

Application of collaborative filtering in movie recommendation systems and improvements by hyperparameter tuning

Yicheng Long

Oxford Brookes University & Chengdu University of Technology, No. 1, East 3rd Road, Erxianqiao, Chenghua District, Chengdu, 610059, China

2624448961@qq.com

Abstract. In the current era of explosive data growth, accurately recommending movies to users has become a challenge for traditional recommendation algorithms. In this paper, we propose enhancements to the traditional item-based Collaborative Filtering recommendation algorithm by focusing on three aspects: the proportion of the training set and test set, the new similarity algorithm, and the new recall index. These enhancements aim to achieve better recommendation results. We conducted experiments using a movie recommendation system as the testbed and implemented an item-based recommendation algorithm using the Python language. A control experiment was performed using the dataset from the official MovieLens website. The experimental results demonstrate that the improved algorithm exhibits enhanced recommendation accuracy.

Keywords: Collaborative Filtering, Recommendation Algorithm, Similarity Calculation, Recall At K.

1. Introduction

In the era of big data, with the explosion of data, improving the accuracy of recommendation algorithms has become a prominent research topic. From a practical standpoint, enhancing recommendation accuracy can assist users in obtaining valuable information quickly and efficiently. Therefore, designing a well-founded recommendation algorithm to improve accuracy is of paramount importance.

This paper aims to address the challenge of prediction in recommendation systems by analyzing different collaborative filtering algorithms based on item and user perspectives. The advantages and disadvantages of these approaches will be examined, along with an analysis of how to consider similarity and provide recommendations based on similarity to enhance the accuracy of the recommendation algorithm. Currently, the prevalent algorithm for addressing recommendation problems is collaborative filtering, first proposed by Goldberg, Nichols, Oki, and Terry in 1992 and implemented in the Tapestry system [1]. Collaborative filtering algorithms can be broadly categorized into two ideas: user-based and item-based. The user-based approach assumes that users with similar scores on one item are likely to have similar scores on other items. Users similar to a target user are considered neighbors, and the algorithm predicts the user's evaluation of an item based on the evaluations of the item by the nearest neighbors. On the other hand, the item-based approach predicts that a user will assign similar scores to

items that are similar to an item they have already scored [2]. After determining the approach, the collaborative filtering algorithm calculates similarity using three traditional methods: cosine similarity, modified cosine similarity, and correlation similarity [3]. Finally, recommendations are generated by incorporating the calculated similarity into the recommendation strategy. Currently, most collaborative filtering recommendation systems employ the average weighting strategy [4]. Collaborative filtering has demonstrated promising results in e-commerce recommendations and other domains.

This article will be divided into six main parts. The second part will provide a literature review, exploring the concepts and theories of collaborative filtering recommendation algorithms and summarizing previous research. The third part will focus on the traditional item-based collaborative filtering recommendation algorithm, presenting the formulas and related concepts involved in the improved algorithm. The fourth part will explain the experimental process and present the results. In the fifth part, the experimental results will be discussed, and the advantages, disadvantages, and potential applications of the improved recommendation algorithms will be analyzed from three perspectives. Finally, the sixth part will provide a summary, discussing the limitations of this study and suggesting future research directions.

2. Literature review

The first collaborative filtering-based recommendation system was Tapestry, but it had limitations and couldn't be applied to large-scale user groups [5]. Subsequently, the GroupLens system gained prominence for recommending news and movies based on ratings [6]. E-commerce platforms like Amazon also started utilizing collaborative filtering recommendation systems [7]. However, challenges still exist in collaborative filtering recommendation systems. Zhang et al. [8] proposed a new algorithm that overcomes the limitations of traditional algorithms when dealing with extremely sparse data. This algorithm employs cloud modeling and knowledge-level user similarity comparison. It calculates user similarity using the cloud model, introduces a user similarity comparison method based on the cloud model (LICM) to quantify the role of bridges in knowledge transformation, and then calculates the similarity matrix using the LICM method. The user's nearest neighbors are determined based on the user to be recommended and the item to be evaluated, and the item's score is predicted using a weighted average strategy. Experimental results using the MovieLens dataset demonstrate that this algorithm outperforms traditional recommendation algorithms. Five years later, Li et al. [9] proposed a collaborative filtering recommendation algorithm based on user scenario fuzzy clustering. This algorithm efficiently addresses the issues of data sparsity and poor scalability inherent in traditional recommendation algorithms. It utilizes a fuzzy clustering algorithm to classify user groups with similar scenarios based on user scenario information. Additionally, it applies the SlopeOne algorithm to fill the user-item scoring matrix prior to collaborative filtering. Testing with the MovieLens dataset shows that the improved algorithm significantly enhances recommendation accuracy and resolves data sparsity problems. In recent years, researchers have proposed various ideas to enhance recommendation accuracy. For instance, Wang et al. [10] introduced a collaborative filtering recommendation algorithm based on item fuzzy similarity. This algorithm aims to improve the resolution of fuzzy and sparse problems in recommendation algorithms, leading to enhanced prediction accuracy. The algorithm employs trapezoidal fuzzy numbers to describe the mapping relationship between scoring and satisfaction, enhances the fuzzy similarity calculation strategy, utilizes membership functions to assess tag-item ownership, calculates similarity based on item tags, and improves the scoring prediction strategy. Experimental results using the MovieLens 100K and 1M datasets demonstrate that this algorithm achieves improved mean absolute error (MAE), coverage, accuracy, and efficiency compared to traditional collaborative filtering algorithms. It also helps alleviate the ambiguity problem and mitigates the adverse impact of sparse scoring data to some extent. Hao et al. [11] proposed a collaborative filtering algorithm that integrates multiple types of context information to enhance personalized recommendation quality. This algorithm represents user-item interactions as a bipartite graph and constructs different similarity networks based on distinct context characteristics. By designing a joint matrix decomposition objective function constrained by multiple context information networks, it learns

the representation learning of user-item pairs. Experimental results using the Amazon Software dataset, which represents today's top question and answer context, show that the algorithm improves various recommendation indicators and effectively addresses sparsity in recommendation system data. Sun and She [12] optimized the collaborative filtering recommendation algorithm using a dichotomy-based approach and developed an intelligent recommendation model for sports training resources. The algorithm improves the existing algorithm by incorporating the Euclidean formula and binary k-means algorithm. Furthermore, it combines them with the dichotomy search algorithm to obtain an optimized recommendation algorithm. Experimental results using a designated dataset show that the optimized recommendation algorithm outperforms other algorithms in terms of mean absolute error (MAE) and exhibits higher recommendation accuracy. Additionally, Lin et al. [13] proposed a collaborative filtering algorithm based on singular value decomposition (SVD) and a popularity-based recommendation algorithm for food stores. This approach addresses the adaptability issue between recommended restaurants and user preferences. The method models users, restaurant sign-in records, and restaurant popularity. It captures users' latent preferences based on restaurant sign-in records and recommends restaurants by considering popularity. Experimental results using the Hejing Community Food Dataset, which consists of 40,000 restaurants, 540,000 users, and 4.4 million comments, show that the hybrid recommendation algorithm combining popularity and SVD significantly improves recommendation accuracy and comprehensiveness in the food restaurant domain, yielding positive recommendation outcomes.

3. Methodology

The algorithm primarily used in this paper is item-based collaborative filtering algorithm, which consists of the following steps:

Step 1: Data collection and preprocessing

Step 2: Splitting the sample data into training and testing sets in a certain ratio

Step 3: Calculating the similarity between items using the training set samples to obtain a similarity matrix

Step 4: For the target user's item interaction records in the testing set, identifying highly similar items and calculating the weighted sum to determine the recommendation score for each item with respect to that user

Step 5: Sorting the items based on the recommendation score and selecting the top K items for recommendation

Step 6: Evaluating the model's performance using statistical metrics such as accuracy, precision, recall, and coverage

This paper aims to improve the item-based collaborative filtering recommendation algorithm from three perspectives: the ratio of training and testing sets, similarity algorithm, and a new recall rate metric. The influence of these factors on the recommendation algorithm will be studied to achieve better recommendation results.

Traditionally, the training set ratio is commonly set to 0.75. This paper will attempt to change the ratio to 0.70, 0.80, and 0.90, respectively, and conduct separate tests to observe the impact of the training and testing set ratio on the recommendation results.

In traditional collaborative filtering, the similarity calculation method is given by $sim_{ij} = \frac{n_{ij}}{\sqrt{n_i * n_j}}$

where n_{ij} represents the number of users who have interacted with both items i and j , and n_i, n_j represent the number of users who have interacted with items i and j individually. However, it is observed that users with different numbers of interacted items have varying contributions to the similarity calculation between two items. Users with more interacted items have a smaller contribution to the similarity calculation between the two items. Therefore, we consider the different contributions of each user as $\frac{1}{\ln(1+u)}$, where u represents the cumulative number of items the user has interacted with.

Thus, the improved similarity calculation formula is as follows:

$$\frac{1}{\ln(1+u)} sim_{ij} = \frac{\sum \frac{1}{\ln(1+u)}}{\sqrt{n_i * n_j}}$$

On the other hand, the recall algorithm in traditional collaborative filtering does not consider the impact of the system's recommended quantity on the accuracy of recommendations. Therefore, recall at K is introduced to analyze the recommendation performance. The calculation formula is as follows:

$$recallatK = \frac{hit}{\sum min(testcount, K)}$$

where hit represents the total number of successful recommendations for all users, the sum runs over all users, and testcount represents the number of items the user has interacted with in the testing set.

4. Results

4.1. Dataset and preprocessing

The dataset used in this paper is the official MovieLens dataset, which includes userID, movieID, rating, and timestamp. The dataset consists of 600 users who provided 100,000 ratings for 9,000 movies and applied 3,600 tags. The dataset was last updated in September 2018. The experiments were conducted using Python programming language, with Python interpreter version 3.9 and Pycharm 3.3 as the compiler.

4.2. Numerical experiments

First, a set of baseline experiments was conducted as the control group, where the number of recommended movies for each user was set to 10. The experimental conditions were as follows: the initial training set ratio was set to 0.75, traditional similarity calculation method was used $sim_{ij} = \frac{n_{ij}}{\sqrt{n_i * n_j}}$, and model performance evaluation was conducted using metrics such as precision, recall, and coverage. The recall metric was based on the traditional recall index. The experimental results are shown below:

Table 1. Baseline experiment results.

Percentage of training set	Precision	Recall	Coverage
0.75	0.2803	0.0676	0.0665

Next, three sets of experiments were conducted:

4.2.1. Based on the control group, the training and testing set ratio was adjusted to 0.70, 0.80, 0.90, and so on, and repeated experiments were performed to obtain the following results:

Table 2. Experimental results with adjusted training set ratio.

Percentage of training set: Precision Recall Coverage.

Percentage of training set:	Precision	Recall	Coverage
0.75	0.2803	0.0676	0.0665
0.70	0.2890	0.0585	0.0714
0.80	0.2539	0.0766	0.0604
0.90	0.1485	0.0897	0.0497

4.2.2. Based on the control group, the new improved similarity calculation formula $sim_{ij} = \frac{\sum \frac{1}{\ln(1+u)}}{\sqrt{n_i * n_j}}$ was tested, and repeated experiments were conducted, resulting in the following results:

Table 3. Experimental results with adjusted similarity calculation.

	Percentage of training set:	Precision	Recall	Coverage
Traditional similarity	0.75	0.2803	0.0676	0.0665
New Similarity	0.75	0.2956	0.0711	0.0599

4.2.3. Based on the control group, additional consideration was given to the recall at K metric, and repeated experiments were conducted, resulting in the following results:

Table 4. Experimental results using recall at K.

Ratio of training set to test set:	Similarity	Precision	Coverage	Recall at K
0.75	Traditional	0.2659	0.0627	0.2659

5. Discussion

Based on the analysis and comparison of the experimental results in 4.2.1, it can be observed that the Precision and Recall metrics show a negative correlation. When the training set ratio decreases, Precision tends to increase while Recall decreases. Conversely, when the training set ratio increases, Precision tends to decrease while Recall increases. The main reason for this phenomenon is as follows: Precision represents the proportion of correctly predicted data among the predicted data, which, in this experiment, refers to the proportion of successfully recommended movies out of the 10 recommended movies for each user. Clearly, for Precision, the denominator, which is the number of recommended movies (i.e., 10), remains fixed, while the numerator, which represents the number of correctly recommended movies, decreases when the training set ratio increases. Consequently, Precision decreases. On the other hand, Recall represents the probability of correctly identified samples out of all the correct samples as perceived by the model. In this experiment, it represents the proportion of correctly recommended movies out of all the movies. The numerator of Recall is limited by the number of recommended movies (i.e., 10), indicating that the maximum number of correctly recommended movies can only be 10. In this case, the magnitude of Recall is mainly determined by the denominator, which is the total number of movies in the test set, representing the number of movies that the user has watched. Therefore, when the training set ratio increases, the test set ratio decreases, resulting in a smaller denominator and higher Recall. Additionally, it is observed from the results that increasing the training set ratio leads to a decrease in coverage. This is because an increase in the training set ratio reduces the sample size of the test set. Coverage has the total number of movies as its denominator and the number of recommended movies in the test set as its numerator. Thus, a decrease in the test set sample size leads to a decrease in the numerator and subsequently a decrease in coverage.

According to the analysis and comparison of the experimental results in 4.2.2, it can be concluded that after considering the factor that users have different numbers of interacted items and users with more interacted items have a smaller contribution to the similarity calculation between two items, adopting the improved similarity calculation with the logarithmic function, which depends on the specific user's interaction quantity with items, results in lower contributions from users with more interacted items in the calculation of item similarity. When compared to the control group data, we found that under the same conditions of training set ratio and Recall metric, both Precision and Recall increased simultaneously. Specifically, Precision increased by 1.53% and Recall increased by 0.35%. This is a strong indication that the new algorithm can effectively improve the recommendation performance. Typically, Precision and Recall are negatively correlated according to their definitions. However, after improving the similarity algorithm in this experiment, both Precision and Recall increased, which to some extent indicates that the measure of improving the similarity algorithm can enhance the overall recommendation performance of the model.

Based on the results in 4.2.3, it is evident that the new metric, Recall at K, directly increased from the original value of 0.0676 to 0.2659. Compared to the original Recall, Recall at K has a more significant reference value in terms of magnitude. The reason for this result is that the numerator of Recall represents the number of correctly recommended movies, constrained by the total number of recommended movies. In this experiment, the total number of recommended movies is set to 10. On the other hand, the denominator of Recall represents the total number of movies that the user has watched. When a user has watched more than 10 movies, the denominator increases, resulting in a loss of the original reference significance based on recommending 10 movies. Therefore, by adopting the new metric Recall at K, when the number of movies watched by the user exceeds the number of recommended movies, the number of recommended movies is used as the denominator, making Recall at K a more reasonable and meaningful metric to consider.

6. Conclusion

To conclude, this article focuses on enhancing traditional recommendation algorithms to achieve improved recommendation accuracy. A comprehensive review of the application and development of Collaborative filtering algorithms in this domain was conducted. Furthermore, a Python program for an item-based recommendation algorithm was implemented to recommend users' favorite movies. Comparative tests were then performed, considering three aspects: modifying the ratio of the training set and test set, refining the similarity algorithm, and enhancing the recall index algorithm. The experimental dataset utilized was obtained from the official MovieLens website. The results of the experiments revealed several key findings. Firstly, reducing the proportion of the training set led to higher precision but lower recall. Secondly, the new similarity algorithm demonstrated a precision improvement of 1.53% and a recall improvement of 0.35%. Lastly, the new recall rate algorithm, Recall at K, provided a more accurate reflection of the recommendation effectiveness by avoiding errors arising from users encountering more movies than recommended ones. These results demonstrate that the three proposed improvements to the Collaborative filtering algorithm in this paper contribute to enhancing the recommendation effectiveness to a significant extent.

References

- [1] Goldberg, D., Nichols, D., Oki, B., & Terry, D.. (1992). Using Collaborative Filtering to Weave an Information Tapestry.
- [2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J.. (2001). Item-based collaborative filtering recommendation algorithms. ACM.
- [3] Xu, C., Xu, J., & Du, X.. (2006). Recommendation algorithm combining the user-based classified regression and the item-based filtering. Proceedings of the 8th International Conference on Electronic Commerce: The new e-commerce - Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet, 2006, Fredericton, New Brunswick, Canada, August 13-16, 2006. ACM.
- [4] Ai-Lin, D., Yang-Yong, Z., & Bai-Le, S.. (2003). A collaborative filtering recommendation algorithm based on item rating prediction. Journal of Software, 14(9), 54-65.
- [5] Ma, H. W., Zhang, G. W., & Li, P.. (2009). Survey of Collaborative Filtering Algorithms. Journal of Chinese Computer Systems(7), 7.
- [6] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J.. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Acm Conference on Computer Supported Cooperative Work. MIT Center for Coordination Science.
- [7] Yan, W. U., Jie, S., Tian-Zhu, G. U., Xiao-Hong, C., Hui, L. I., & Shu, Z.. (2007). Algorithm for sparse problem in collaborative filtering. Application Research of Computers, 24(6), 94-97.
- [8] Zhang, G. W., Li, D. Y., Li, P., Kang, J. C., & Chen, G. S.. (2007). A Collaborative Filtering Recommendation Algorithm Based on Cloud Model. Journal of Software.
- [9] Li, H., Zhang, Y., & Sun, J. H.. (2012). Research on Collaborative Filtering Recommendation Based on User Fuzzy Clustering. Computer Science, 39(12), 4.

- [10] Wang, S., Chen, L., & Zhang, J.. (2021). Collaborative filtering recommendation algorithm based on item fuzzy similarity. *Application Research of Computers*.
- [11] Hao, Z. F., Liao, X. C., When, W., & Cai, R. C.. (2021). Collaborative filtering Recommendation Algorithm Based on Multi-context Information. *Computer Science*, 048(003), 168-173.
- [12] Sun, Y., & She, L.. (2022). Intelligent Sports Auxiliary Training Method Based on Collaborative Filtering Recommendation Algorithm. *Wireless Communications and Mobile Computing*, 2022. <https://doi.org/10.1155/2022/8703707>
- [13] Lin, S. J., Yu, T., & Chen, F. Y.. (2012). Food store recommendation algorithm based on Collaborative filtering. *Computer Knowledge and Technology*.