Computing and Informatics, Vol. 42, 2023, 861–877, doi: 10.31577/cai_2023_4_861

EXPERT MINING COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM BASED ON SIGNAL FLUCTUATION

Shuo Wang, Jing Yang*, Fanshu Shang, Jingyun Sun

Harbin Engineering University
College of Computer Science and Technology
No. 145, Nantong Street, Nangang District
Harbin City, Heilongjiang Province, China
e-mail: yangjing@hrbeu.edu.cn

Abstract. This paper proposes an advanced expert collaborative filtering recommendation algorithm. Although ordinary expert system filtering algorithms have improved the recommendation accuracy of collaborative filtering technology to a certain extent, they have not screened the level of expertise of experts, and the credibility of experts varies. Therefore, this paper proposes an expert mining system based on signal fluctuations. The algorithm uses signal processing technology to filter the level of experts. This method introduces a kurtosis factor. Regarding the user's rating sequence as a random discrete signal, and then randomly sorting the user's ratings k times, the average kurtosis of the user is obtained. And take the kurtosis value as the credibility of expert users. Through experiments on multiple datasets including MovieLens, Jester, Booking-Crossings, and Last.fm, we have proved the advancement and reliability of our method.

Keywords: Recommendation system, machine learning, expert system, kurtosis, collaborative filtering

1 INTRODUCTION

With the advent of the era of big data, information technology has developed rapidly, and data has grown explosively. How to quickly and effectively obtain valuable in-

* Corresponding author

formation from complex data has become a key issue in the current development of big data. Recommendation algorithm [1] as one of the most effective means to solve the "information overload" [2], it is now not only widely used in daily life such as shopping, social networking, entertainment and other platforms, but even in medical treatment, education, etc. There is also rapid progress in the field. It can effectively use existing resources to filter out valuable information from massive data and feed it back to users. In artificial intelligence, an expert system is a computer program that replicates the ability for evaluation of a human expert. Expert systems are created to reason through knowledge bases that are mostly expressed as if-then rules, as opposed to using conventional procedural code. A knowledge base, a search or inference engine, a knowledge acquisition system, and a user interface or communication system are the four main parts of an expert system. Knowledge systems execute inference operations using explicitly stated knowledge to solve challenging challenges in the real world.

In the early phases of development, a typical recommendation approach is to simply arrange goods according to sales, topic clicks, or news reading, etc., before selecting the top N items to generate a ranking list and suggest them to users. This technique produces excellent results, and many websites still use comparable features today. However, this approach also has a serious drawback in that it can only recommend a small number of highly ranked items and cannot harvest more long-tail data. Therefore, the primary objective of study in the field of recommendation systems has become how to fully utilize current resources (items) to produce recommendations that are as accurate and thorough as feasible. At present, many papers have conducted in-depth research on recommendation systems and proposed many recommendation algorithms. Among many recommendation algorithms, collaborative filtering algorithm [3, 4, 5, 6], as the earliest and most successful recommendation technology, is the mainstream research direction in the field of recommendation systems. Its task is to use the scoring matrix of users and items to predict high-scoring items and recommend them to users. It first recommends user entry items to target users based on attributes. Recommender systems that recommend things based on consumer collaborations are the most extensively used and validated technique of giving recommendations. User-to-user collaborative filtering and item-to-item collaborative filtering are the two forms, both of which are based on user-to-user similarity. To solve some of the drawbacks of content-based filtering, collaborative filtering provides recommendations based on similarities between users and items simultaneously.

Although the traditional collaborative filtering method is effective, it is not reliable enough [7]. In social psychology, there is a principle of authority: that is, authority has a powerful force that can affect people's behavior, and people are more willing to listen to the opinions of experts. Users must accept their employees' opinion with respect and gratitude, even if they disagree, if they want to effectively impact their employees' performance utilizing the principle of authority. This demonstrates to users' staff that users are paying attention and that they are free to share their own knowledge. Therefore, many scholars have carried out research on integrating expert opinions into recommendation systems. In 2016, Hwang et al. [8] proposed a method combining category experts and collaborative filtering technology-CE method. This method selects a small number of users as experts in each category, and replaces their scores with those of ordinary neighbors' scores. Although the recommendation accuracy of collaborative filtering technology has been improved to a certain extent, this method does not screen the level of expertise of experts, and the credibility of experts varies. Therefore, this paper proposes an expert mining collaborative filtering recommendation algorithm based on signal fluctuation method, which uses signal processing technology. In general, signal fluctuation reduces system performance when compared to nonfluctuating signals. This loss is significantly reduced when there is perfect independence between subsequent signals and rather prominent when there is total correlation between signals. Section 2 of this paper introduces related work; Section 3 gives a detailed description of the method proposed in this paper. Section 4 gives the experimental settings and description of the dataset. Section 5 gives the experimental results and analysis of the experimental results. Section 6 analyzes the time complexity. Section 7 is the conclusion and future work.

2 RELATED WORK

As an important part of the recommendation system, the collaborative filtering (CF) algorithm has received extensive attention from industry and academia. The collaborative filtering algorithm is based on a strong presupposition: if it is observed that a user has consumed item A, then there is a high probability that the user will like item B similar to A, and similar users will likely like the same one entry [9]. Therefore, the core of collaborative filtering is to describe the similarity between items and users, and use the behavior of users similar to the users to be recommended to infer the preference of the users to be recommended for a particular product, and then make corresponding recommendations based on this preference. Currently, collaborative filtering algorithms mainly include two types: model-based and neighbor-based. Model-based algorithms learn prediction models from known scores, which have obvious advantages in improving prediction accuracy and coping with data sparsity. Collaborative filtering is a subset of models used in recommendation systems that examines for patterns between users or between objects using ratings or preferences that have been collected for both the person and the item. Neighborhood-based collaborative filtering algorithms, also known as memory-based algorithms, were ones of the first collaborative filtering algorithms developed [10]. Literature [11] uses a collaborative filtering recommendation method based on a deep latent factor model. This method uses deep matrix decomposition to solve the problem of recovering partially filled matrices in the collaborative filtering problem. However, it also has some shortcomings, such as the high cost of constructing the model. Literature [12] proposed a client/server framework to create a private recommendation system (PrivateRS). In the case that the ordinal meaning of the rating is significantly blurred, the method can still generate accurate recommendations with acceptable losses. This method effectively utilizes the private mode of users or items, and can to a certain extent circumvent the privacy risks caused by the mining of user preferences by the recommendation algorithm. Literature [13] proposed a fuzzy clustering collaborative filtering method (FCCF) for time-aware POI recommendation to obtain higher POI performance. A fuzzy clustering based collaborative filtering algorithm (FCCF) is proposed for time-aware POI recommendation. The fuzzy c-means technique can reduce repeated calculation and comparison and is used to group similar users. The collaborative filtering technique also provides suggestions for a number of the top-N POIs at a specific time to a target user. The above methods can provide users with appropriate recommendation results when the amount of data sets is limited, but when the amount of data sets increases, they all face scalability problems. Algorithms based on neighbors do not need to build a specific model, but use a user score matrix to calculate the similarity between users or items, so the collaborative filtering algorithm based on neighbors is easier to implement. User score matrix is used to calculate the similarity between the users or items. The recommendation system combines the similarity determined by the score value with the similarity determined by the user score probability and the type of project to increase the accuracy of the similarity between users. The neighbor-based collaborative filtering algorithm first calculates the similarity between users (products) based on the user's historical information, and then uses the evaluation of other products by neighbors with higher similarity to the target user (product) to predict the user's preference for a specific product degree. The system recommends target users based on this degree of preference. Literature [14] proposed a UBCF method based on the coverage-based rough set theory. Compared with traditional UBCF, this method adds a user reduction process, which can remove redundant users among users. Literature [15] proposed a new method of similarity measurement method, based on the attributes of items to calculate the similarity between users. In order to calculate the similarity more accurately, the user's likes and dislikes of the similar attributes of a certain item are respectively considered. When there are no users with common ratings in the similarity data set, this method can have a good recommendation effect. Literature [16] proposed a user collaborative filtering method based on fuzzy C-means. In collaborative filtering, clustering technology can be used to group the most similar users into some clusters. Fuzzy clustering is one of the most commonly used clustering techniques. Compared with other clustering methods, it has a greater improvement effect on collaborative filtering methods. Combining the center of gravity defuzzification fuzzy clustering with the Pearson correlation coefficient improves the recommendation accuracy. Literature [17] proposes a method to find nearby users through subspace clustering. In this method, the author extracts different subspaces under the categories of interest, disinterest, disinterest, and disinterest. Based on the subspace, a tree structure of neighboring users is drawn for the target user. The problem of data sparseness in the system filtering algorithm based on nearest neighbors can be slightly alleviated. Literature [18] proposed a social recommendation method based on adaptive neighbor selection mechanism on this basis. The user's initial neighborhood set is determined using this process, which combines historical ratings and social data about other users to build the user's initial neighborhood set. The initial rating of things that are invisible is predicted using these neighbor sets. A confidence model is also suggested in order to establish a new adaptive neighborhood set by removing pointless persons from the user's initial neighborhood. In order to forecast new invisible item ratings and suggest items of interest to active users, the new user-adaptive neighborhood set is used. The collaborative filtering method based on neighbors has been enhanced by the aforementioned techniques in a variety of ways, but there are still some drawbacks. For example, some models still cannot completely overcome the high dependence on user scores or the sparseness of the collaborative filtering method based on neighbors. It is difficult to find stable and reliable neighbors in the rating of user items, so the running time is long and the prediction accuracy drops sharply. The emergence of expert collaborative filtering algorithms largely compensates for this shortcoming.

3 METHODOLOGY

3.1 Problem Description

For the convenience of the following description, here is a unified description of the labels used in the text, as shown in Table1.

Symbol	Description
u, v	User u and user v
i, j	Item i and item j
I_u	A collection of items rated by user u
I_v	A collection of items rated by user v
U_i	A collection of users who rated item i
$\bar{r_u}$	The rating mean of user u
$\bar{r_v}$	The rating mean of user v
$r_{u,i}$	User u's rating for item i
$r_{v,i}$	Expert v's rating for item i
$P_{u,i}$	User u's prediction score for category c item i
$\bar{r_{u,c}}$	User u's average rating of category c items
$\bar{r_{v,c}}$	Average value of expert v's scores on category c
E_c	Experts in category c items

Table 1. Symbol and Description

Obtain the user rating matrix R in the historical purchase records, $R = r_{ui}$ $(1 \leq u \leq m, 1 \leq i \leq n)$. Where m represents the number of users and n represents the number of projects. If user A does not rate item i, then the value is 0, as shown in Formula (1).

$$r_{u,i} = \begin{cases} r_{u,i}, & \text{if } \textit{rating}, \\ 0, & \text{if } \textit{no rating}. \end{cases} \tag{1}$$

Calculate the similarity between users in the training set to form a similarity matrix. Combined with the expert algorithm, according to the item i to be scored and the item category matrix, various signal processing methods are used to screen out the experts of the category of item i. Finally, the prediction score matrix R_{pred} is formed.

3.2 Model Introduction

Collaborative filtering (CF), as an important part of the recommendation system, has received extensive attention from the industry and academia [19]. The collaborative filtering algorithm uses the behavior of users similar to the user to be recommended to infer the user's preference for a specific product, and then makes corresponding recommendations based on this preference. At present, the collaborative filtering recommendation algorithm mainly includes two types: neighbor-based and model-based. Algorithms based on nearest neighbors directly use known scores to make predictions; while model-based algorithms learn prediction models from known scores [20].

3.2.1 Nearest Neighbor Algorithm

The nearest neighbor model has the advantages of simplicity, reasonability, efficiency and stability. The common framework for predicting items on a given user is based on the nearest neighbor method [21]. The basic principle of the nearest neighbor model is to find k nearest neighbors to replace the current user. In order to solve the problem of finding neighbors, we first need to find a way to express the relationship between users. The Pearson correlation algorithm is a memory-based collaborative filtering technique which solves the scalability issue by determining how similar two user-rated items are to one another or how similar two user-rated items are to one another. The distance relationship, PCC (Pearson Correlation Coefficient) is the most commonly used measurement algorithm for collaborative filtering algorithms based on neighbors, and its calculation formula is as Formula (2):

$$\operatorname{sim}(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}. \tag{2}$$

3.2.2 Expert Algorithm

The expert algorithm, as the name suggests, divides items into categories, finds users in each category that have public reference significance for the recommendations of other target users, and defines them as experts. Expert users should meet the following definitions.

Definition 1. For project category A, expert E_c is defined by Formula (3):

$$|I_u| \leq |I_c| \left((u \in U - E_c, \forall v \in E_c \right). \tag{3}$$

The expert algorithm will satisfy the above-mentioned users to become expert users. The earliest expert algorithm consists of two parts: finding experts and generating recommended values. In actual operation, the recommendation effect of this structure is not ideal. As more and more professionals participate in research and improvement, the expert algorithm currently consists of three parts: finding experts, calculating similar values between experts and users, and generating recommended values.

Find the Experts. According to the item to be scored and the item-category matrix, determine the category of the item, calculate the number of times to evaluate all items of the category for all users, and arrange them in descending order. Descending order is a method of arranging integers from greatest to lowest. The first step in organizing the numbers is to start with the greatest number and work our way down to the smaller ones one by one. The number of experts is determined by the definition of experts and preset thresholds. Preset Threshold refers to the preset amount per Card that users have determined, up to which the Card is topped up each month and which the user may then use to consume food and beverages that month. The Preset Threshold will be supplemented each month by Available Funds.

Generate Recommended Value. In the expert algorithm of literature [14], when calculating the predictive score, only the expert suggestions with high similarity to the current user are considered, and the strategy of unconditionally trusting the expert-EA (expert algorithm) is adopted. When measuring the similarity between experts and users, Formula (4) is used, and the final score prediction uses the following Formula (4):

$$p_{u,i,c} = r_{\bar{u},c} + \frac{1}{k} \sum_{v \in E_c} (r_{v,i} - \bar{r}_{v,c}). \tag{4}$$

When the item to be predicted belongs to multiple categories, Formula (5) is used to calculate the score value.

$$p_{u,i,} = \frac{1}{|c_i|} \sum_{c \in c_i} p_{u,i,c}. \tag{5}$$

Among them: c is the number of categories to which the project belongs; P is the predicted value of each category. The running time of the above prediction scoring algorithm is relatively short, but it performs generally in terms of prediction accuracy. The difference between observed and predicted values should be used to calculate predictive accuracy. The projected values, however, may pertain to many types of knowledge. The consequent predictive accuracy can therefore be used to relate to many concepts. When the project is determined, the prediction score of this algorithm for different users is almost the same, because the expert's choice does not consider the current user, but only considers the items that the current user needs to predict.

Kurtosis Factor. Kurtosis k is a numerical statistic reflecting the distribution characteristics of random variables, a normalized 4^{th} order central moment, and a signal waveform characteristic. In mechanical principles, the kurtosis coefficient means that when fatigue failure occurs on the working surface of the bearing, the shock pulse generated at the defect of the working surface per revolution, the greater the failure, the greater the impact response amplitude, and the more obvious the failure phenomenon. Fatigue failures are closely correlated to components that withstand cyclic pressures or strains that permanently damage them. This builds up until a fracture forms, propagates, and leads to failure. Fatigue is the term used to describe the cyclic loading-induced process of damage development and failure. Shock Pulse Monitoring (SPM) is a patented predictive maintenance system that measures vibration and shock pulses of joints in motors to determine their condition and operational life before the next overhaul procedure. The kurtosis coefficient can represent the probability of the occurrence of large amplitude pulses caused by faults. The kurtosis coefficient is used to determine if a density is more or less peaked around its center than a normal curve, and negative values are frequently used to signify that a density is overstated around its center than a normal curve. This recommendation system introduces the kurtosis factor. Regarding the user's rating sequence as a random discrete signal, and then randomly sorting the user's ratings k times, the average kurtosis of the user is obtained, which can be written as Formula (6):

$$C_q = \frac{1}{k} \sum_{i=1}^{k} \frac{\frac{1}{N} \sum_{i=1}^{N} \left(|U_i| - \bar{(}r_u)\right)^4}{\bar{(}R_{u\mathrm{rms}}^{4})}. \tag{6}$$

After introducing the kurtosis factor, the method of finding experts is shown in Figure 1. The hollow blue circle on the left side of the figure represents the user, the orange hollow circle above represents the item, and the blue solid circle represents the user's rating of the item. The darker the color, the higher the score. It can be seen from the figure that the scores of different items are quite different, and they are considered experts.

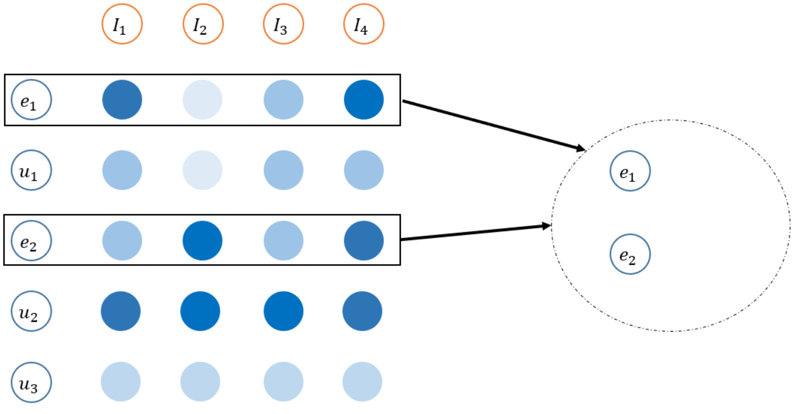

Figure 1. Schematic diagram of expert selection using kurtosis factor

4 EXPERIMENTS AND DATASETS

4.1 Datasets

MovieLens. This experimental data set uses the MovieLens data set provided by the GroupLens Research laboratory, which contains the ratings of movies by anonymous users, and each user has rated at least 20 of the movies. The rating value ranges from 1 to 5, with 1 indicating the lowest rating, 5 indicating the highest rating, and 0 indicating that the user has not rated the movie. In addition to scoring data, the data set also contains attributes of users and projects, such as the user's gender, age, occupation, project name, release year, style genre, etc. The style genre and scoring data of the user's movie are required for this experiment.

Jester. Jester was created by Ken Goldberg and his team at the University of California, Berkeley, and comprises about 6 million ratings for 150 jokes. Jester uses online user reviews to compile its ratings, just like MovieLens. Jester stands apart from other data sets in two ways: first, it has a continuous scale from -10 to 10, and second, it has the highest score density in terms of magnitude. What does "how many things are reviewed by each user" indicate in terms of the rating density? The rating density will be 100 % if each user has given a rating for every item. It will be 0 % if nothing has been rated. Jester has a 30 % density, which means that the majority of users only rated 30 % of the jokes. In example, MovieLens 1M has a density of 4.6 % (other data sets have densities of less than 1 %). Of course, it is not quite that easy. The quantity of goods that each user reviews varies. While most users only rate a small number of goods, other people rate numerous items.

Book-Crossings. Based on information from bookcrossing.com, Cai-Nicolas Ziegler developed the book score data set known as Book-Crossings. It has 1.1 million reviews from 90 000 individuals for 270 000 books. Cai-Nicolas Ziegler collected the BookCrossing (BX) dataset during a 4-week crawl (August/September 2004) with permission of Ron Hornbaker, CTO of Humankind Systems. It has 1 149 780 ratings (explicit or implicit) of 271 379 books from 278 858 individuals who have been anonymized but have demographic information. User reviews of books are gathered in the BookCrossing dataset. It has both explicit ratings (from 1 to 10 stars) and implicit ratings (based on how readers interacted with the book). The score, which includes explicit and implicit scores, is between 1 and 10. One of the least dense data sets, and the least dense data set with a score, is the Book-Crossings data collection.

Last.fm. A collection of music recommendations is provided by Last.fm. A list of the top artists and the quantity of plays are provided for each user in the data set. A tag designed to incorporate all music available on last.fm, regardless of category, and to promote a sense of community. When finished, the "all" tag should give access to everyone an incredibly diverse range of music. Additionally, it has tags that users have added that can be conducted to generate content vectors. Some information (about a particular song or the time someone is listening to the music) will be lost after the Last.fm data has been aggregated. In these examples, it is the only data set with information about the user's social network.

4.2 Statistics

We have conducted statistics on the four data sets to better understand the datasets. Statistics, also known as the "Science of Facts", allows us to generate conclusions from a set of data. Additionally, it may help people across all sectors in obtaining responses to their research or business-related queries and predict results, such as what program they might want to watch on their preferred video app next. Figure 2 shows a histogram of the number of users and the number of items in the four datasets. For the convenience of observation, we use thousands as the unit, and only part of the histogram of the Book-Crossings dataset is shown. As can be seen from the figure, the number of users of the Jester dataset is far greater than the number of items. The number of users in the Last.fm dataset is much smaller than the number of items. The number of users and items of MovieLens and Book-Crossings is relatively balanced. In addition, we can also see that the number of samples in the Book-Crossings dataset is much larger than the other three datasets.

4.3 Evaluation Metric

The criteria for evaluating the prediction accuracy of the recommendation system are divided into two categories: decision-making accuracy standards and statistical

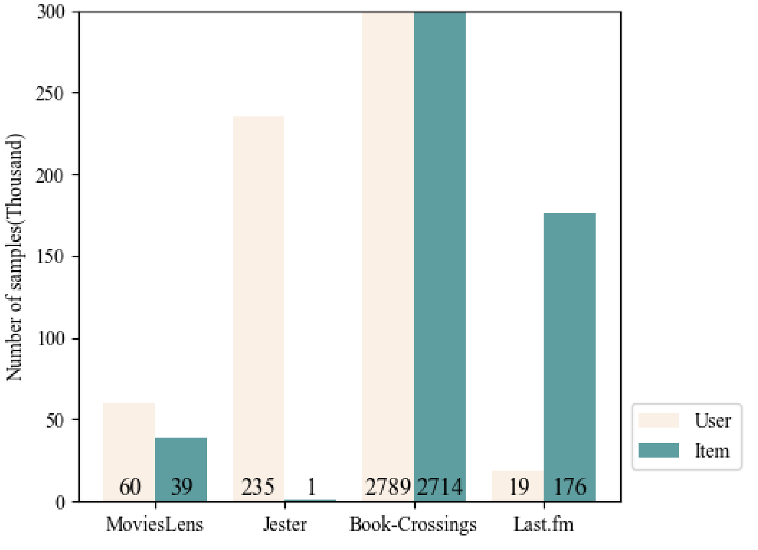

Figure 2. Number of users and items in the four datasets

accuracy standards. The actual response of an organization to a task is compared to the required response, and accuracy is calculated by allocating a cost to the difference. The organization's capacity to respond within a time frame established between the demands of the work is reflected in its ability to be responsive. A series of measurements are evaluated for accuracy to determine if they are generally accurate. This paper adopts the root mean square error (RMSE), which is sensitive to the response of very large or very small errors. The root mean square error or root mean square deviation is one of the most often employed metrics for assessing the accuracy of predictions. It illustrates the Euclidean distance between measured true and predicted values. In recommendation systems, RMSE is widely used as a common measurement error standard. The principle is to calculate the square root of the ratio of the user's projected value and the true value of the project to the square root of the ratio of the number of users n, as shown in Formula (7):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(h(x_i) - x_i)^2}. \tag{7}$$

4.4 Experiments Environment

We use Python 3.6 as the programming language for method implementation. We use Pytorch 1.7.1 to implement the neural network. We use Scikit-learn 0.24.0 to implement machine learning. Scikit-learn is a free machine learning package for

Python. It supports a number of techniques, including support vector machines, random forests, and k-neighbors, as well as the NumPy and SciPy libraries from Python. The most efficient and dependable machine learning library is Python's Scikit-learn (Sklearn) package. Through a Python interface, it provides a range of efficient techniques for statistical modeling and machine learning, including dimensionality reduction, clustering, and regression. All pre-trained models are loaded from Hugging Face Transformers. Hugging Face Transformers is a platform that offers the community APIs to access and use cutting-edge pre-trained models accessible through the Hugging Face hub. PreTrained Model is responsible for maintaining the configuration of the models and handles methods for loading, downloading, and saving models as well as a few methods that are common to all models. These methods can be used to load or save a model from a local file or directory or from a pretrained model configuration that the library distributes. The GPU used for model training is GTX1660 6G.

5 EXPERIMENTAL RESULTS AND ANALYSIS

We conduct experiments on the above five datasets to verify the advanced nature of our method. The KNN algorithm does not work well with large datasets. The cost of computing the distance between the new point and each existing point is prohibitively expensive and it also reduces the performance. The KNN technique must be used to any dataset after feature scaling (standardization and normalization). It can be seen from Table 2 that our method surpasses the KNN method on all four data sets. Among them, our method obtains the best result on the Jester dataset, that is, the root mean square error is 0.15. And our method has obtained the most effect improvement on the Last.fm data set, that is, the mean square error is reduced by 0.06. In addition, our method improves by 0.05 on the MovieLens dataset, 0.03 on the Jester dataset, 0.05 on the Book-Crossings dataset, and 0.06 on the Last.fm dataset. This fully demonstrates that our method has universal applicability on a variety of data sets.

Dataset	Method		Promote
	KNN	Ours	
MovieLens	0.23	0.18	−0.05
Jester	0.18	0.15	−0.03
BookCrossings	0.25	0.20	−0.05
Last.fm	0.22	0.16	−0.06

Table 2. Dataset, Method, and Promote

We try different k to get the best kurtosis coefficient. Figure 3 shows the experimental results obtained by using different kurtosis coefficients on four datasets. Kurtosis is observed in a symmetric distribution, and its predicted value is 3. Positive kurtosis is indicated by a kurtosis value larger than three. The range of the kurtosis value in this situation ranges from 1 to infinity. It can be seen that our method can obtain the best effect when the kurtosis coefficient is about 5.

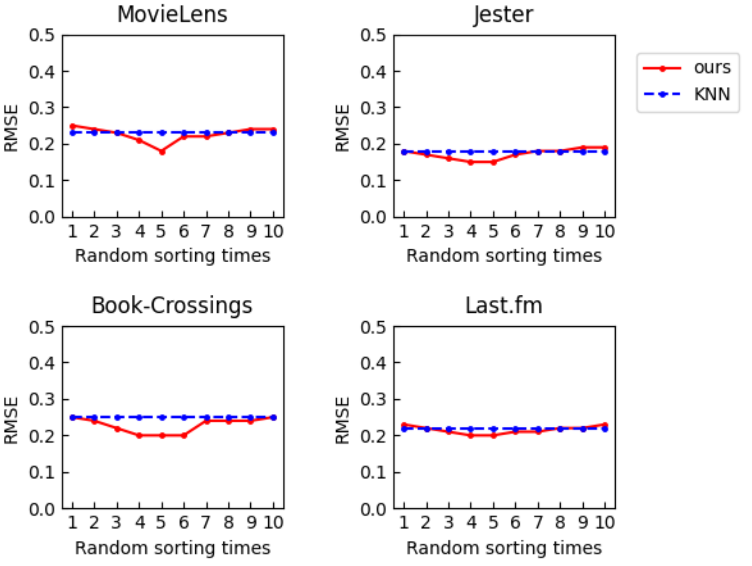

Figure 3. The effect of k on the results

6 TIME COMPLEXITY ANALYSIS

Although the method proposed in this paper can effectively improve the recommendation effect. However, the introduction of additional calculations will also increase the time complexity. In this article, the influencing factor most closely related to time complexity is the length of the user rating sequence. The term "temporal complexity" refers to how many operations an algorithm uses to complete a task (considering that each operation takes the same amount of time). The algorithm that completes the job with the fewest operations is thought to be the most effective one in terms of time complexity. Figure 4 shows the changes in the training time and test accuracy of the model as the length of the user rating sequence increases. Among them, the abscissa represents the length of the user rating sequence. The ordinate on the left represents the time required for the model to complete the entire training process. The ordinate on the right represents the accuracy of the model on the test set. The GPU used for training here is GTX1660 6G.

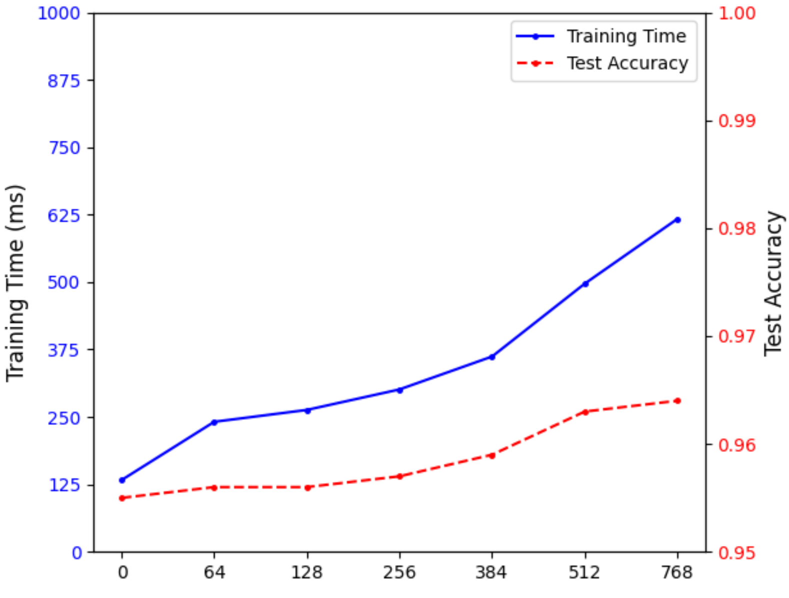

Figure 4. Time complexity graph

7 CONCLUSIONS AND FUTURE

This paper proposes a recommendation algorithm that combines expert algorithms and collaborative filtering algorithms. This method uses expert algorithms to screen out more valuable users as expert users, and uses the numerical statistics that can reflect the distribution characteristics of random variables in the mechanical field. The degree factor is used as a criterion to measure whether an expert is qualified. Collaborative filtering algorithm, as an evergreen algorithm in the field of recommendation systems, can meet the individual needs of users to a large extent, and is complementary to expert algorithms. Experiments on multiple data sets of MovieLens, Jester, Booking Crossings, Last.fm show that this method can effectively improve the recommendation accuracy and is reliable.

In the future, we will continue to study the sparseness, interpretability, and relational reasoning of recommendation systems, and devote ourselves to designing models that take into account indicators such as popularity, diversity, operational strategies, and logic. Specifically, the current algorithm we design is based on collaborative filtering, but the method proposed in this paper does not make full use of the advantages of model-based methods. Model-based methods have strong advantages in the face of data sparsity. Therefore, this article intends to incorporate the idea of using kurtosis factors to screen experts into both the nearest neighbor-based and model-based methods. We believe that this paper may effectively improve the accuracy of recommendation in the case of insufficient data.

REFERENCES

[1] Hong, H.—Kim, H. J.: Antecedents and Consequences of Information Overload in the COVID-19 Pandemic. International Journal of Environmental Research and Public Health, Vol. 17, 2020, No. 24, Art. No. 9305, doi: 10.3390/ijerph17249305.

[2] Alhijawi, B.—Al-Naymat, G.—Obeid, N.—Awajan, A.: Novel Predictive Model to Improve the Accuracy of Collaborative Filtering Recommender Systems. Information Systems, Vol. 96, 2021, Art. No. 101670, doi: 10.1016/j.is.2020.101670.

[3] Su, X.—Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, Vol. 2009, 2009, Art. No. 421425, doi: 10.1155/2009/421425.

[4] Li, Z.—Zhang, L.: Fast Neighbor User Searching for Neighborhood-Based Collaborative Filtering with Hybrid User Similarity Measures. Soft Computing, Vol. 25, 2021, No. 7, pp. 5323–5338, doi: 10.1007/s00500-020-05531-1.

[5] Huang, X.—Wang, G.: Learning Recommendation Based on Hybrid Collaborative Filtering Algorithm. Journal of Physics: Conference Series, Vol. 1629, 2020, No. 1, Art. No. 012008, doi: 10.1088/1742-6596/1629/1/012008.

[6] Wang, H.—Wu, J.: Collaborative Filtering-Based Film Recommendation Technique Utilizing Time and Film Genres. Journal of Physics: Conference Series, Vol. 1631, 2020, No. 1, Art. No. 012105, doi: 10.1088/1742-6596/1631/1/012105.

[7] Amatriain, X.—Lathia, N.—Pujol, J. M.—Kwak, H.—Oliver, N.: The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web. Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09), 2009, pp. 532–539, doi: 10.1145/1571941.1572033.

[8] Hwang, W. S.—Lee, H. J.—Kim, S. W.—Lee, M.: On Using Category Experts for Improving the Performance and Accuracy in Recommender Systems. Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), 2012, pp. 2355–2358, doi: 10.1145/2396761.2398639.

[9] Cho, J.—Kwon, K.—Park, Y.: Collaborative Filtering Using Dual Information Sources. IEEE Intelligent Systems, Vol. 22, 2007, No. 3, pp. 30–38, doi: 10.1109/MIS.2007.48.

[10] Yan, Y.—Xie, H.: Collaborative Filtering Recommendation Algorithm Based on User Preferences. Journal of Physics: Conference Series, Vol. 1549, 2020, No. 3, Art. No. 032147, doi: 10.1088/1742-6596/1549/3/032147.

[11] Hwang, W. S.—Lee, H. J.—Kim, S. W.—Won, Y.—Lee, M. S.: Efficient Recommendation Methods Using Category Experts for a Large Dataset. Information Fusion, Vol. 28, 2016, pp. 75–82, doi: 10.1016/j.inffus.2015.07.005.

[12] Liu, Q.—Cheng, B.—Xu, C.: Collaborative Filtering Based on Star Users. 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011, pp. 223–228, doi: 10.1109/ICTAI.2011.41.

[13] Pham, X. H.—Jung, J. J.—Nguyen, N. T.: Integrating Multiple Experts for Correction Process in Interactive Recommendation Systems. In: Nguyen, N. T., Hoang, K., Jędrzejowicz, P. (Eds.): Computational Collective Intelligence. Technologies and Applications (ICCCI 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7653, 2012, pp. 31–40, doi: 10.1007/978-3-642-34630-9_4.

[14] Pham, X. H.—Nguyen, T. T.—Jung, J. J.—Nguyen, N. T.: <A,V>-Spear: A New Method for Expert Based Recommendation Systems. Cybernetics and Systems, Vol. 45, 2014, No. 2, pp. 165–179, doi: 10.1080/01969722.2014.874822.

[15] Gogna, A.—Majumdar, A.: Matrix Completion Incorporating Auxiliary Information for Recommender System Design. Expert Systems with Applications, Vol. 42, 2015, No. 14, pp. 5789–5799, doi: 10.1016/j.eswa.2015.04.012.

[16] Li, Y.—Wu, H.—Li, L.—Yang, Y.—Zhang, C.—Bie, R.: Improved Field of Experts Model for Image Completion. Optik, Vol. 142, 2017, pp. 174–182, doi: 10.1016/j.ijleo.2017.05.077.

[17] Nikzad-Khasmakhi, N.—Balafar, M. A.—Feizi-Derakhshi, M. R.: The State-of-the-Art in Expert Recommendation Systems. Engineering Applications of Artificial Intelligence, Vol. 82, 2019, pp. 126–147, doi: 10.1016/j.engappai.2019.03.020.

[18] Pujahari, A.—Sisodia, D. S.: Aggregation of Preference Relations to Enhance the Ranking Quality of Collaborative Filtering Based Group Recommender System. Expert Systems with Applications, Vol. 156, 2020, Art. No. 113476, doi: 10.1016/j.eswa.2020.113476.

[19] Ortega, F.—González-Prieto, Á.: Recommender Systems and Collaborative Filtering. Applied Sciences, Vol. 10, 2020, No. 20, Art. No. 7050, doi: 10.3390/app10207050.

[20] Zhang, Z.—Zhang, Y.—Ren, Y.: Employing Neighborhood Reduction for Alleviating Sparsity and Cold Start Problems in User-Based Collaborative Filtering. Information Retrieval Journal, Vol. 23, 2020, No. 4, pp. 449–472, doi: 10.1007/s10791-020-09378-w.

[21] Srifi, M.—Oussous, A.—Ait Lahcen, A.—Mouline, S.: Recommender Systems Based on Collaborative Filtering Using Review Texts – A Survey. Information, Vol. 11, 2020, No. 6, Art. No. 317, doi: 10.3390/info11060317.

Shuo Wang is currently working toward her Ph.D. degree in computer science and technology at Harbin Engineering University, China. She majors in recommendation system and deep learning.

Jing Yang is Professor, Doctoral Supervisor in the College of Computer Science and Technology at the Harbin Engineering University, China. She obtained her Ph.D. from the College of Computer Science and Technology at Harbin Engineering University, China. Her research interests include database theories, data mining, data stream and privacy protection. She has published over 130 papers in journals and conferences.

Fanshu Shang is currently working toward her Ph.D. degree in computer science and technology at the Harbin Engineering University in China. Her research areas mainly include deep learning and privacy.

Jingyun Sun is a doctoral student majoring in computer science at the Harbin Engineering University, majoring in natural language processing.