bradley K, Smyth B. Improving Recommendation Diversity [C]. In: *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science*. Maynooth, Ireland. 2001.

[7] Zhang M, Hurley N. Avoiding Monotony: Improving the Diversity of Recommendation Lists [C]. In: *Proceedings of the 2nd ACM Conference on Recommender Systems*. ACM, 2008.

[8] Chen J, Liu Y, Hu J, et al. A Novel Framework for Improving Recommender Diversity // Behavior and Social Computing [M]. Springer International Publishing. 2013.

[9] Aytekin T, Karakaya M Ö. Clustering-based Diversity Improvement in Top-N Recommendation [J]. *Journal of Intelligent Information Systems*, 2014, 42(1): 1-18.

[10] Bobadilla J, Ortega F, Hernando A, et al. Recommender Systems Survey [J]. *Knowledge Based Systems*, 2013, 46: 109-132.

[11] Lacerda A, Ziciani N. Building User Profile to Improve User Experience in Recommender Systems [C]. In: *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. 2013.

[12] Park Y J. The Adaptive Clustering Method for the Long Tail Problem of Recommender Systems [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2013, 25(8): 1904-1915.

[13] Fleder D, Hosanagar K. Blockbuster Culture' s Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity [J]. *Management Science*, 2009, 55(5): 697-712.

[14] Wang Sen. A Recommendation Algorithm Based on Aggregate Diversity Enhancement [J]. *Computer Engineering & Science*, 2016, 38(1): 183-187.

Author Contributions

Jiang Shuhao, Zhang Liyi: Conceived the research idea and designed the study; **Jiang Shuhao, Zhang Zhixin:** Conducted the experiments; **Zhang Zhixin:** Collected, cleaned, and analyzed the data; **Jiang Shuhao:** Drafted the manuscript; **Zhang Liyi, Jiang Shuhao:** Revised the final version of the paper.

Conflict of Interest Statement

All authors declare that they have no conflict of interest.

Supporting Data

The supporting data is self-archived by the authors, E-mail: mr_{jiang1980}@163.com.

[1] Jiang Shuhao. u.user.txt. MovieLens user raw data.
[2] Jiang Shuhao. u.item.txt. MovieLens item raw data.
[3] Jiang Shuhao. u.data.txt. MovieLens user rating raw data.
[4] Jiang Shuhao. u.genre.txt. MovieLens movie genre raw data.
[5] Jiang Shuhao. Test set recommendation results (traditional cosine similarity).txt. Test set recommendation results without RS.
[6] Jiang Shuhao. Test set recommendation results (relative similarity).txt. Test set recommendation results using RS.

Received: 2016-08-15
Revised: 2016-09-19

Note: Figure translations are in progress. See original paper for figures.