

A Collaborative Filtering Algorithm Based on Sentiment Analysis in Review Texts

Xinyu Yang¹, Ning Liu^{1,2,*}

¹College of Economics Management, Shangluo University, Shangluo, 726000, China

²Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi Province, Shangluo, 726000, China

*Corresponding author

Keywords: Sentiment analysis, review text, collaborative filtering, recommendation algorithm

Abstract: To improve the performance of collaborative filtering algorithm, a collaborative filtering algorithm based on sentiment analysis in review texts (CF_SA) is proposed in this paper. First, LDA (Latent Dirichlet Allocation) is used to form the user topic feature matrix and calculate user review similarity. Secondly, the ALBERT(A Lite Bidirectional Encoder Representation from Transformers) model and BiLSTM(Bi-directional Long Short-Term Memory) neural network are used to mine users' emotional tendencies in item review texts, improve the user rating table, and calculate user rating similarity. Next, the user review similarity and user rating similarity are combined to obtain the final user similarity and predict the user's rating for the item. Finally, experiments were conducted on the Douban Film Review dataset. Compared with classic recommendation algorithms, the results show that the proposed algorithm has good recommendation performance.

1. Introduction

With the rise and development of the Internet, people's lives have undergone tremendous changes. Various fields in life, such as shopping, healthcare, transportation, entertainment, etc., are filled with a large amount of data. Although the exponential growth of data on the internet has brought convenience to people, the massive amount of data has also increased the difficulty of obtaining effective information. Data overload makes mining effective information from massive amounts of data become a difficult problem to solve. In order to solve it, recommendation systems have emerged [1, 2].

The recommendation system analyzes user historical behavior data, understands user needs and interests, and provides users with the most likely accepted information in a timely manner. In recent years, recommendation systems have been proven to effectively solve the problem of data overload in practical scenarios, making them a hot research topic for domestic and foreign experts and scholars [3, 4].

Collaborative filtering recommendation algorithms and content-based recommendation algorithms are representatives of traditional recommendation algorithms [5, 6]. Among them, the collaborative filtering recommendation algorithm analyzes users' interest preferences through their

interaction with the system, and recommends similar items for users with similar interests; the content-based recommendation algorithm is to mine the correlation between items based on their metadata, and then recommends similar items to users based on their historical preferences.

When the amount of data in the system is too large, it is difficult for users to obtain effective data sparsely distributed in the system. To solve this problem, recommendation systems need to analyze the characteristics of user interaction behavior in multiple dimensions, obtain more relevant data through features, and then solve the problem of data sparsity. Because user review information is an important carrier reflecting user interests and preferences, domestic and foreign researchers are committed to extracting more effective data from user review information to assist in recommendations.

GUANG et al. [7] and CATALDO et al. [8] used the way of non-negative matrix decomposition and Dirichlet distribution to extract the topic distribution of reviews and the distribution of each word in the topic to achieve a more accurate recommendation method than a rating data model. However, these two methods did not take into account the dependency relationship between words in the review text. Dong et al. [9] proposed a collaborative filtering algorithm based on user interest changes and comments, which uses LDA to calculate user review topic similarity, and weights it with user rating similarity to perform collaborative filtering recommendation. WU et al. [10] designed a context aware user item representation learning model. This model dynamically learns user interaction features through two independent learning components, utilizing user review data and rating interaction data, respectively. TAY et al. [11] utilizes the review level attention mechanism to select the most important review, and then uses the word level attention mechanism to extract the feature information of important reviews. However, this model ignores low-level review information, resulting in incomplete extraction of review feature information and a decrease in recommendation accuracy. Reference [12] proposed NARRE (Neural Attentional Regression model with Review-level Explanations). This model uses the Word2Vec model to obtain hidden vectors of user and item reviews, and obtains hidden factor features through a parallel neural network. Finally, the hidden factor features and rating table are used as inputs to the hidden factor model for rating prediction. However, the words generated by Word2Vec in this model are static and do not take into account context. Due to the fact that the same word represents different information in different contexts, ignoring contextual information can lead to a deviation in the model's understanding of word semantics.

Although the above methods are committed to continuously improving the quality of recommendation algorithms, they can only extract shallow information from the review text and cannot obtain contextual dependencies within the text. Meanwhile, these methods did not effectively combine sentiment analysis and rating tables in user reviews to improve recommendation performance.

To improve the performance of collaborative filtering algorithm, a collaborative filtering algorithm based on sentiment analysis in review texts (CF_SA) is proposed in this paper. LDA (Latent Dirichlet Allocation) is used to form the user topic feature matrix and calculate the user's review similarity. The ALBERT(A Lite Bidirectional Encoder Representation from Transformers[13]) model and BiLSTM(Bi-directional Long Short-Term Memory) neural network are used to mine users' emotional tendencies in item review texts, improve the user rating table, and calculate user rating similarity. The user review similarity and user rating similarity are combined to obtain the final user similarity and predict the user's rating on the item. Finally, experiments were conducted on the Douban Film Review dataset to prove the superiority of the proposed algorithm.

2. Our Method

2.1. Algorithm Framework

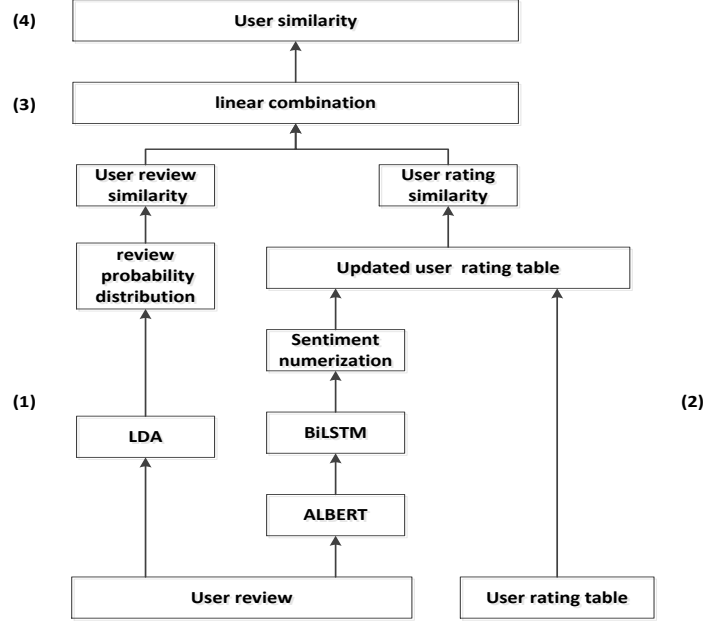


Figure 1: Algorithm Flow Chart

CF_SA proposed in this paper fully utilizes review information to mine user emotional tendencies, generate user similarity, and assist in recommendations. The flowchart of CF_SA is shown in Figure 1, which mainly includes the following parts. (1) LDA is used to mine user review information, form a user feature matrix based on topic probability, and calculate user review similarity; (2) The emotional tendencies of users in review texts are mined, and their emotional tendencies are binary classified and numerically substituted into the user rating table. The rating table is modified and updated to calculate user rating similarity; (3) The user review similarity and user rating similarity are combined to obtain the final user similarity; (4) The user similarity is used to predict items ratings for users and achieve recommendations.

2.2. Implementation Process

2.2.1. Calculate User Review Similarity

The user's review text is preprocessed by word segmentation and trained by LDA. According to the requirements of algorithm performance optimization, the number of topics is determined. The theme represents the feature information of the review text with a certain probability, forming a user theme feature matrix. Formula (1) represents the probability distribution of user u 's review on k topics.

$$\theta(D^{(u)}) = [\theta_1^{(u)}, \theta_2^{(u)}, \dots, \theta_k^{(u)}], \quad (1)$$

Where u denotes the user, $D(u)$ denotes u 's review, and k denotes the number of topics.

Based on the probability distribution of user reviews, the cosine similarity method is used to calculate the similarity of user reviews, as shown in formula (2).

$$sim_c(u, v) = \frac{\theta(D^{(u)}) \cdot \theta(D^{(v)})}{\|\theta(D^{(u)})\| \cdot \|\theta(D^{(v)})\|}, \quad (2)$$

Where $\theta(D^{(u)})$ and $\theta(D^{(v)})$ represents the probability distribution of reviews from user u and user v respectively, $sim_c(u, v)$ represents the review similarity between u and v .

2.2.2. Calculate User Rating Similarity

In order to fully utilize the implicit sentiment information in review text and enable rating data to more accurately and comprehensively represent users' true emotional tendencies. Here, we extract the sentiment information contained in user review text, quantify it, fuse it with the rating table, modify and update the rating table. The specific steps are as follows:

Firstly, the review information is input into the ALBERT layer for pre-training, which converts a review statement into multiple word vectors.

Secondly, word vectors are introduced into the BiLSTM layer to capture bidirectional semantic dependency information between word vectors through forward and backward LSTM, and analyze emotional tendencies in user review texts.

Thirdly, emotional tendencies are classified into two categories: positive emotions and negative emotions, which are introduced into the numerical layer of emotions. Emotional tendencies are numerically processed in the emotional digitization layer, as shown in formula (3).

$$score_i = \begin{cases} 0, & \text{negative} \\ 1, & \text{positive} \end{cases}. \quad (3)$$

Fourthly, the emotional numerical rating is applied to the user rating table, and the rating table is updated.

In the *ratindenotesg* table, the smaller the mean square deviation of two users' ratings for the same item is, the higher the similarity between the two users is. Therefore, the user rating mean square deviation is used to calculate user similarity, as shown in formula (4).

$$msd(u, v) = \frac{1}{|I_{u,v}|} \sum_{i \in I_{u,v}} (r_{ui} - r_{vi})^2, \quad (4)$$

Where, $I_{u,v}$ denotes a collection of items that have been evaluated by both user u and user v . $msd(u, v)$ denotes the mean square deviation of user u and user v 's ratings for the same item.

The calculation method for user rating similarity is shown in formula (5).

$$sim_r(u, v) = \frac{1}{msd(u, v) + 1}. \quad (5)$$

2.2.3. Calculate User Similarity

The user review similarity and user rating similarity are linearly combined to obtain the final user similarity. When selecting combination parameters, different data needs to be used for testing in the experiment to obtain the optimal parameters. The calculation method is as shown in formula (6).

$$sim(u, v) = b \cdot sim_c(u, v) + (1 - b) \cdot sim_r(u, v), \quad (6)$$

Where $b \in [0, 1]$, is used to adjust the proportion of user review similarity and user rating

similarity. The larger the b is, the higher the proportion in user review similarity is. The smaller b is, the higher the proportion in user rating similarity is.

2.2.4. Rating Prediction

The user similarity is sorted, and several users with the highest similarity to the target user are selected as the nearest neighbor set of the target user. The method of calculating the prediction rating in the traditional collaborative filtering algorithm is used to calculate the prediction rating of item i by user u . The calculation method is as shown in Formula (7)

$$\hat{P}(u,i) = \bar{r}_u + \frac{\sum_{v \in L_u} (r_{v_i} - \bar{r}_v) \cdot \text{sim}(u,v)}{\sum_{v \in L_u} |\text{sim}(u,v)|}, \quad (7)$$

Where L_u denotes the nearest neighbor set of user u , \bar{r}_v denotes the average rating of user v in the recommendation system, r_{v_i} denotes the rating of user v for item i , $\text{sim}(u,v)$ denotes the similarity between user u and user v .

3. Experiment

3.1. Experimental Data, Settings, and Evaluation Indicators

3.1.1. Experimental Data

The experimental data was selected from the Douban Film Comment Dataset - DMSCD Dataset, and user review data was crawled multiple times in different time periods for popular movies. The fields crawled include user ID, user name, movie ID, movie name, review information, etc. Through preliminary collation of the data, 500000 reviews from 10000 independent users on 4000 movies were extracted from the original data set.

3.1.2. Experimental Settings, and Evaluation Indicators

In the experiment, the dataset was divided into two parts in a 7:3 ratio: the training set and the test set. The data in the training set was used for calculating various steps in the algorithm and predicting movie ratings in the test set. The data in the test set is used to measure the recommendation quality of the algorithm.

Five experiments were conducted, and the average value was taken as the experimental result. Before the experiment, the data was cleaned to eliminate duplicate records, logic error records and information missing records, and the reviews texts were preprocessed such as word segmentation and stop word removal.

In the experiment, the Mean absolute error (MAE) was selected to test the accuracy of the algorithm's rating prediction. MAE represents the average value of the absolute difference between the actual value and the predicted value in the data set, which measures the average value of the residuals in the data set. The calculation formula of MAE is shown in Formula (8).

In the experiment, the effectiveness of the algorithm in TOP-N recommendation was verified, and F1-score was selected to evaluate the effectiveness of TOP-N recommendation. The calculation method for F1-score is shown in formula (9).

$$MAE = \frac{1}{m} \sum_{i=1, j=1}^{m, n} |R_{ij} - \hat{R}_{ij}| \quad (8)$$

Where m denotes the number of users, n denotes the number of items, R_{ij} represents the actual value of the rating, \hat{R}_{ij} denotes the predicted value of the rating.

$$F1-Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

Where Precision denotes the percentage of "actual purchased item quantity" and "recommended item". Recall denotes the percentage of "correctly recommended item quantity" and "actual purchased item quantity". F1-Score is the harmonic average of precision and recall, which integrates these two indicators to reflect the overall quality of recommendations. The larger the F1-Score value is, the better the recommendation effect of the model is.

3.2. Comparative Algorithms

In order to demonstrate the superiority of the algorithm CF_SA proposed in this paper, a comparison was made between the proposed algorithm and three classic recommendation algorithms.

(1) UCF [2]: UCF is traditional User-based Collaborative Filtering algorithm, which does not use external information such as reviews to assist in recommendation.

(2) TTCF [9]: TTCF is a collaborative filtering algorithm based on user interest changes and comments. It uses the Ebbinghaus Forgetting curve to collaboratively calculate the distribution and similarity of users' reviews. The similarity between user reviews and user ratings is combined to obtain the final user similarity and predict item ratings.

(3) ABFR [14]: ABFR is a recommendation algorithm based on sentiment classification of user comment texts. It integrates users' emotional tendencies into the user rating table, utilizing similarity calculation and item prediction rating to provide personalized recommendations for neighboring users.

(4) CF_SA. CF_SA is proposed in this paper. The emotional tendencies in user review texts are mined and numerically integrated into the user rating table. The rating table is revised and updated to calculate the similarity of user review. The similarity between user review and user ratings is combined to obtain the final user similarity and predict item ratings.

3.3. Experimental Results and Analysis

In the same experimental condition, the recommendation algorithm CF_SA proposed in this paper was compared with classic UCF, TTCF, and ABFR through experiments. The experiment was divided into two parts.

3.3.1. Rating Prediction Experiment

In the rating prediction experiment, the MAE values are shown in Table 1, where N denotes the recommended number of items.

It can be seen from Table 1 that compared with UCF, TTCF and ABFR, the MAE value of the proposed algorithm CF_SA is lower than that of the three comparison algorithms. It indicates that CF_SA has a low rating prediction error rate and good rating prediction performance.

This is because the proposed algorithm CF_SA can fully utilize text review information and

rating tables to assist in recommendation. It fills and updates the rating table, while also mining users' potential emotions, enabling a more realistic and accurate understanding of users' preferences, and has high rating prediction performance.

Table 1: The MAE on dataset

method	N=5	N=10	N=15	N=20	N=25	N=30	N=35	N=40
UCF	0.6341	0.6198	0.6054	0.5947	0.5981	0.5847	0.5641	0.5176
TTCF	0.6147	0.5873	0.5702	0.5831	0.5774	0.5633	0.5524	0.5098
ABFR	0.5554	0.5510	0.5321	0.4721	0.4621	0.4321	0.4231	0.4218
CF_SA	0.5210	0.5148	0.5020	0.4622	0.4441	0.4287	0.4149	0.4032

3.3.2. TOP-N Recommended Experiment

F1-Score in the recommendation results are shown in Table 2, where N denotes the recommended number of items.

Table 2: F1-Score on dataset

method	N=5	N=10	N=15	N=20	N=25	N=30	N=35	N=40
UCF	0.2214	0.3120	0.4011	0.4678	0.4812	0.5145	0.5221	0.5137
TTCF	0.2314	0.3214	0.4214	0.4699	0.4851	0.4942	0.4989	0.5172
ABFR	0.2525	0.3724	0.4421	0.4721	0.5098	0.5324	0.6240	0.6255
CF_SA	0.2675	0.3802	0.4436	0.4869	0.5158	0.5741	0.6442	0.6521

From Table 2, it can be seen that the CF_SA algorithm proposed in this paper has a significant improvement in accuracy compared to UCF, TTCF, and ABFR. This is because the CF_SA can fully utilize review information to assist in recommendation, accurately capture users' emotional tendencies in movie review text, and binarize emotional tendencies into the user's rating table for the movie. At the same time, it combines user review similarity and user rating similarity to make recommendations more accurately.

Compared to UCF, CF_SA can fully utilize review information to assist in recommendation, effectively solving the problem of data sparsity and improving recommendation performance. Compared to TTCF, CF_SA can fully explore the semantic information of review information to update the rating table, and accurately capture users' emotional tendencies in movie review texts. Compared to ABFR, CF_SA can analyze the semantic features of reviews using resources such as ratings and reviews. At the same time, CF_SA can make full use of rating table information, use a combination of user review similarity and user rating similarity to accurately reflect user preferences.

4. Conclusion

In order to further improve the superiority of the recommendation system in collaborative filtering, this paper proposes a collaborative filtering algorithm based on sentiment analysis of user reviews. The proposed algorithm uses natural language processing technology to mine the implicit information in reviews, and uses the user review similarity and user rating similarity to reflect the user's preferences, which can better improve the recommendation performance. In the future, we will also consider strengthening the calculation of similarity by analyzing users' social relationships, hoping to further improve the superiority of recommendation algorithms.

Acknowledgement

This work was supported by Natural Science Basic Research Program of Shaanxi (Program No. 2022JM-339); Science and Technology Planning Program of Shangluo City (No.2022-J-0009); Shangluo University's 2023 Innovation and Entrepreneurship Training Program for College Students.

References

- [1] Duo Lin, Yang Bing. (2020) A recommendation rating prediction method based on user interest concept lattice. *Journal of Chinese Computer Systems*, 10, 2104-2108.
- [2] Huang Liwei, Jiang Bitao, Lv Shouye, et al. (2018) Review of recommender systems based on deep learning. *Chinese Journal of Computers*, 7, 1619-1647.
- [3] Deng S, Huang L, Xu G, et al. (2017) On Deep learning for trust-aware recommendations in social networks. *IEEE Transactions on Neural Networks & Learning Systems*, 5, 1164-1177.
- [4] Zhao Yehui, Liu Lin, Wang Hailong, Han Haiyan, Pei Dongmei. (2023) Survey of Knowledge Graph Recommendation System Research. *Journal of Frontiers of Computer Science and Technology*, 4, 771-791.
- [5] Wu Sen, Dong Yaxian, Wei Guiying, Gao Xiaonan. (2022) Research on User Similarity Calculation of Collaborative Filtering for Sparse Data. *Journal of Frontiers of Computer Science and Technology*, 5, 1043-1052.
- [6] Du Y, Wang L, Peng Z, et al. (2021) Based hierarchical attention cooperative neural networks for recommendation. *Neurocomputing*, 447, 38-47.
- [7] Guang L, Michael R, Irwin K. (2014) Ratings meet reviews, a combined approach to recommend. *8th ACM Conference on Recommender Systems*. Foster City: ACM, 105-112.
- [8] Cataldo M, Claudio G, Alessandro S, et al. (2016) Ask me any rating: A content-based recommender system based on recurrent neural networks. *IIR. Proceedings of the Gustav Lorentzen Natural Working Fluids Conference*. Edinburgh: International Institute of Refrigeration, 288-296.
- [9] Dong Chenlu, Ke Xinsheng. (2018) Study on Collaborative Filtering Algorithm Based on User Interest Change and Comment. *Computer Science*, 3, 213-217.
- [10] Wu L, Quan C, Li C, et al. (2019) A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems*, 2, 1-29.
- [11] Tay Y, Luu A T, Hui S C. Multi-pointer co-attention networks for recommendation. *ACM Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London: ACM, 2018, 2309-2318.
- [12] Chen Chong, Zhang Min, Liu Yiqun, et al. (2018) Neural attentional rating regression with review-level explanations. *The 2018 World Wide Web Conference, Lyon, France*, 1583-1592.
- [13] Lan Z Z, Chen M, Goodman S, et al. (2019) ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv: 1909.11942*.
- [14] Zhu Qianqian, Lan Wenfei, Sun Hui, et al. (2022) Recommendation algorithm ABFR based on sentiment classification of user comment text. *Journal of South-Central Minzu University (Natural Science Edition)*, 41(03):333-338.