Hindawi
Scientific Programming
Volume 2018, Article ID 5497070, 8 pages
https://doi.org/10.1155/2018/5497070

*Research Article*

# Leveraging Image Visual Features in Content-Based Recommender System

**Fuhu Deng, Panlong Ren, Zhen Qin, Gu Huang, and Zhiguang Qin**

*University of Electronic Science and Technology of China, Chengdu, China*

Correspondence should be addressed to Zhen Qin; qinzhen@uestc.edu.cn

Received 26 April 2018; Accepted 17 July 2018; Published 12 August 2018

Academic Editor: José M. Lanza-Gutiérrez

Content-based (CB) and collaborative filtering (CF) recommendation algorithms are widely used in modern e-commerce recommender systems (RSs) to improve user experience of personalized services. Item content features and user-item rating data are primarily used to train the recommendation model. However, sparse data would lead such systems unreliable. To solve the data sparsity problem, we consider that more latent information would be imported to catch users' potential preferences. Therefore, hybrid features which include all kinds of item features are used to excavate users' interests. In particular, we find that the image visual features can catch more potential preferences of users. In this paper, we leverage the combination of user-item rating data and item hybrid features to propose a novel CB recommendation model, which is suitable for rating-based recommender scenarios. The experimental results show that the proposed model has better recommendation performance in sparse data scenarios than conventional approaches. Besides, training offline and recommendation online make the model has higher efficiency on large datasets.

## 1. Introduction

With the vigorous development of the Internet, a large volume of data is generated everyday. People face an arduous task of finding optimal information which matches their preferences from such tremendous amount of information. RSs use data mining and information filtering techniques to recommend items to potential users according to their preferences [1] and have been regarded as an important tool to solve the severe problem of information overload [2]. Meanwhile, business is benefitted with a growth of sales.

Generally, RS can be categorized as three main branches—CB, CF, and hybrid algorithms [3]. CB approaches exploit the content features of items to recommend the relevant items based on the users' preferences. Most content features are textual features extracted from web pages and product descriptions or tagged by users. Unlike structured data, there are too few attributes with well-defined values [4]. Besides, CB has shortcomings of the novelty of recommendation and others [5]. CF predicts user preferences based on the known user ratings of items. However, when user ratings are scarce, it is not sufficient to reflect users' preferences. In such scenario, it is unreliable to find neighbors through user ratings. Moreover, with the increase of the interactions between users and systems, the growth of the rating matrix is extraordinarily fast. It needs to cost considerable time to find users' neighbors on the whole dataset. In addition, CF suffers from the new user cold-start problem [6]. Therefore, RSs usually combine CB and CF to overcome each other's shortcomings [7].

Most RSs typically exploit textual features in order to generate item recommendation; they usually ignore the positive effects brought by visual features. However, some researchers have achieved a huge success in the study of visual features in RSs. The paper [8] written and researched by Deldjoo et al. uses mise-en-scene visual features based on MPEG-7 and deep learning for movie recommendation. Their work in [9, 10] discusses video recommendation based on low-level features and visual features. And their work in [11, 12] discusses visual features in movie recommendation. In addition, Boutemedjet and Ziou [13] propose a graphical model for context-aware visual content recommendation.

Melo et al. [14] discuss content-based filtering enhanced by human visual attention applied to clothing recommendation. Filho et al. [15] leverage deep visual features in content-based movie recommender systems. Their previous works inspired us to use image visual features of items to explore users' potential preferences for recommendation.

When using RS-based websites, a majority of people choose items according to items' image visual features, such as colors, shapes, and textures. Besides, content features and some other features of items can reflect users' potential preferences to certain extent. Due to these reasons, in the proposed approach, we combine user ratings and hybrid features which include all kinds of item features discussed above. In particular, by transforming user-item ratings to the ratings of hybrid features, the users' potential preferences of hybrid features are used to find the nearest neighbors. In this way, the data sparsity problem and the low efficiency problem are solved. The primary contributions of this work are listed below:

(i) Image visual features are taken into consideration creatively. By transforming user-item ratings to the ratings of features, we calculate users' potential preferences of hybrid features, which include item content features, image visual features, and so on.

(ii) A novel recommendation model based on CB algorithms is proposed. It is a generic recommendation model which is suitable for rating-based recommender scenarios.

(iii) Plentiful of experiments are performed on the real-world datasets to evaluate the proposed model. The results show that our approach has better recommendation performance and higher efficiency.

The structure of this paper is organized as follows: Section 2 introduces our dataset and the feature description. A detailed explanation of the proposed model is given in Section 3. Following that, the experimental evaluation and results are shown in Section 4. Finally, we conclude the study in Section 5.

## 2. Dataset and Feature Description

In this section, it introduces the dataset we used and the description of hybrid features we used in this work.

*2.1. Dataset.* The dataset used in this paper is based on a public dataset: hetrec2011-movielens-2k [16], which is an extension of the original MovieLens 10M dataset [17], published by GroupLens Research Group at the University of Minnesota.

Table 1 shows some statistics about the hetrec2011-movielens-2k dataset and our extension. To summarize, there are 2,113 users, 10,197 movies, and 855,598 ratings ranging from 0.5 to 5.0, in increments of 0.5. There is an average of 404.921 ratings per user and 84.637 ratings per movie. The density of the dataset is 3.97%. There are a total of 13,222 unique tags, which fall into 47,957 tag assignment tuples of the form (user, tag, movie). Besides, the movies' cultural backgrounds are classified into 72 countries and 20 genres. In the dataset extension, it referenced the corresponding web pages of movies at the IMDB website [18], and 9189 movies' posters are downloaded from the URLs of IMDB. Some of the URLs have errors in accessing them so that we could not get every movies' posters. For the fairness of the experiments, we removed the corresponding movies which do not contain posters. So there are 9189 movies used in experiments in total. More information about the format and statistics of the data is available at the hetrec2011-movielens-2k website [19].

Table 1: Summary statistics of our dataset.

| | |
|---|---|
| Number of users | 2,113 |
| Number of movies | 10,197 |
| Number of ratings | 855,598 |
| Number of countries | 72 |
| Number of tags | 13,222 |
| Number of tag assignments | 47,957 |
| Number of movie genres | 20 |
| Number of movie posters | 9,189 |

*2.2. Feature Description.* In this paper, we primarily divide the hybrid features into three main parts—editorial features, user-generated features, and image visual features. Editorial features are extracted from the textual information of items, and user-generated features are extracted from the interactions between users and systems. Specifically, in the above dataset, editorial features include all kinds of content features of items, user-generated features include user tags mainly, and image visual features are image features of movie posters. Next, the details about these kinds of features will be given.

*2.2.1. Editorial Features.* In our dataset, editorial features are mainly content features of movies; specifically, movie content features consist of movie genres, movie actors, languages, and so on. As shown in Table 1, movie genres and countries are chosen as the editorial features of movies. Reasonably, movie genres have a deep influence on choosing movies to many people. Some people might take a keen interest in horror movies, while others might never watch them. Similarly, the countries of movies also have effects on people's preferences.

*2.2.2. User-Generated Features.* In our dataset, mainly, user tags are the reflection of user interactions. The movies that users have tagged can reflect that users are attracted by these movies to a certain degree. Whether the content of tags is good or bad, these tags are something relevant to the movies, especially the tags that many users have tagged. Therefore, by calculating the term frequency (TF) of each tag, we choose tags that are tagged more than 5 times by users to enrich the features of movies. In total, 1,245 tags are chosen as the user-generated features.

*2.2.3. Image Visual Features.* When users browse an item on the Internet, in most cases, the first thing that catches their attention is the picture of the item; especially for movies, sufficient potential information can be revealed from the movie posters. For example, horror movies' posters could look bleak and cold, romantic movies might have warm-toned posters. Therefore, there is a possibility that some people may choose movies by their preferences of movie posters.

However, movie posters are very complex, diversified, and heterogeneous, so that it is very difficult to extract some typical features, except for the posters' dominant color which is quite discriminating. Therefore, the dominant hue of each porter is extracted through the RGB color model. More specifically, we calculate the proportion of the different color pixels in the whole image. After that, as shown in Table 2, the movie posters are classified into 8 categories by the range of their RGB values, and each category can attract the attention of a specific kind of people.

## 3. Recommendation Model Based on Hybrid Features

In conventional CF recommendation algorithms, user-item rating data are usually used to calculate the users' similarity matrices and the nearest neighbors. However, they always suffer from the problems of data sparsity and low efficiency as described in Section 1. To solve the data sparsity problem, we consider that more latent information would be imported to build users' similarity matrices. According to CB recommendation algorithms, instead of building users' similarity matrices by user-item ratings, the proposed model uses users' potential interests of hybrid features to calculate the nearest neighbors. In particular, it transforms user-item ratings into the ratings of hybrid features and then calculates users' potential interest values of each feature to build a user profile, which is the vector representation of user interests in the feature space spanned. In this way, user preferences are reflected from many-sided features' rating data when user ratings are scarce. Moreover, when user-item rating matrix grows extremely large, the proposed model which uses hybrid features' rating matrix can greatly reduce the time consumption because the number of features is much less than items, so the hybrid features' rating matrix is much smaller than the user-item rating matrix.

As shown in Figure 1, the detailed workflow of the proposed model is divided into *Feature Interest Measure* process and *Recommendation* process. Firstly, our model input various kinds of features' matrices and user-item rating matrix. And we combine user-item rating matrix with the input features' matrices to build several hybrid features' rating matrices by converting user-item ratings into hybrid features' ratings. Next, we calculate users' potential interest values of each feature to generate a user-feature interest measure matrix. In the above *Feature Interest Measure* process, all of these calculations and transformations are completely offline. After that, in the online *Recommendation* process, the user similarity matrix and neighbor set are trained and generated based on user-feature interest measure matrix. Finally, the predicted rating values of items are calculated through the known user ratings and neighbor set.

Table 2: The categories of the movie posters.

| The range of RGB values | Poster style |
|---|---|
| $\{R \in [0, 128),\ G \in [0, 128),\ B \in [0, 128)\}$ | Dark |
| $\{R \in [0, 128),\ G \in [0, 128),\ B \in [128, 256)\}$ | Deep blue |
| $\{R \in [0, 128),\ G \in [128, 256),\ B \in [0, 128)\}$ | Green |
| $\{R \in [128, 256),\ G \in [0, 128),\ B \in [0, 128)\}$ | Red |
| $\{R \in [0, 128),\ G \in [128, 256),\ B \in [128, 256)\}$ | Light blue |
| $\{R \in [128, 256),\ G \in [0, 128),\ B \in [128, 256)\}$ | Purple |
| $\{R \in [128, 256),\ G \in [128, 256),\ B \in [0, 128)\}$ | Yellow |
| $\{R \in [128, 256),\ G \in [128, 256),\ B \in [128, 256)\}$ | White |

*3.1. Feature Interest Measure.* We have introduced our dataset in Section 2; the hybrid features include 20 movie genres, 72 movie countries, 8 movie poster styles, and 1,245 movie tags. Let $G_i\,(i = 1, 2, \dots, 20)$ represent 20 movie genres, $C_i\,(i = 1, 2, \dots, 72)$ represent 72 movie countries, $S_i\,(i = 1, 2, \dots, 8)$ represent 8 movie poster styles, and $T_i\,(i = 1, 2, \dots, 1,245)$ represent 1,245 movie tags. The format and example of the hybrid features' matrix is shown in Table 3. The value 1 means the movie includes the feature, and the value 0 means it does not. Apparently, the ratings of movies can be converted to the ratings of features. The problem is that there are much more movies than features, and the ratings of features are probably more than once in most cases. So it needs to use an exact value to reflect users' real interest of a feature as much as possible. And the example of user-feature interest measure matrix is shown in Table 4. We were inspired by a calculation method proposed by Wang et al. [20] to propose our calculation methods. Here we describe some related concepts and definitions which are used in our calculations. The specific calculation processes are as follows:

(i) Feature rating ratio: It is defined as the ratio of user $u$'s total effective rating values for feature $k$ to user $u$'s total rating values, which is expressed as $\mathrm{FRR}(u,\ k)$:

$$\mathrm{FRR}(u,\ k) = \frac{\sum_{i \in I_k \subset I_u, r \geq \sigma/2} r_{ui}}{\sum_{i \in I_u} r_{ui}}, \qquad (1)$$

where $I_u$ is the set of items rated by user $u$, $I_k$ is the set of items with feature $k$, $r_{ui}$ denotes user $u$'s rating for item $i$, and $\sigma$ denotes the maximum possible rating value in system. In the model, items with the rating larger than $\sigma/2$ are regarded as the ones that users are interested. FRR can reflect how much is user $u$ interested in feature $k$ to some extent. And the denominator of FRR can avoid the negative effect brought by different users' liveness (the amount of ratings).

(ii) Feature rating frequency ratio: It is defined as the ratio of user $u$'s total effective rating times for feature

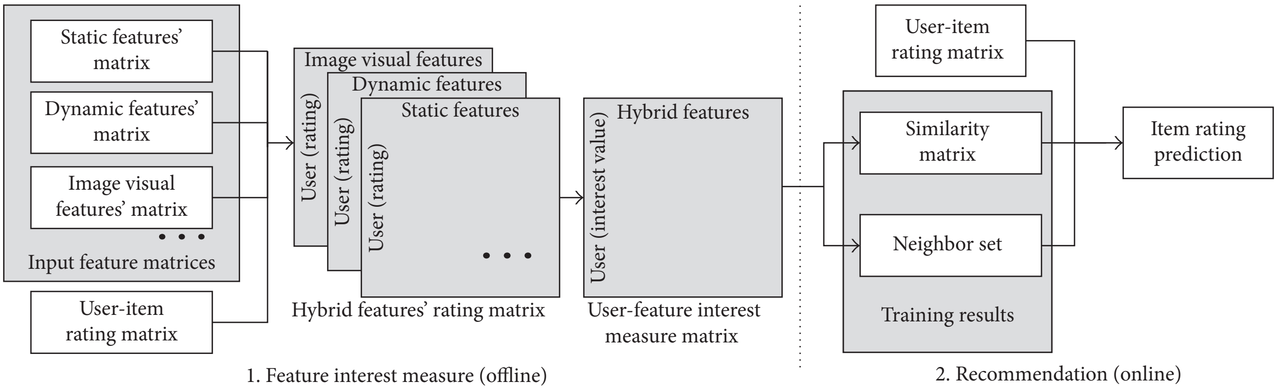


Figure 1: The workflow of the proposed model.

Table 3: Input features' matrix.

| Movie | $G_1, G_2, \ldots, G_{20}$ | $C_1, C_2, \ldots, C_{72}$ | $S_1, S_2, \ldots, S_8$ | $\cdots$ |
|---|---|---|---|---|
| $m_1$ | $1, 0, \ldots, 1$ | $1, 0, \ldots, 0$ | $1, 0, \ldots, 0$ | $\cdots$ |
| $m_2$ | $0, 1, \ldots, 0$ | $0, 0, \ldots, 1$ | $0, 0, \ldots, 1$ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $m_k$ | $1, 1, \ldots, 0$ | $0, 1, \ldots, 1$ | $0, 0, \ldots, 0$ | $\cdots$ |

Table 4: Feature interest measure matrix.

| User | $G_1, G_2, \ldots, G_{20}$ | $C_1, C_2, \ldots, C_{72}$ | $S_1, S_2, \ldots, S_8$ | $\cdots$ |
|---|---|---|---|---|
| $u_1$ | $3.5, 1, \ldots, 2.5$ | $3.5, 2, \ldots, 4.5$ | $4.5, 1, \ldots, 2$ | $\cdots$ |
| $u_2$ | $4.5, 2, \ldots, 0.5$ | $1.5, 4, \ldots, 1.5$ | $2.5, 4, \ldots, 1$ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $u_n$ | $2.5, 2, \ldots, 1.5$ | $0.5, 4, \ldots, 3.5$ | $3.5, 3, \ldots, 2$ | $\cdots$ |

$k$ to user $u$'s total rating times, which is expressed as $\mathrm{FRFR}(u, k)$:

$$\mathrm{FRFR}(u, k) = \frac{\sum_{i \in I_k \subset I_u} \delta(r_{ui})}{|T_u|},$$
$$\delta(r_{ui}) = \begin{cases} 1, & r_{ui} \geq \dfrac{\sigma}{2}, \\ 0, & r_{ui} < \dfrac{\sigma}{2}. \end{cases} \tag{2}$$

Here, user $u$'s total rating times $|T_u|$ can remove the negative effect introduced by the liveness of different users. FRFR also can reflect user $u$'s interest for feature $k$ to a certain extent.

However, FRFR calculates all effective rating times with the same weight. Actually, user preferences are also reflected by the rating values. So we modified (2) to reflect user preferences more reliable as below.

(iii) Weighted feature rating frequency ratio: It is expressed as $\mathrm{WFRFR}(u, k)$:

$$\mathrm{WFRFR}(u, k) = \frac{\sum_{i \in I_k \subset I_u} \omega_r * \delta(r_{ui})}{\sigma/2 * |T_u|}, \tag{3}$$

where

$$\omega_r = r_{ui} - \frac{\sigma}{2} + 1, \tag{4}$$

where $\omega_r$ is the weight value of $\delta(r_{ui})$, which is calculated through user rating $r_{ui}$. In this way, WFRFR can overcome the shortcomings of FRFR.

(iv) Feature interest measure: It is defined as user $u$'s interest measure value for feature $k$, which is expressed as $\mathrm{FIM}(u, k)$:

$$\mathrm{FIM}(u, k) = \frac{2\sigma * \mathrm{FRR}(u, k) * \mathrm{WFRFR}(u, k)}{\mathrm{FRR}(u, k) + \mathrm{WFRFR}(u, k)}, \tag{5}$$

where FIM is the harmonic mean value of FRR and WFRFR. The maximum possible rating value $\sigma$ is the normalization factor to make the value of FIM ranges from 0 to $\sigma$, so this calculation method is suitable for all kinds of rating-based RS.

*3.2. Recommendation.* After the generation of user-feature interest measure matrix offline, the proposed model processes the recommendation online. Firstly, as shown in (6), by employing Pearson correlation coefficient [21], the similarity of users $u$ and $v$ is calculated through their interest values of all the features, where $r_{uk}$ denotes $\mathrm{FIM}(u, k)$, which is the interest value of user $u$ for feature $k$, $\overline{r}_u$ denotes the average value of user $u$'s interest values, and $n$ is the number of features co-rated by users $u$ and $v$.

$$\mathrm{sim}(u, v) = \frac{\sum_{k=1}^{n} (r_{uk} - \overline{r}_u)(r_{vk} - \overline{r}_v)}{\sqrt{\sum_{k=1}^{n} (r_{uk} - \overline{r}_u)^2 \sum_{i=1}^{n} (r_{vk} - \overline{r}_v)^2}}. \tag{6}$$

When the similarity calculation is completed, user similarity matrix and neighbor set are generated. Based on the similarity matrix and neighbor set, the user-item rating matrix is combined to calculate the predicted ratings of items [21]. $\widehat{r}_{ui}$ denotes user $u$'s predicted rating for item $i$. The calculation method of $\widehat{r}_{ui}$ is as follows:

$$\widehat{r}_{ui} = \overline{r}_u + \frac{\sum_{v=1}^{n} \mathrm{sim}(u, v)(r_{vi} - \overline{r}_v)}{\sum_{v=1}^{n} |\mathrm{sim}(u, v)|}, \tag{7}$$

where $\overline{r}_u$ denotes the average rating of user $u$, $v$ is a neighbor user of user $u$, and there are $n$ neighbors in total. $r_{vi}$ denotes the rating of user $v$ for item $i$.

## 4. Experimental Evaluation

In this section, it describes how we evaluate the proposed model on the real dataset. In order to construct several sparse data scenarios in the evaluation, we randomly select user rating data from the full dataset described in Section 2, while keeping the rating number proportion of each user unchanged. Two methods are used to generate the sparse datasets: one is that the average user rating number changes from 10 to 80 in increments of 10; the other is that the average user rating number and user number change from 25% of full dataset to 100% in increments of 25%.

*4.1. Experimental Setup.* The experiments are performed on an operating system of Windows 10, Intel(R) Core(TM) i5-6500 CPU @ 3.20 GHz, and 8 GB RAM. Primarily, user-based K-nearest neighbors (UserKNN) algorithm [22] is chosen as the baseline. By importing the proposed model to KNN, it is called hybrid feature-based KNN (HFB-KNN) algorithm. In addition, if the proposed model only uses the content features for comparison, it is called content feature-based KNN (CFB-KNN) algorithm. To evaluate the performance of above algorithms, to avoid bias, we used 10-fold cross validation to avoid any fortunate occurrences, and the experiments are deployed as follows:

(1) Selection of user neighbor number.

To have a better performance, by varying the user neighbor number and setting user's average rating number as a fixed value (e.g., *60 ratings*), the above algorithms are evaluated to find the optimal user neighbor number.

(2) Experiments on sparse datasets.

To compare the recommendation performance, by varying the average user rating number from 10 to 80 in increments of 10, the above algorithms are evaluated on 8 sparse datasets.

(3) Comparison of recommendation time.

To compare the recommendation time, by varying the average user rating number and user number from 25% of full dataset to 100% in increments of 25%, the above algorithms are evaluated on 2 groups of different-scale datasets.

*4.2. Evaluation Metrics.* In order to verify the advance of the proposed model, mean absolute error (MAE) and root mean-squared error (RMSE) are adopted as the evaluation metrics. According to Herlocker et al. [23], by computing the value distinction between predicted values and real values, MAE and RMSE are usually used to evaluate predictive accuracy. The smaller values of the MAE and RMSE indicate a better performance in recommendation because they amplify the contributions of the absolute errors between the predictions and the true ratings. Assuming $r_i$ is the predicted rating, $\widehat{r}_i$ is the actual rating, $n$ is the total number of ratings from all users, MAE and RMSE are defined as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n} \left| r_i - \widehat{r}_i \right|, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left( r_i - \widehat{r}_i \right)^2}. \end{aligned} \tag{8}$$

### *4.3. Results*

*4.3.1. Selection of User Neighbor Number.* Figure 2 shows the results of MAE and RMSE with different numbers of user neighbors. It can be noticed that the proposed model has better rating prediction accuracy because HFB-KNN's MAE and RMSE values are always smallest in all the scenarios of different user neighbor numbers. In addition, the number of neighbor is set as 70 in the following experiments because such configuration leads to better recommendation performance.

*4.3.2. Experiments on Sparse Datasets.* Figure 3 shows the comparison results of rating prediction accuracy on 8 sparse datasets. Obviously, the prediction accuracy of 3 algorithms gets better when the average number of user rating changes from 10 to 80, and HFB-KNN outperforms both UserKNN and CFB-KNN. Moreover, when the data are quite sparse (e.g., *the average number of user rating is less than 60*), it should be noticed that the performance of HFB-KNN and CFB-KNN is dramatically better than that of UserKNN. This is because both HFB-KNN and CFB-KNN are based on the proposed model, which can enrich the feature matrix and get more latent information of user preferences in the sparse data scenarios. The reason why HFB-KNN outperforms CFB-KNN is that CFB-KNN only exploits content features; however, HFB-KNN leverages editorial features, user-generated features, and image visual features.

*4.3.3. Comparison of Recommendation Time.* Figure 4 shows the comparison of recommendation time on 2 groups of different-scale datasets. Figure 4(a) shows the scenario of varying the average user rating number from 1/4 of all user ratings to the full dataset, while Figure 4(b) shows the scenario of varying the user number from 1/4 of all users to the full dataset. We can find that all the three algorithms take more recommendation time with the increase of dataset scale. However, HFB-KNN and CFB-KNN take much less time than UserKNN in the two scenarios. Thus, it verifies the higher efficiency of the proposed model. In particular, it should be noticed that the larger scale the dataset is, the more superior in time consumption the proposed model is.

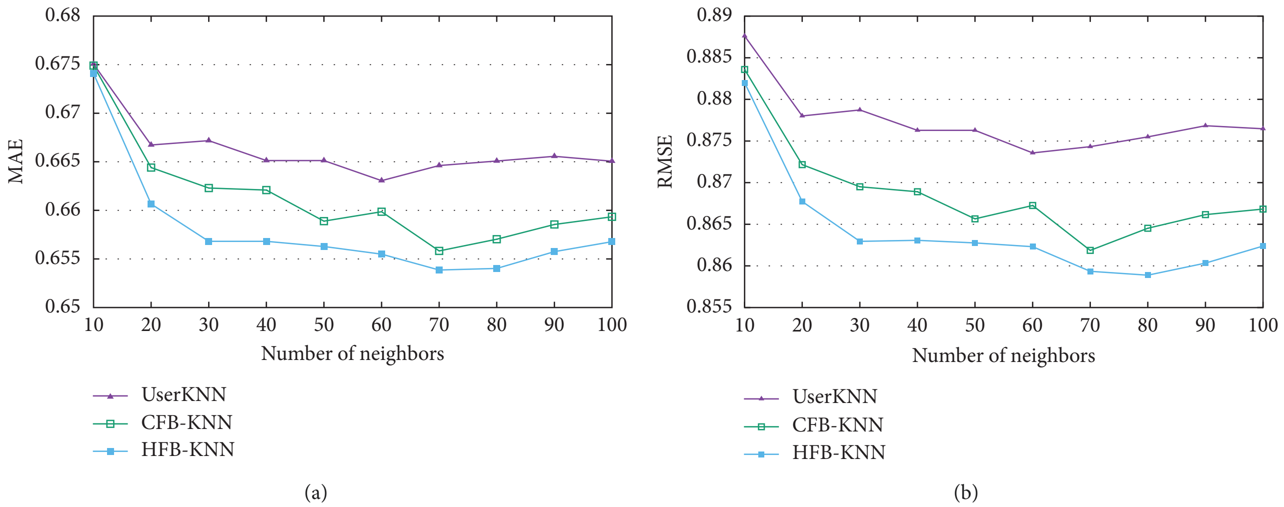


Figure 2: The comparison of rating prediction accuracy with different user neighbor numbers.

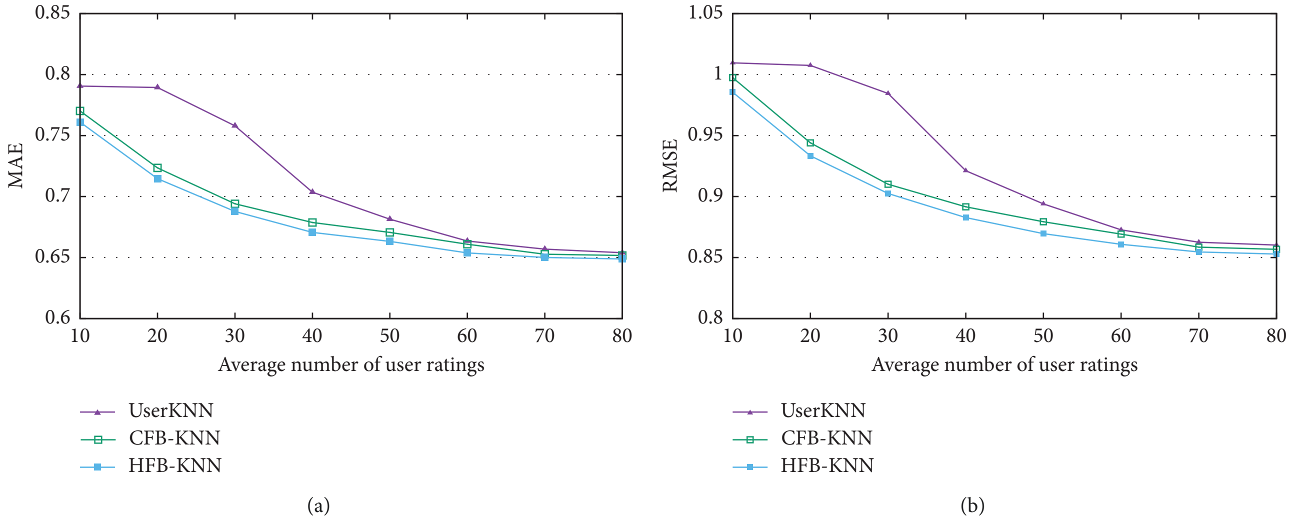


Figure 3: The comparison of rating prediction accuracy on sparse datasets.

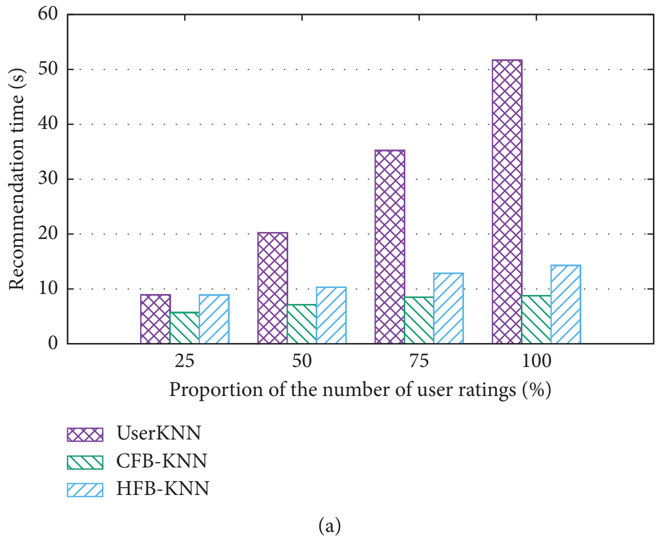
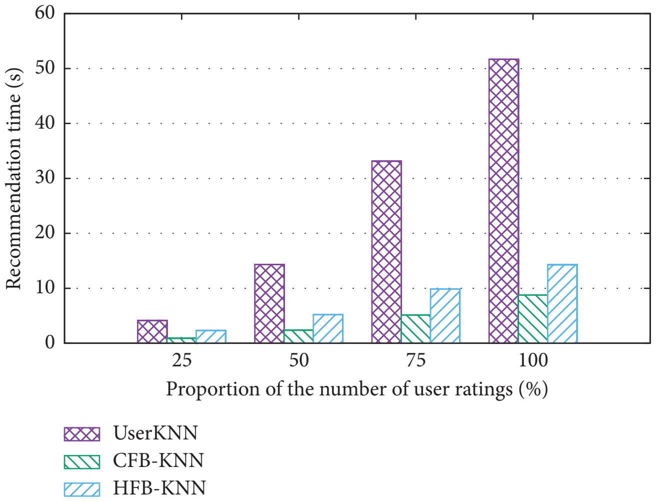


Figure 4: Continued.



(b)

Figure 4: The comparison of recommendation time. (a) Recommendation time with different user rating number. (b) Recommendation time with different user number.

## 5. Conclusion

In this paper, we proposed an approach to calculate users' potential preferences based on hybrid features. In particular, by importing editorial features, user-generated features, and image visual features of items for consideration, transforming user-item ratings into hybrid feature ratings, and calculating users' potential interest values of each feature, recommendation is processed based on the feature interest values. The experimental results showed that the proposed method had better recommendation performance on the sparse datasets and has higher efficiency on the large datasets. In future work, we will try to extract more features that can reflect users' potential preferences, especially more image visual features based on pixel information, rather than dominant hue.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. 61672135 and 61472064), the National Science Foundation of China—Guangdong Joint Foundation (No. U1401257), the Sichuan Science-Technology Support Plan Program (Nos. 2018GZ0236, 2016JZ0020, and 2017FZ0004), the Fundamental Research Funds for the Central Universities (Nos. 2672018ZYGX2018J057 and ZYGX2015KYQD136), and the Neijiang Science and Technology Incubating Project (No. 170676).

## References

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, Springer, Berlin, Germany, 2015.

[2] H. Costa and L. Macedo, "Emotion-based recommender system for overcoming the problem of information overload," in *Proceedings of International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 178–189, Salamanca, Spain, May 2013.

[3] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015.

[4] P. Lops, M. D. Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*, Springer, New York, NY, USA, 2014.

[5] M. J. Pazzani and D. Billsus, *Content-Based Recommendation Systems, in: The Adaptive Web*, Springer, Berlin, Germany, 2007.

[6] L. H. Son, "Dealing with the new user cold-start problem in recommender systems: A comparative review," *Information Systems*, vol. 58, no. 5, pp. 87–104, 2016.

[7] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[8] Y. Deldjoo, M. Quadrana, M. Elahi, and P. Cremonesi, "Using mise-en-scene visual features based on mpeg-7 and deep learning for movie recommendation," arXiv preprint arXiv: 1704.06109, 2018.

[9] Y. Deldjoo, M. Elahi, M. Quadrana, and P. Cremonesi, "Toward building a content-based video recommendation system based on low-level features," in *Proceedings of International Conference on Electronic Commerce and Web Technology*, pp. 45–56, Valencia, Spain, September 2015.

[10] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, "Content-based video recommendation system based on stylistic visual features," *Journal on Data Semantics*, vol. 5, no. 2, pp. 99–113, 2016.

[11] Y. Deldjoo, M. Elahi, P. Cremonesi, F. B. Moghaddam, and A. L. E. Caielli, "How to combine visual features with tags to improve movie recommendation accuracy?," in *Proceedings of International Conference on Electronic Commerce and Web Technologies*, pp. 34–45, Porto, Portugal, September 2016.

[12] Y. Deldjoo, M. Elahi, and P. Cremonesi, "Using visual features and latent factors for movie recommendation," in *Proceedings of ACM Recommender Systems Conference, CEUR-WS*, Boston, MA, USA, September 2016.

[13] S. Boutemedjet and D. Ziou, "A graphical model for context-aware visual content recommendation," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 52–62, 2008.

[14] E. V. D. Melo, E. A. Nogueira, and D. Guliato, "Content-based filtering enhanced by human visual attention applied to clothing recommendation," in *Proceedings of IEEE International Conference on TOOLS with Artificial Intelligence*, pp. 644–651, San Jose, CA, USA, November 2016.

[15] R. J. R. Filho, J. Wehrmann, R. C. Barros et al., "Leveraging deep visual features for content-based movie recommender systems," in *Proceedings of International Joint Conference on Neural Networks*, pp. 604–611, Anchorage, AK, USA, May 2017.

[16] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (HETREC 2011)," in *Proceedings of 5th ACM Conference on Recommender Systems (RecSys)*, Chicago, IL, USA, October 2011.

[17] Grouplens, 2018, http://www.grouplens.org.

[18] Imdb, 2018, http://www.imdb.com.

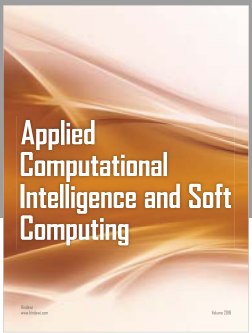
[19] Hetrec, 2011, https://grouplens.org/datasets/hetrec-2011/.

[20] Q. Wang, X. Yuan, and M. Sun, "Collaborative filtering recommendation algorithm based on hybrid user model," in *Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, pp. 1985–1990, Shandong, China, August 2010.

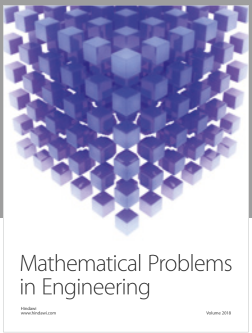
[21] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems," *Computer Science Review*, vol. 20, pp. 29–50, 2016.

[22] G. Suganeshwari and S. S. Ibrahim, *A Survey on Collaborative Filtering Based Recommendation System*, Springer International Publishing, Basel, Switzerland, 2016.

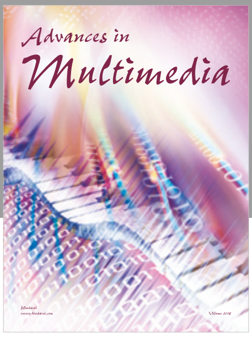
[23] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

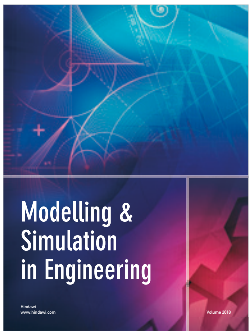

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

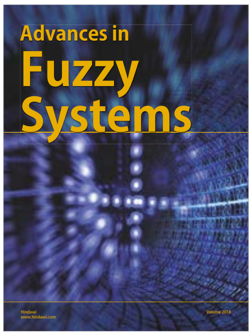
Journal of Engineering

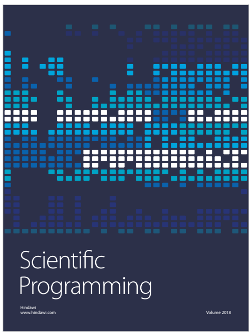

Advances in Multimedia

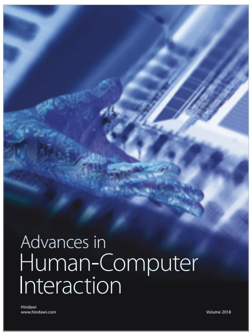

Modelling & Simulation in Engineering

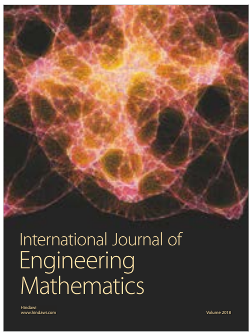
Advances in Artificial Intelligence

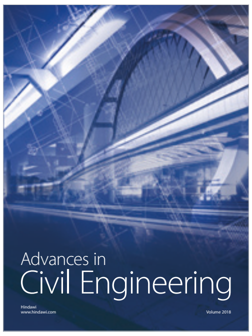
International Journal of Reconfigurable Computing

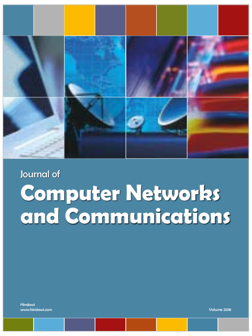

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

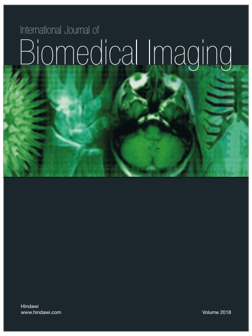

International Journal of Engineering Mathematics


Advances in Civil Engineering

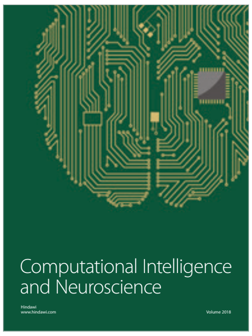

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience