

Research Article

A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework

Yibo Wang,¹ Mingming Wang,¹ and Wei Xu ^{1,2}

¹School of Information, Renmin University of China, Beijing 100872, China

²Smart City Research Center, Renmin University of China, Beijing 100872, China

Correspondence should be addressed to Wei Xu; weixu@ruc.edu.cn

Received 2 December 2017; Accepted 3 January 2018; Published 22 March 2018

Academic Editor: Yin Zhang

Copyright © 2018 Yibo Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Movie recommendation in mobile environment is critically important for mobile users. It carries out comprehensive aggregation of user's preferences, reviews, and emotions to help them find suitable movies conveniently. However, it requires both accuracy and timeliness. In this paper, a movie recommendation framework based on a hybrid recommendation model and sentiment analysis on Spark platform is proposed to improve the accuracy and timeliness of mobile movie recommender system. In the proposed approach, we first use a hybrid recommendation method to generate a preliminary recommendation list. Then sentiment analysis is employed to optimize the list. Finally, the hybrid recommender system with sentiment analysis is implemented on Spark platform. The hybrid recommendation model with sentiment analysis outperforms the traditional models in terms of various evaluation criteria. Our proposed method makes it convenient and fast for users to obtain useful movie suggestions.

1. Introduction

The popularity of mobile devices makes people's daily lives more dependent on mobile services. People get business information, product information, promotion information, and recommendation information from mobile devices. An important application of mobile services is movie recommendation. A movie recommender system has proven to be a powerful tool on providing useful movie suggestions for users. The suggestions are provided to support the users in their effort to cope with the information overload and help them find appropriate movies fast and conveniently. Different from the demand on the personal computers (PCs), mobile services place more emphasis on timeliness, which requires fast processing and calculation from service providers. Therefore, movie recommendation in mobile services needs to be promoted in both the recommendation accuracy and the timeliness.

Movie recommendation is a comprehensive and complicated task which involves various tastes of users, various genres of movies, and so forth. Therefore, lots of techniques for recommendation have been proposed to solve the problems.

For example, content-based recommender system, collaborative filtering recommender system, and hybrid recommender system. Each technique has its own advantage in solving specific problems. Considering the usage of online information and user-generated content, collaborative filtering is supposed to be the most popular and widely deployed technique in recommender system. Collaborative filtering method recommends items by measuring the similarity between users. The similarity between users' preference can be measured by correlation calculation. In this way, users who have similar interest in movies are sorted in the same group, and then movies are recommended by their reviews and ratings of movies that they have seen. However, the correlation and similarity are difficult to calculate due to the sparsity of user's basic data, such as users' rating on movies that they have watched and their browsing history. Actually, the reviews of users on movies usually contain more information such as users' preference. Moreover, the ignorance of sentiment which users have is also a big problem in movie recommendation. At present, people are increasingly willing to post their own reviews online. In their reviews, users can express their preferences and feelings about movies.

And the feelings contained in these reviews also affect the choice of other users. Users will see the reviews, analyze their personal experience, choose their useful reviews, remove some misleading or even harmful reviews, and ultimately make their own judgments and decisions. Therefore, the sentiment in reviews is a very important aspect in evaluating a movie. Generally speaking, users are more inclined to choose the movies that the majority of people prefer and abandon the movies that the majority of people dislike. The decisions are made according to other people's experience to achieve users' own comfort experience.

With the increase of the amount of data, how to provide users with high-quality recommendations quickly among the massive information has become a serious problem. The arrival of the mobile services makes the response speed an important indicator of the user experience. The text mining and sentiment analysis techniques used to deal with user reviews aggravate the difficulty of the recommender system in the traditional environment. A new generation of recommender system needs to address how to make high-quality recommendations quickly in massive amounts of data and how to make the system highly scalable. Big data technology is one of the powerful tools to solve these problems. Some recommender systems based on Hadoop can alleviate the calculation pressure caused by the increase in the amount of data. However, in the circumstance of complex process or large number of iterations, Hadoop is not an appropriate tool because of enormous I/O access. Extremely long processing time is a critical flaw for Hadoop under the requirement of high timeliness. Fortunately, the emergence of Spark meets these needs. Different from disk-based storage of Hadoop, Spark is more inclined to save the intermediate results in memory in the calculation process, and the iterative calculation process has also been optimized. So Spark's processing efficiency is better than Hadoop in recommender systems.

In this paper, a sentiment-enhanced hybrid collaborative filtering and content-based recommendation method is proposed to recommend appropriate movies to users on Spark platform. Sentiment analysis is more reliable than simple rating, due to the fact that it contains more emotional information, which proves to be powerful in arts items such as movies. Moreover, the high efficiency of Spark makes it possible to improve the timeliness of mobile services.

The remaining sections of this paper are organized as follows: Section 2 summarizes the existing research work. The sentiment-enhanced recommendation framework is proposed in Section 3. The empirical analysis and experimental results are shown in Section 4. Finally, conclusions and future work are given in Section 5.

2. Related Work

2.1. Movie Recommender Systems. A recommender system is a program that predicts users' preferences and recommends appropriate products or services to a specific user based on users' information and products or services information. The research on recommender systems is started by GroupLens research team from the University of Minnesota. Their research object is a movie recommender system called

MovieLens. Early research is mainly focused on the content of the recommender system which analyzed the characteristics of the object itself to complete the recommendation task [1]. However, this recommendation method can only be confined to content analysis, which makes researchers and practitioners invest great efforts in designing new recommender systems. Researchers have proposed recommender systems based on collaborative filtering, association rules [2], utility, knowledge, social network [3], multiobjective programming [4], clustering [5], and other theories and techniques.

Researchers have also studied recommendations on mobile devices. Most of the research on mobile recommendation focuses on location-based services. For example, Zheng et al. utilized GPS trajectory data to solve mobile recommendation problems [6]. They proposed a user-centered collaborative location and activity filtering method based on user-location-activity relations and collaborative filtering recommendation method. On the basis of this study, Zheng et al. came up with an algorithm using ranking-based collective tensor and matrix factorization (MF) to recommend activities to users [7]. Moreover, Park et al. recommended users with restaurants using Bayesian networks based on location and some other information [8].

2.2. Content-Based Recommendation. Content-based movie recommendation methods have been widely explored in the past few years. Basu et al. proposed a content-based movie recommender system using ratings of the movies as the social information [1]. The experiments proved that their methods were more flexible and accurate. What is more, Ono et al. employed Bayesian networks to construct users' movie preference models based on their context [9]. Obviously, a variety of methods were used to excavate features of users and movies to recommend appropriate movies. In addition to use new technologies to explore features, new perspectives are also explored to build accurate profiles of users and movies. For example, Szomszor et al. introduced semantic web to analyze folksonomy hidden in the movies to help users discover appropriate movies [10]. De Pessemier et al. used social network to analyze the individual context features on users' purchasing behavior [11]. However, the design of effective profiles is always the bottleneck of content-based recommender systems. Both researchers and practitioners have made great efforts in designing a new recommendation method to avoid the shortcoming of content-based recommender systems.

2.3. Collaborative Filtering-Based Recommendation. Collaborative filtering is used to make up for the shortcomings of content-based algorithm [12]. The collaborative filtering algorithm was divided into parts for deep analysis in movie recommendation by Herlocker et al. [12]. In the process of recommendation, Koren found that users' preference changed over time, so he came up with a recommendation method using temporal dynamics to solve the problem [13]. What is more, Hofmann implemented Gaussian probabilistic latent semantic analysis in the collaborative filtering method on movie recommendation research [14]. Researchers invested

great efforts by adding new technologies to improve the performance of collaborative filtering methods on movie recommendation and they achieved good results.

Collaborative filtering is a prevalent tool used in recommender systems [15]. Marlin came up with a collaborative filtering method based on ratings [16]. Salakhutdinov and Mnih proposed a collaborative filtering method, Probabilistic Matrix Factorization, which can handle large scale of dataset [17]. At the same time, Salakhutdinov et al. employed Restricted Boltzmann Machines to improve the performance of collaborative filtering [18]. The experiment results showed that Restricted Boltzmann Machines outperformed singular value decomposition (SVD) on Netflix dataset. Moreover, Koren combined improved latent factor models and neighborhood models on Netflix dataset [19]. The latent factor model used is SVD while the neighborhood model is optimized on loss function. What is more, researchers also introduced other data mining methods to optimize the recommender systems. For example, Rendle proposed Factorization Machines (FM) which combine support vector machines (SVM) with factorization models [20]. Zhen et al. used the regularized MF used in Probabilistic Matrix Factorization (PMF) with tagging information of movies [21].

However, collaborative filtering method introduced new drawbacks in making up for some of the shortcomings of the content-based method. For example, the scalability of collaborative filtering is poor. When users produce new behavior, it is difficult for collaborative filtering to respond immediately. Therefore, both researchers and practitioners are inclined to hybridize collaborative filtering method and content-based method to solve the problem [22, 23]. For example, Debnath et al. presented a collaborative filtering and content-based movie recommender system [24]. In the content-based part of the hybrid system, the importance of the feature is expressed in a weighted manner. Nazim Uddin et al. proposed a diverse-item selection algorithm for optimizing the output of collaborative filtering method to improve the performance of hybrid recommender system [25]. Gunawardana and Meek introduced unified Boltzmann machines to hybrid collaborative filtering method and content-based method by encoding their information [26]. On the basis of integration of content-based method and collaborative filtering method, Soni et al. joined the analysis of review based text mining algorithm, making the recommendation more accurate [27]. Moreover, Ling et al. employed a rating model with a topic model based on reviews to make accurate predictions [28]. As can be seen from the above studies, the hybrid recommender system can not only improve the efficiency, but also improve the scalability of movie recommendation. Therefore, a hybrid recommendation model is an appropriate method of movie recommendation.

2.4. Sentiment Analysis. Sentiment analysis is the process of analyzing, processing, summarizing, and reasoning the emotional text [29]. Sentiment analysis began in 2002 by Pang et al.'s research [30] and has been greatly developed in the online commentary about the emotional polarity analysis. At present, the accuracy of emotional polarity analysis based on online commentary text is gradually increasing, but one of

the problems existing in emotional analysis is the lack of in-depth analysis and application of the influence of sentiment analysis.

Pang et al. used supervised learning method in machine learning to classify emotional polarity of the movie commentary text into positive one and negative one, by using the part of speech (POS) N-gram grammar (n-gram) and maximum entropy (ME) [30]. Turney implemented the unsupervised learning of machine learning to study the polarity of the text emotion [31]. He first used tags to extract the word pair from reviews and then used Pointwise Mutual Information and Information Retrieval (PMI-IR) method to calculate the similarity between the words in the text and words in the corpus to determine the emotional polarity of the text. The commentary data come from the online comment site <http://Epinions.com>. The method obtained an accuracy of 65.83% in the movie reviews dataset.

The polarity of reviews of the movies and other goods or services can be divided into positive, negative, and neutral. In general, the researchers believe that positive information has a positive effect while negative information has a negative effect [32]. Based on this conclusion, some studies introduced sentiment analysis into the user's reviews and obtained the polarity of the reviews. Then movies with most positive information were recommended to users [33]. Sun et al. came up with a sentiment-aware social media recommender system [34]. Diao et al. analyzed sentiment of reviews in collaborative filtering by applying a topic model [35].

2.5. Big Data Analytics for Recommendation. The scalability problem of recommender system also makes it harder for researchers and practitioners to provide users with convenient and efficient services. Many efforts have been taken to solve the problem [36–38]. Parallel computing is one of the most prevalent solutions. Zhou et al. built a parallel Matlab platform to implement a movie recommender system with collaborative filtering method [39]. In parallel computing, the operation efficiency of recommendation algorithms is higher than that of single machine operation. The introduction of the distributed computing framework makes the efficiency of the recommender system improve qualitatively. For example, Hadoop could help the collaborative filtering method achieve linear speedup [40, 41]. And larger datasets could get a better speedup than smaller ones [42]. Although Hadoop alleviates the scalability of recommendation algorithms to some extent, the support of MapReduce for collaborative filtering algorithms is not perfect. The reason is that collaborative filtering requires constant reading and writing of data in computation of similarities. However, Hadoop is a framework based on hard disk, and constant reading and writing of data become the bottleneck in computation. Therefore, memory based framework Spark has become a prevalent solution for recommender systems. Panigrahi et al. used Alternating Least Square (ALS) on Spark and k -means to avoid the data sparsity and scalability of collaborative filtering algorithms [43]. Wijayanto and Winarko implemented multicriteria collaborative filtering using Spark framework [44]. The experiments' results showed that efficiency of algorithms improved with the number of nodes in Spark clusters. Therefore, in order

to obtain higher computing efficiency, it is necessary to use Spark in recommender systems.

2.6. The Contribution of Our Work. As mentioned before, various recommendation models have been suggested as powerful tools for movie recommendation. Previous practitioners and academic researchers focus on the improvement of the recommendation performance by using the combination of recommendation models. However, they ignored that, with the increase of users and items to recommend, the computational overhead has heavily increased. Therefore, this paper proposes a sentiment-enhanced movie recommendation framework based on Spark platform to meet the requirement of mobile services in aspects of high timeliness. In our method, both the content-based method and the collaborative filtering method are taken into consideration. Based on collaborative filtering method and content-based method, the preliminary output is optimized by the analysis of the effect from both positive and negative information. Finally, experiments are carried out to prove the performance of our proposed method.

3. A Sentiment-Enhanced Recommendation Framework

As mentioned before, this paper uses collaborative filtering and content-based hybrid recommender systems. Collaborative filtering and content-based approaches can compensate for the shortcomings of each other, thus ensuring the accuracy and stability of the recommender system. On the one hand, collaborative filtering can make up for the lack of personalization of content-based method; on the other hand, content-based method can make up for the flaw of collaborative filtering method whose scalability is relatively weak. In general, the hybrid recommendation method is first executed based on user data and movie data to achieve a preliminary recommendation list. Then sentiment analysis is implemented to optimize the preliminary list and get the final recommendation list. Furthermore, on the basis of the hybrid recommendation framework, this paper fully considers the efficiency of the recommender system. In the process of recommending movies, this paper focuses on the user's reviews on movies. Under the influence of the herding effect, users are inclined to choose goods or services that most people prefer. Therefore, compared to movies with many negative reviews, movies with more positive reviews will be given priority to be recommended to users. After optimization, final recommendation list is generated, as shown in Figure 1.

3.1. Data Collection. In this paper we use data derived from Douban movie (<https://movie.douban.com/>) to verify the validity of the model we proposed. Douban movie data can be divided into user data, movie data, and review data. User data and movie data are used as the input of collaborative filtering method, while review data are used as the material of content-based method. As input of the model, data need preprocessing, which includes data clean, data integration, and data transformation.

3.2. The Hybrid Recommendation Module. In our proposed method, hybrid recommendation method is basic to the generation of a preliminary recommendation list. To process the hybrid method on Spark, the following steps are needed.

Step 1 (collect user preferences and item representation). Collaborative filtering method is used to discover principles from users' behavior and preferences, so how to collect the user's preferences becomes the basis of the method. Users have lots of ways to provide their own preferences for the system, such as ratings and clicks. In our proposed method, users' ratings on movies are taken into consideration. We need to preprocess the data before we import the data into the collaborative filtering model. The core of the work is normalization and reducing noise. First, noise should be filtered out because the existence of noise will result in a decrease in the efficiency and effectiveness of the recommender system. Second, the input data need normalization. By normalizing data, the method can be made more accurate.

Through the above steps, we get a two-dimensional table, in which one dimension is the user list, and the other dimension is the movie list, while the value is the user's ratings for movies. The preference data is transformed into user-movie resilient distributed datasets (RDD), which can be processed by Spark. From user's behavior and preferences we can discover some disciplines to help the following recommendation.

Due to the high timeliness requirement of mobile services, we need to improve the efficiency of the calculation. In the process of computing user preferences, the data are stored in the memory of Spark. If the calculation steps of content-based recommendation are processed after the data are written to disks, unnecessary I/O will be carried out. As a result, we tend to read data into memory and compute user preferences for collaborative filtering method and item representation for content-based method simultaneously. In our proposed method, movies are represented by their genres, directors, and actors.

Step 2 (distributed process). In order to process the data in a distributed form, Spark platform calculated the total number of items each user prefers and the total number of items that any two users prefer at the same time. The two kinds of statistics can be distributed on the computing nodes of the Spark platform and the results are stored in the form of RDD, respectively.

Step 3 (find similar users). After getting the user's preferences by analyzing users' behavior, similar users and items can be calculated based on the users' preferences.

To find similar users, similarity between users should be calculated. In this paper, we employ Euclidean distance to measure the similarity. Therefore, the similarity between users u_x, u_y , can be calculated by

$$\text{sim}(u_x, u_y) = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}}, \quad (1)$$

where x_i, y_i represent the ratings from u_x, u_y on movie i .

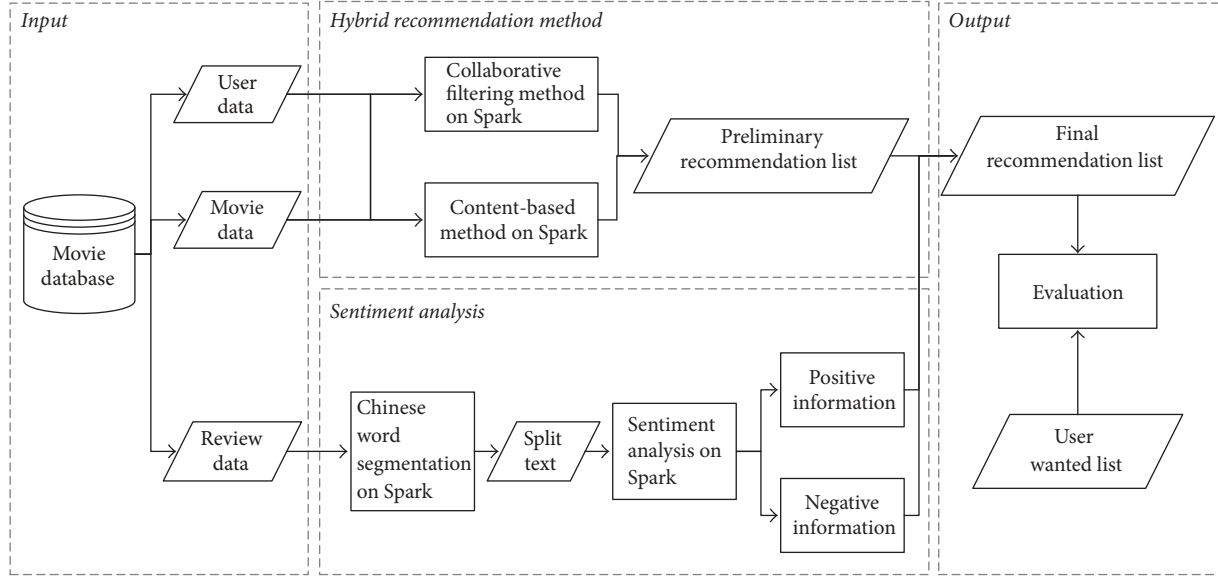


FIGURE 1: A sentiment-enhanced hybrid recommendation framework.

Step 4 (calculate and recommend). In the previous steps, all users can be ranked according to the value $\text{sim}(u_x, u_y)$. In order to recommend movies to user u_x , top K most similar users are selected. Then according to their similarities and preferences for movies, a list of recommended movies is calculated to be supplied for user u_x . Moreover, the similarities between the preference of user u_x and item representation vectors are also taken into consideration. Movies that are not suitable for the user u_x will be removed from the list. Then the list is the preliminary recommendation list to be used as the foundation of our proposed method. We calculate scores derived from the two recommendation methods.

$$\begin{aligned} \text{Score}_{\text{CF},m} &= \sum_i \text{sim}(u_t, u_i) R_{i,m} \\ \text{Score}_{\text{CB},m} &= R_{t,m} \sum_j \text{sim}(m, m_{j,t}), \end{aligned} \quad (2)$$

where $\text{Score}_{\text{CF},m}$ represents the score of movie m in collaborative filtering. $\text{sim}(u_t, u_i)$ denotes the similarity between user u_t and candidate user u_i . $R_{i,m}$ is the rating from candidate user u_i on movie m . $\text{Score}_{\text{CB},m}$ represents the score of movie m in content-based recommendation method. $\text{sim}(m, m_{j,t})$ denotes the similarity between movie m and movies which user u_t have already watched. $R_{t,m}$ is the rating from user u_t on movie m .

3.3. The Sentiment-Based Recommendation Module. First, the algorithm will encounter text information that cannot be used directly. Therefore, text mining is introduced to extract information hidden in the text data. From the point of text processing involved in this article, there is no association between different reviews, so the data can be distributed directly without special treatment.

3.3.1. Chinese Word Segmentation. Text mining is used to extract useful information from text data [45, 46]. Due to the complexity of text data, researchers have invested great efforts to seek solutions for computers to understand the meaning of text [47]. Accordingly, some methods and changes must be done to process text data. First, we employ Chinese word segmentation to solve the problem. The tool used for Chinese word segmentation is ICTCLAS [48].

The movie reviews appear in the form of long sentences in different structures. Nevertheless, in one sentence, the main information of reviews exists in several words [49]. Hence, a few key words instead of the whole sentence should be analyzed. Chinese word segmentation is the basis of text mining in Chinese. For a Chinese sentence, Chinese word segmentation is the basis for computers to recognize meanings of text [50]. Unlike English and other languages, there is no space in Chinese as a natural separator [51]. At the lexical level, Chinese word segmentation is more complicated than English word segmentation. Different segmentation may lead to different understanding of Chinese. In this paper, the Chinese reviews of movies are divided into Chinese word sequences. At the same time, stop words are excluded to avoid their negative impact on the following sentiment analysis. After the Chinese word segmentation, the rest of the words are more relevant to our study.

3.3.2. Sentiment Analysis. After the Chinese word segmentation, we analyze the result of segmentation by sentiment analysis. Finally the review is expressed as a vector space model (VSM). The VSM assumes the words that make up the text are independent of each other, so that the text can be represented by these words, which provides the basis for the representation of the mathematical model. The expression of text as a VSM can make the text representation and processing convenient. The text category is only related to

specific words contained in the text and its frequency in the text. The review D can be expressed as a vector $D = \{(t_1, w_1), (t_2, w_2) \cdots (t_n, w_n)\}$. t_i represents the i th word in the review. w_i represents the weight of t_i . In this paper, we use term frequency-inverse document frequency (TF-IDF) value as feature weights.

After the vector space representation of the movie reviews are obtained, the sentiment analysis based on the lexicon can be carried out smoothly. We first classify the reviews into positive and negative parts according to the sentiment lexicon. The lexicon is built according to the field of movies. Words such as “good” and “wonderful” in reviews indicate that the user had a positive impression of the movie. If most users have positive evaluation on the movie, the movie should be deemed a priori one to be recommended to users who have not watched it.

After analyzing and processing the sentiment words in the movie reviews and the sentiment lexicon of the corresponding categories of movie reviews, the sentiment value H is calculated, and H_m represents the sentiment value of review m .

$$H_m = \sum_{n=1}^l W_l w_l, \quad (3)$$

where W_l represents the weight of the words in the lexicon of the corresponding movie category and w_l represents the weight of the words in the vector space representation.

3.4. Ranking and Recommendation. The preliminary recommendation list based on hybrid recommendation method contains movies ranked by their scores. The scores are derived from the calculation of the collaborative filtering and content-based recommendation method, as shown in the following formula:

$$\text{Score}_{\text{hybrid},m} = \text{Score}_{\text{CF},m} + \text{Score}_{\text{CB},m}, \quad (4)$$

where $\text{Score}_{\text{hybrid},m}$ represents the score of movie m in the hybrid recommendation system.

Sentiment analysis will optimize the preliminary recommendation list. The sentiment score will be added to the score of the movie. Therefore, the score of each film is as follows:

$$\text{Score}_{\text{final},m} = W_{\text{hybrid}} \text{Score}_{\text{hybrid},m} + W_{\text{SA}} \text{Score}_{\text{SA},m}, \quad (5)$$

where W_{hybrid} and W_{SA} represent the weights of two recommendation methods and $\text{Score}_{\text{SA},m}$ represents the score of movie m derived from sentiment analysis. $\text{Score}_{\text{SA},m}$ is the sum of all H_m of reviews for the movie.

Final recommendation list is generated according to the new score. The wanted list is a group of movies with no order. To adapt to this situation, the final recommendation list will be present with no order. Therefore, in order to select enough appropriate movies, more movies are selected by hybrid recommendation method and some are discarded by final scores.

The recommender system displays the optimized list of recommendations to users. Douban movie users have

TABLE 1: The confusion matrix.

	Recommendation list	
	In the list	Not in the list
Wanted list		
In the list	TP	FN
Not in the list	FP	TN

“wanted list” which lists movies that users want to see but have not seen. Therefore, this paper uses the “wanted list” to evaluate the proposed model.

4. Empirical Analysis

4.1. Data Description. The data used in this paper is real-world data derived from Douban movie, a website that provide users with information of movies. Users can make reviews on each movie they have seen. The user-generated reviews are shown to other users who have desire to see the movie.

We ultimately get 12253 available items in the data, and each item represents a movie. As a whole, there are 6179857 reviews of these movies from 205754 users. On average, there are about 504 reviews for each movie and every user makes 30 reviews. Moreover, users’ ratings on these movies are also obtained from the website.

4.2. Evaluation Criterion. To evaluate the performance of our model, four criteria are used to evaluate the results. The criteria are derived from the confusion matrix, as shown in Table 1.

Precision and recall are contradictory to some extent, so we employ F -measure. F -measure is the weighted harmonic average of precision and recall, which can better measure the performance of the model in a more comprehensive prospect.

$$\begin{aligned} \text{TP rate} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ F1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\ \text{FP rate} &= \frac{\text{FP}}{\text{FP} + \text{TN}}. \end{aligned} \quad (6)$$

4.3. Experimental Results. The output of sentiment analysis applied on the reviews of the movies is affiliated to the evaluation of preliminary recommendation list. Sentiment analysis can optimize the candidate movie list. Therefore, the combination of collaborative filtering and content-based method with sentiment analysis makes our model performs better. For comparison, we also evaluate some recommendation method. The experimental results are shown in Table 2 and Figure 2. Our model performs better than basic recommendation in terms of TP rate, which means that our model is stronger in the ability to identify appropriate movies. CF is

TABLE 2: The performance of recommendation models.

	TP rate	FP rate	Precision	F1
CF + CB	0.645	0.355	0.531	0.582
CF + CB + SA	0.761	0.239	0.782	0.771

TABLE 3: The running time of the hybrid recommender system on Spark.

Number of nodes	Full data running time on Spark/seconds	Half data running time on Spark/seconds
(1)	463	246
(2)	276	151
(3)	197	114
(4)	153	92
(5)	101	63
(6)	86	56
(7)	75	48
(8)	65	44
(9)	58	41

short for collaborative filtering. CB represents content-based method, and SA is short for sentiment analysis.

We also compared running time on different number of nodes and different amount of data. The experimental results are shown in Table 3 and Figure 3.

First, as the number of nodes in the computational cluster increases, the computational efficiency of Spark is increasing, and the corresponding experimental result shows that the running time decreases. Second, when our model is applied in larger data, the speedup of computational efficiency is better. The results show that our proposed method performs well both in accuracy and efficiency. On the one hand, it can help merchants avoid customer churn due to delayed information and recommendation provided for mobile services users. On the other hand, it can provide help for improving the timeliness satisfaction of mobile services users.

5. Conclusions and Future Work

Mobile recommender system requires both accuracy and timeliness. In this paper, a movie recommendation framework based on hybrid recommendation and sentiment analysis is proposed to improve the accuracy of recommender systems. Furthermore, Spark is used to improve the timeliness of the system. Our proposed method makes it convenient and fast for users to obtain useful movie suggestions. Movie recommendation is a comprehensive task which involves various kinds of users and various kinds of movies. Considering the useful information hidden in reviews posted by users, collaborative filtering is considered to be the most popular and widely deployed technique in recommender system. Moreover, due to the characteristics of movie recommendation, the user watching history is very important, so we add content-based recommendation method to collaborative filtering to compose a hybrid recommender system. Moreover, it is

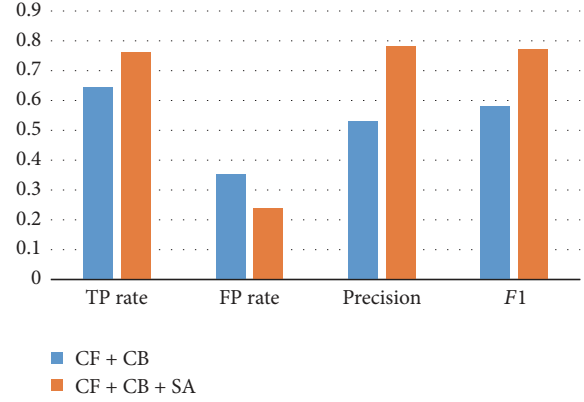


FIGURE 2: The performance of recommendation models.

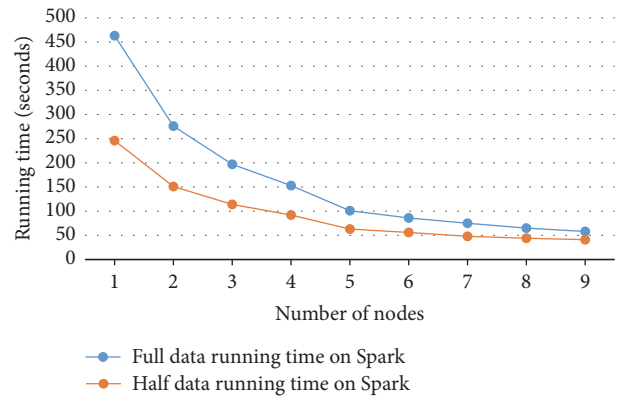


FIGURE 3: The running time of the hybrid recommender system on Spark.

better to consider the sentiment of positive and negative information during the analysis of recommender system. In general, people tend to think that positive reviews have a positive impact and negative reviews have negative effects. Sentiment analysis will help us improve the accuracy of recommendation results. Furthermore, as we illustrated in our experimental results, it is necessary to employ distributed system to solve the scalability and timeliness of recommender system.

The proposed framework can be improved in several aspects. First, this method can be verified in more data sets. Different data can be used by different sentiment analysis, so the model can be tuned to accommodate more situations. Second, in the analysis process of the sentiment analysis, different kinds of subjective ideas are involved inevitably, which implements adverse effects on the results. Therefore, future work will focus on the eliminating of individual characteristics hidden in the text description from users.

Disclosure

Mingming Wang and Wei Xu are the corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grants nos. 71301163, 71771212), Humanities and Social Sciences Foundation of the Ministry of Education (nos. 14YJA630075, 15YJA630068), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China (no. 15XNLQ08), and the Outstanding Innovative Talents Cultivation Funded Programs 2017 of Renmin University of China.

References

- [1] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," in *Proceedings of the 1998 15th National Conference on Artificial Intelligence, AAAI*, pp. 714–720, July 1998.
- [2] W. Xu, J. Wang, Z. Zhao, C. Sun, and J. Ma, "A Novel Intelligence Recommendation Model for Insurance Products with Consumer Segmentation," *Journal of Systems Science and Information*, vol. 2, no. 1, pp. 16–28, 2014.
- [3] Y. Xu, X. Guo, J. Hao, J. Ma, R. Y. K. Lau, and W. Xu, "Combining social network and semantic concept analysis for personalized academic researcher recommendation," *Decision Support Systems*, vol. 54, no. 1, pp. 564–573, 2012.
- [4] D. Guo, Z. Zhao, W. Xu et al., "How to find a comfortable bus route - Towards personalized information recommendation services," *Data Science Journal*, vol. 14, article no. 14, 2015.
- [5] D. Guo, Y. Zhu, W. Xu, S. Shang, and Z. Ding, "How to find appropriate automobile exhibition halls: Towards a personalized recommendation service for auto show," *Neurocomputing*, vol. 213, pp. 95–101, 2016.
- [6] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: a user-centered approach," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 236–241, 2010.
- [7] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: learning from GPS history data for collaborative recommendation," *Artificial Intelligence*, vol. 184/185, pp. 17–37, 2012.
- [8] M.-H. Park, J.-H. Hong, and S.-B. Cho, "Location-based recommendation system using Bayesian user's preference model in mobile devices," in *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing*, pp. 1130–1139, 2007.
- [9] C. Ono, M. Kurokawa, Y. Motomura, and H. Asoh, "A context-aware movie preference model using a bayesian network for recommendation and promotion," in *Proceedings of 11th International Conference on User Modeling*, pp. 247–257, 2007.
- [10] M. Szomszor, C. Cattuto, H. Alani et al., "Folksonomies, the semantic web, and movie recommendation," in *Proceedings of 4th European Semantic Web Conference*, pp. 1–14, 2007.
- [11] T. De Pessemier, T. Deryckere, and L. Martens, "Context aware recommendations for user-generated content on a social network site," in *Proceedings of the EuroITV'09 - 7th European Conference on European Interactive Television Conference*, pp. 133–136, Belgium, June 2009.
- [12] J. L. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 230–237, Berkeley, Calif, USA, August 1999.
- [13] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.
- [14] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 259–266, Toronto, Canada, 2003.
- [15] Y. Zhang, D. Zhang, M. M. Hassan, A. Alamri, and L. Peng, "CADRE: Cloud-Assisted Drug REcommendation Service for Online Pharmacies," *Mobile Networks and Applications*, vol. 20, no. 3, pp. 348–355, 2015.
- [16] B. Marlin, "Modeling user rating profiles for collaborative filtering," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, pp. 627–634, 2003.
- [17] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257–1264, 2007.
- [18] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, vol. 227, pp. 791–798, Corvallis, Oregon, June 2007.
- [19] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, New York, NY, USA, August 2008.
- [20] S. Rendle, "Factorization machines," in *Proceedings of the 10th IEEE International Conference on Data Mining, ICDM 2010*, pp. 995–1000, Australia, December 2010.
- [21] Y. Zhen, W.-J. Li, and D.-Y. Yeung, "TagiCoFi: Tag informed collaborative filtering," in *Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys'09*, pp. 69–76, USA, October 2009.
- [22] G. Lekakos and P. Caravelas, "A hybrid approach for movie recommendation," *Multimedia Tools and Applications*, vol. 36, no. 1-2, pp. 55–70, 2008.
- [23] Y. Zhang, Z. Tu, and Q. Wang, "TempoRec: Temporal-Topic Based Recommender for Social Network Services," *Mobile Networks and Applications*, vol. 22, pp. 1182–1191, 2017.
- [24] S. Debnath, N. Ganguly, and P. Mitra, "Feature weighting in content based recommendation system using social network analysis," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 1041–1042, Beijing, China, April 2008.
- [25] M. Nazim Uddin, J. Shrestha, and G.-S. Jo, "Enhanced content-based filtering using diverse collaborative prediction for movie recommendation," in *Proceedings of the 2009 1st Asian Conference on Intelligent Information and Database Systems, ACIIDS 2009*, pp. 132–137, Viet Nam, April 2009.
- [26] A. Gunawardana and C. Meek, "A unified approach to building hybrid recommender systems," in *Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys'09*, pp. 117–124, USA, October 2009.
- [27] K. Soni, R. Goyal, B. Vadera, and S. More, "A Three Way Hybrid Movie Recommendation System," *International Journal of Computer Applications*, vol. 160, no. 9, pp. 29–32, 2017.
- [28] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014*, pp. 105–112, USA, October 2014.
- [29] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "IDoctor: personalized and professionalized medical recommendations

- based on hybrid matrix factorization,” *Future Generation Computer Systems*, vol. 66, pp. 30–35, 2017.
- [30] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing—Volume 10 (EMNLP ’02)*, pp. 79–86, Association for Computational Linguistics, Stroudsburg, Pa, USA, July 2002.
- [31] P. D. Turney, “Thumbs up or thumbs down?” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, Philadelphia, Pennsylvania, July 2002.
- [32] J. R. Priester and R. E. Petty, “The Gradual Threshold Model of Ambivalence: Relating the Positive and Negative Bases of Attitudes to Subjective Ambivalence,” *Journal of Personality and Social Psychology*, vol. 71, no. 3, pp. 431–449, 1996.
- [33] H. Li, J. Cui, B. Shen, and J. Ma, “An intelligent movie recommendation system through group-level sentiment analysis in microblogs,” *Neurocomputing*, vol. 210, pp. 164–173, 2016.
- [34] J. Sun, G. Wang, X. Cheng, and Y. Fu, “Mining affective text to improve social media item recommendation,” *Information Processing & Management*, vol. 51, no. 4, pp. 444–457, 2015.
- [35] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, “Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS),” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’14)*, pp. 193–202, ACM, New York, NY, USA, August 2014.
- [36] Y. Zhang, “GroRec: A Group-Centric Intelligent Recommender System Integrating Social, Mobile and Big Data Technologies,” *IEEE Transactions on Services Computing*, vol. 9, no. 5, pp. 786–795, 2016.
- [37] D. Liu, W. Xu, W. Du, and F. Wang, “How to choose appropriate experts for peer review: An intelligent recommendation method in a big data context,” *Data Science Journal*, vol. 14, article no. 16, 2015.
- [38] W. Xu, J. Sun, J. Ma, and W. Du, “A personalized information recommendation system for RD project opportunity finding in big data contexts,” *Journal of Network and Computer Applications*, vol. 59, pp. 362–369, 2016.
- [39] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, “Large-scale parallel collaborative filtering for the Netflix Prize,” in *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pp. 337–348, 2008.
- [40] Z.-D. Zhao and M.-S. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010*, pp. 478–481, Thailand, January 2010.
- [41] J. Sun, W. Xu, J. Ma, and J. Sun, “Leverage RAF to find domain experts on research social network services: A big data analytics methodology with MapReduce framework,” *International Journal of Production Economics*, vol. 165, pp. 185–193, 2015.
- [42] J. Jiang, J. Lu, G. Zhang, and G. Long, “Scaling-up item-based collaborative filtering recommendation algorithm based on Hadoop,” in *Proceedings of the 7th IEEE World Congress on Services*, pp. 490–497, IEEE, Washington, DC, USA, July 2011.
- [43] S. Panigrahi, R. K. Lenka, and A. Stitipragyan, “A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark,” in *Proceedings of the 7th International Conference on Ambient Systems, Networks and Technologies, ANT 2016 and the 6th International Conference on Sustainable Energy Information Technology, SEIT 2016*, pp. 1000–1006, Spain, May 2016.
- [44] A. Wijayanto and E. Winarko, “Implementation of multi-criteria collaborative filtering on cluster using Apache Spark,” in *Proceedings of the 2nd International Conference on Science and Technology-Computer, ICST 2016*, pp. 177–181, Indonesia, October 2016.
- [45] R. Feldman and I. Dagan, “Knowledge discovery in textual databases (KDT),” in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp. 112–117, 1995.
- [46] A. H. Tan, “Text mining: the state of the art and the challenges,” in *Proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, pp. 65–70, 1999.
- [47] A. Hotho, A. Nürnberger, and G. Paaß, “A brief survey of text mining,” *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, vol. 20, pp. 19–62, 2005.
- [48] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu, “HHMM-based Chinese lexical analyzer ICTCLAS,” in *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN ’03)*, pp. 184–187, Sapporo, Japan, July 2003.
- [49] Y. Wang and W. Xu, “Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud,” *Decision Support Systems*, vol. 105, pp. 87–95, 2018.
- [50] F. Wu, Y. Huang, Y. Song, and S. Liu, “Towards building a high-quality microblog-specific Chinese sentiment lexicon,” *Decision Support Systems*, vol. 87, pp. 39–49, 2016.
- [51] Y. Wang, W. Xu, and H. Jiang, “Using text mining and clustering to group research proposals for research project selection,” in *Proceedings of the 48th Annual Hawaii International Conference on System Sciences, HICSS 2015*, pp. 1256–1263, USA, January 2015.



Hindawi

Submit your manuscripts at
www.hindawi.com

