

A Self-Evaluating Collaborative Filtering Recommendation Algorithm: A Case Study of Anime Recommendations

Baiqian Zhang

School of Data Sciences, Brunel University London, London, UK

13522861833@163.com

Abstract. This paper introduces a collaborative filtering recommendation algorithm aimed at addressing the issues of information insufficiency and the mismatch of user personalization needs in anime recommendations. Firstly, we review relevant literature to explore the application of collaborative filtering algorithms in recommendation systems and previous research findings. Then, we detail the design and implementation of collaborative filtering algorithms based on anime and user data, calculating similarities between anime and between users respectively for recommendation purposes. Finally, a self-evaluation is conducted to achieve optimal recommendation performance. Experimental results show that when recommending 10 anime titles, the ROC analysis results indicate a high level of precision, with both algorithms performing well in terms of accuracy.

Keywords: recommendation system, collaborative filtering, ROC analysis, personalized anime recommendation.

1. Introduction and Background

With the rapid development of internet technology, people increasingly rely on digital platforms for information, entertainment, and social interaction in their daily lives. As a popular form of entertainment, anime has achieved great success over the past few decades and has become a significant part of the global cultural entertainment industry. In this diverse and vast anime market, the introduction and classification of anime are often insufficient and inadequate. Viewers usually rely on anime communities, video websites, and other channels to learn about anime works before choosing to watch them. Providing personalized anime recommendations has thus become a challenging task.

To address this challenge, recommendation systems have emerged as a solution. Anime recommendation systems not only enhance the viewing experience for users but also provide personalized services for anime platforms, increasing user engagement and satisfaction. Collaborative filtering recommendation algorithms, as a classic and effective recommendation technique, can play an important role in anime recommendations through reasonable implementation and optimization. Additionally, ROC analysis is introduced in experiments to evaluate recommendation performance, aiming to achieve the best recommendation results.

This paper is divided into several sections: first, a review of the classic literature, discussing the application of collaborative filtering algorithms in the recommendation field and previous research findings; next, a detailed introduction to the design and implementation details of collaborative filtering recommendation algorithms in anime recommendations; then, an evaluation and analysis of the

algorithm, exploring its advantages, disadvantages, and application scenarios; finally, a summary of the research content, discussion of its limitations, and suggestions for future research directions.

2. Literature Review

In recent years, recommendation systems and collaborative filtering algorithms have received widespread attention, with researchers proposing various improvement methods and application scenarios. Zhao Ming et al. proposed a robust recommendation algorithm based on trust relationships and interest similarity, achieving a recommendation accuracy of 90.39% by removing irrelevant noise users [1]. Han Junbo et al. developed a game Feeds Ads recommendation system based on user POI game preference data similarity algorithm [2]. Chen Hui et al. proposed a probabilistic matrix factorization recommendation algorithm incorporating user trust to address the issue of sparse user rating information [3]. Zhang Li et al. introduced explicit user relationship data and designed a hybrid similarity measurement method, proposing a collaborative filtering recommendation algorithm based on rating subspace and trust mechanism. Compared with the TrustASVD++ method, the MAE and RMSE of the URSTM algorithm improved by 0.78% and 0.66%, respectively [4]. Liu Wenwen et al. proposed an improved collaborative filtering recommendation method incorporating item popularity penalty factors, which was experimentally verified [5]. Jiang Jiulei et al. proposed a neural network recommendation algorithm T-NAMF integrating trust and attention mechanisms, significantly improving recommendation effectiveness, especially when the variable length $k=3$ [6]. Sun Ping proposed an XML-based collaborative filtering recommendation method for English learning resources, solving for comprehensive similarity through weighted fusion [7]. Deng Renfeng combined the RFMRQ model and collaborative filtering recommendation algorithm, organizing and analyzing historical data of clothing e-commerce platforms, finding that the model outperforms the traditional RFM model in terms of user classification accuracy and recommendation product precision rate. Combining the RFMRQ model, the user-based algorithm accuracy was 2.80% higher than the item-based algorithm [8].

For recommendation systems, the cold start problem persists, and several researchers have effectively addressed it within their respective fields and algorithm frameworks. Yang Yuzhi studied a book recommendation system based on a collaborative filtering algorithm, improving the recommendation accuracy from 70% to 75% using multi-source data and a time decay model [9]. Jia Dongyan et al. proposed an interest-similar user set selection algorithm and user trust calculation model, increasing recommendation accuracy by 13.12%, performing particularly well in handling cold start users [10]. Deng Cunbin et al. integrated dynamic collaborative filtering algorithms, TimeSVD++, CNN, and LSTM models, proposing the CMLP-TimeSVD++ recommendation algorithm, primarily addressing cold start and information expiration issues [11].

Similarly, data sparsity is another challenge for collaborative filtering algorithms. Jiang Zongli et al. [12] and Cai Xiaojuan et al. [13] proposed collaborative filtering recommendation algorithms integrating user similarity and trust, improving recommendation accuracy through the transmission rules of trust relationships, showing significant advantages under data sparsity conditions. Liu Wenjia et al. [14] and Zhao Yongsheng et al. [15] improved similarity calculation and rating prediction methods, addressing data sparsity and user interest changes, and proposed an improved Pearson similarity calculation method. Jin Dan et al. [16] proposed improvements to collaborative filtering algorithms based on time and spatial dimensions, including sentiment score matrices and time weight factors, solving issues of unreliable data, sparsity, and timeliness, with recall rates over 30% higher than UNCF and UCF algorithms.

3. Related Work

3.1. Collaborative Filtering

Collaborative Filtering (CF) is a recommendation system technique used to predict content that a user might be interested in based on the user's behavior data (such as ratings, purchase history, etc.). It is mainly divided into two types: Item-Based Collaborative Filtering (Item CF) and User-Based

Collaborative Filtering (User CF). CF is widely applied in various recommendation systems and is used for anime recommendations in this paper.

Item-Based Collaborative Filtering (Item CF):

Principle: Identify other items similar to the target item and recommend content based on users' preferences for these similar items.

Steps:

Calculate the similarity between items,

$$sim(m_1, m_2) = \frac{|U(m_1) \cap U(m_2)|}{\sqrt{|U(m_1)| \cdot |U(m_2)|}} \quad (1)$$

where $U(m_1)$ and $U(m_2)$ denote the sets of users who have rated anime m_1 and m_2 , respectively, and $|U(m_1)|$ and $|U(m_2)|$ are the number of users for these two anime.

Select a group of items most similar to the target item.

Predict the user's rating for the target item based on their ratings for these similar items.

User-Based Collaborative Filtering (User CF):

Principle: Identify other users with similar interests to the target user and recommend content based on these users' preferences.

Steps:

Calculate the similarity between users, which is based on the number of common ratings,

$$sim(u, v) = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| \cdot |N(v)|}} \quad (2)$$

where $N(u)$ and $N(v)$ represent the sets of anime rated by users u and v , respectively.

Select a group of users most similar to the target user.

Recommend items based on the preferences of these similar users.

Advantages:

Does not require specific content information of items; relies on user behavior data.

Can handle various types of recommendation tasks.

Limitations:

Cold Start Problem: New users or new items lack sufficient behavior data, making effective recommendations difficult.

Sparsity Problem: Interaction data between users and items are often very sparse, leading to inaccurate similarity calculations.

3.2. ROC Analysis

In ROC analysis, precision, recall, and accuracy are important indicators for evaluating the performance of classification models. These metrics are typically used in binary classification problems but can be extended to multi-class problems. In this experiment, they are mainly used for evaluating the performance of recommendation systems.

Confusion Matrix:

		Prediction	
		Positive	Negative
Reference	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Figure 1. Confusion Matrix Illustration

Symbol Explanation:

TP (True Positive): The number of samples correctly predicted as positive.

FP (False Positive): The number of samples incorrectly predicted as positive.

FN (False Negative): The number of samples incorrectly predicted as negative.

TN (True Negative): The number of samples correctly predicted as negative.

Precision

$$precision = \frac{TP}{TP+FP} \quad (3)$$

Meaning: Precision indicates the proportion of actual positives among all samples predicted as positive. High precision means fewer false positives.

Recall

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Meaning: Recall indicates the proportion of actual positive samples that are correctly predicted as positive by the model. High recall means fewer false negatives.

Accuracy

$$Acc = \frac{TP+TN}{TP+FN+FP+TN} \quad (5)$$

Meaning: Accuracy represents the overall correctness of the model's predictions.

F1 Score

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (6)$$

Meaning: F1 Score provides an overall performance metric, especially useful when precision and recall need to be balanced.

Coverage

$$Coverage = \frac{Covered}{Total} \times 100\% \quad (7)$$

Meaning: Coverage indicates the proportion of samples covered within a specific range.

Summary:

Precision: The proportion of actual positives among the samples predicted as positive.

Recall: The proportion of actual positive samples correctly predicted as positive.

Accuracy: The proportion of correct predictions among all predictions.

F1 Score: The harmonic mean of precision and recall, considering both.

Coverage: The proportion of tested samples within the total samples.

4. Experiment

In this experiment, the dataset used is the Anime Recommendations Database. This interactive dataset consists of user ratings (1-10) for anime works. The dataset contains about 7.81 million rating

interactions, approximately 73,500 users, and about 34,500 anime titles. The anime ratings range from 1 to 10, with higher ratings indicating greater user preference for the anime. Each user bookmarks the anime before rating it. Interaction data without a rating is marked as -1. Since users do not actually interact with this portion of the data and because the proportion of -1 samples is high, resulting in coarse and unsmooth data. To prevent errors, these data points are considered invalid, and more representative data is chosen.

In this experiment, precision indicates the proportion of the recommended anime titles that users are genuinely interested in. Recall indicates the proportion of the user's genuinely interested n anime titles that the algorithm correctly identifies. Accuracy indicates the proportion of all predictions that correctly identify user preferences. Thus, precision is the primary metric for evaluating the performance of the recommendation system.

4.1. Item-Based Collaborative Filtering Recommendation Results:

The dataset is read and divided into a training set (0.75) and a test set (0.25). By calculating user ratings for anime, similar anime are found and recommended to users. The recommendation algorithm used involves weighted calculations:

$$\hat{r}_{ui} = \sum_{m \in N(i)} \text{sim}(i, m) \cdot r_{um} \quad (8)$$

where \hat{r}_{ui} is the prediction for user u on anime i, $N(i)$ is the set of users similar to anime i, $\text{sim}(i, m)$ is the similarity between anime i and m, and r_{um} is the actual rating of user u on anime m.

For the target user u, 20 similar anime are found, and 10-50 anime are recommended in five experiments. By evaluating the precision, recall, and coverage of the model, the performance of the recommendation system is measured. The performance metrics are derived from the overlap between the recommendation results and the test set.

After five experiments, the results are shown in Figure 2:

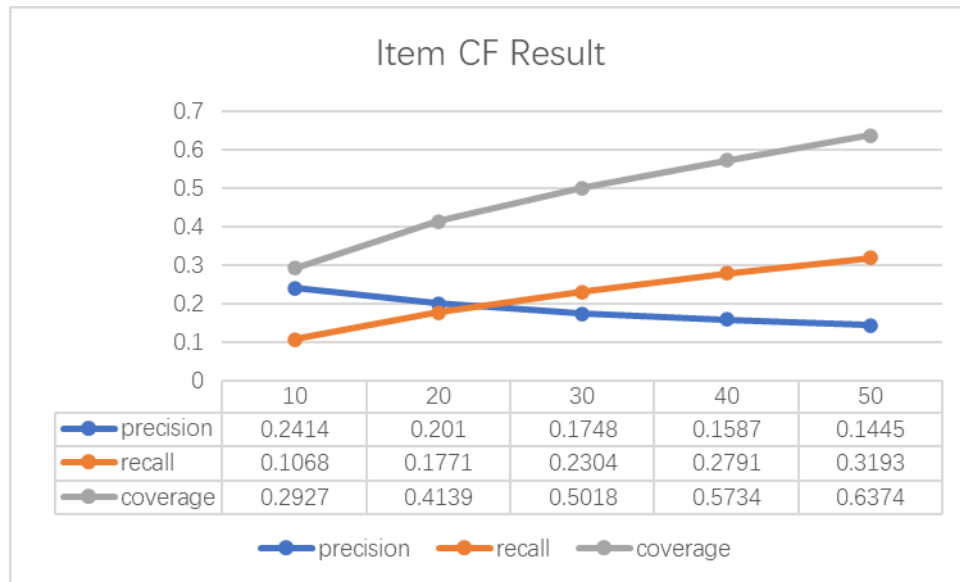


Figure 2. Item-Based Collaborative Filtering Recommendation Statistics

From the data obtained, it can be seen that recommending 10 anime to the target user results in the highest precision, reaching 0.2414, which is the most ideal. As the number of recommendations increases, the precision shows a monotonous decreasing trend. However, additional experiments found that lower numbers of recommendations result in higher precision, such as 0.2585 for 8 recommendations, 0.2786 for 6 recommendations, and 0.3048 for 4 recommendations. However, fewer

recommendations lead to higher noise, unsmooth data, and are inconvenient to apply. Ultimately, recommending 10 anime to the user is chosen.

4.2. User-Based Collaborative Filtering Recommendation Results:

First, 20 similar users are found, parameters are set to recommend 10-50 anime, five experiments are conducted, and the dataset is divided into a training set (0.75) and a test set (0.25).

The dataset is read, and by calculating user similarity, users with similar interests to the target user are found. Recommendations are made based on the preferences of these similar users. The recommendation algorithm used involves weighted calculations:

$$\hat{r}_{ui} = \sum_{v \in N(u)} \text{sim}(u, v) \cdot r_{vi} \quad (9)$$

where \hat{r}_{ui} is the predicted rating of user u on anime i , $N(u)$ is the set of users similar to u , $\text{sim}(u, v)$ is the similarity between users u and v , and r_{vi} is the actual rating of user v on anime i .

Finally, for the target user u , the most similar K users are found, and N recommendations are generated. The model's performance is measured by evaluating precision, recall, and accuracy. The performance metrics are derived from the overlap between the recommendation results and the test set.

After five experiments, the results are shown in Figure 3:

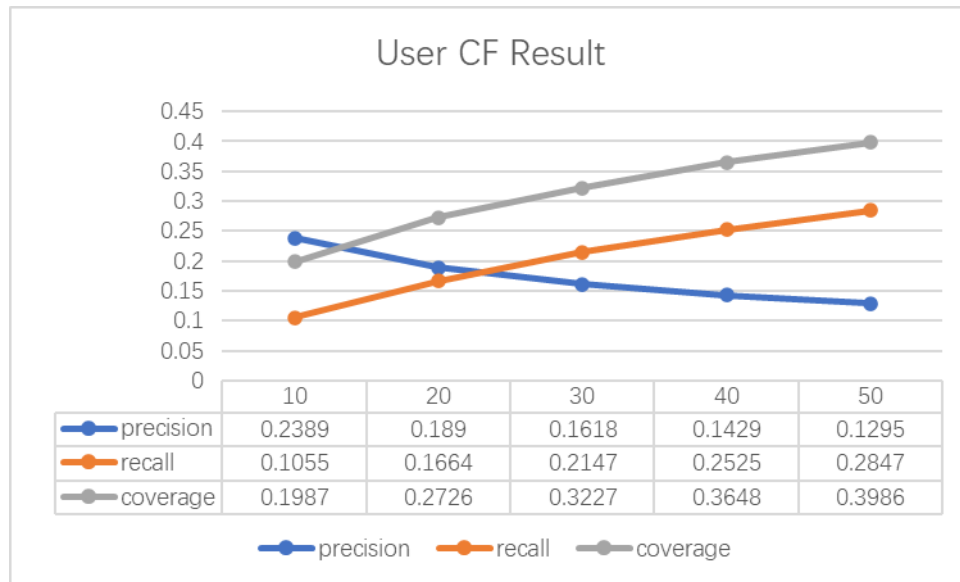


Figure 3. User-Based Collaborative Filtering Recommendation Statistics

From the data obtained, it can be seen that recommending 10 anime to the target user results in the highest precision, reaching 0.2389, which is the most ideal. As the number of recommendations increases, the precision shows a monotonous decreasing trend. Similar to the item-based results, recommending 10 anime to the user is ultimately chosen.

5. Discussion

Regarding data selection, this experiment uses an interactive dataset that includes actual interaction features (bookmarks and ratings) between users and items. Collaborative filtering struggles to effectively process non-interactive feature data. In most scenarios, non-interactive data is the easiest to obtain, as it is usually open-source and does not involve user or creator privacy. This data typically encompasses multimodal data structures, including attribute features, content features, contextual features, social relationship features, and behavioral sequence features. The presence of these non-

interactive features limits traditional collaborative filtering when dealing with complex recommendation problems.

Developing models that are compatible with various feature data structures is an important direction. To overcome these challenges, techniques such as content filtering, deep learning, data augmentation, and reinforcement learning are often combined. Additionally, using richer data and features can improve the performance and applicability of recommendation systems. Combining content filtering and model-based methods can also enhance recommendation system coverage and accuracy, addressing the data sparsity and cold start issues inherent in collaborative filtering.

6. Conclusion

This paper explores the application of collaborative filtering recommendation algorithms in anime recommendation systems, aiming to address the issues of information insufficiency and the mismatch of user personalization needs. By analyzing anime and user data, the similarities between anime and between users are calculated to achieve personalized recommendations. The experimental results demonstrate that both the item-based and user-based collaborative filtering algorithms achieve high precision when recommending 10 anime titles, indicating the effectiveness of the recommendation algorithms. Furthermore, this paper employs ROC analysis to self-evaluate the recommendation results, providing a reference for further improvements.

Collaborative filtering can provide personalized recommendations for each user based on their historical behavior and preferences, thereby enhancing user satisfaction and experience. The applicability of this experiment is broad and can be extended to different application fields such as e-commerce, social networks, and online content platforms.

References

- [1] Zhao Ming, Yan Han, Cao Gaofeng, et al. Robust recommendation algorithm based on core user extraction integrating user trust and similarity [J]. Journal of Electronics & Information Technology, 2019, 41(1): 180-186. DOI: 10.11999/JEIT180142.
- [2] Han Junbo, Li Haixia. Design and implementation of POI user game Feeds Ads recommendation system [J]. Journal of Nanyang Normal University, 2021, 20(1): 44-47. DOI: 10.3969/j.issn.1671-6132.2021.01.007.
- [3] Chen Hui, Wang Kaiyue. Probabilistic matrix factorization recommendation algorithm integrating user trust [J]. Journal of Shandong University of Technology (Natural Science Edition), 2021, 35(3): 10-16. DOI: 10.3969/j.issn.1672-6197.2021.03.003.
- [4] Zhang Li, Sun Xiaohan, Zheng Xiaohan. Collaborative filtering recommendation algorithm based on rating subspace and trust mechanism [J]. Journal of Nanjing Normal University (Engineering Technology Edition), 2023, 23(3): 27-35. DOI: 10.3969/j.issn.1672-1292.2023.03.004.
- [5] Liu Wenwen, Wang Wanyan, Cheng Shulin. Improved collaborative filtering recommendation method integrating item popularity penalty factor [J]. Computer Technology and Development, 2023, 33(3): 15-19. DOI: 10.3969/j.issn.1673-629X.2023.03.003.
- [6] Jiang Jiulei, Pan Ziyi, Li Shengqing. Neural network recommendation algorithm integrating trust [J]. Computer Applications and Software, 2023, 40(8): 274-282, 311. DOI: 10.3969/j.issn.1000-386x.2023.08.043.
- [7] Sun Ping. XML-based collaborative filtering recommendation method for English learning resources [J]. Information Technology, 2023(10): 118-122. DOI: 10.13274/j.cnki.hdzj.2023.10.022.
- [8] Deng Renfeng. Research on collaborative filtering recommendation algorithm based on RFMRQ model [J]. Textile Report, 2023, 42(12): 18-21, 86. DOI: 10.3969/j.issn.1005-6289.2023.12.005.
- [9] Yang Yuzhi. Research on library book recommendation system in colleges and universities based on collaborative filtering algorithm [J]. Science Consulting, 2023(13): 110-113.

- [10] Jia Dongyan, Zhang Fuzhi. Collaborative filtering recommendation algorithm based on double neighbor selection strategy [J]. Journal of Computer Research and Development, 2013, 50(5): 1076-1084.
- [11] Deng Cunbin, Yu Huiqun, Fan Guisheng. Recommendation algorithm integrating dynamic collaborative filtering and deep learning [J]. Computer Science, 2019, 46(8): 28-34. DOI: 10.11896/j.issn.1002-137X.2019.08.005.
- [12] Jiang Zongli, Li Hui. Collaborative filtering recommendation algorithm integrating user similarity and trust [J]. Software Guide, 2017, 16(6): 28-31. DOI: 10.11907/rjdk.162798.
- [13] Cai Xiaojuan, Tan Wen'an. An improved collaborative filtering algorithm integrating similarity and trust [J]. Computer Science, 2022, 49(z1): 238-241. DOI: 10.11896/jsjx.210400088.
- [14] Liu Wenjia, Zhang Jun. Application of an improved collaborative filtering algorithm in the movie recommendation system [J]. Modern Business Trade Industry, 2018(17): 59-62. DOI: 10.19311/j.cnki.1672-3198.2018.17.028.
- [15] Zhao Yongsheng, Qi Yunsong. Research on collaborative filtering algorithm based on improved similarity calculation method [J]. Computer and Digital Engineering, 2021, 49(3): 447-450, 541. DOI: 10.3969/j.issn.1672-9722.2021.03.005.
- [16] Jin Dan, Zhang Jiaojiao, Li Yiling, et al. Research on an improved collaborative filtering algorithm: A case study of the movie recommendation system [J]. International Business (Journal of the University of International Business and Economics), 2020(1): 128-141.