International Journal of Research in Engineering, Science and Management
Volume 4, Issue 4, April 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Book Recommender System using Improved Collaborative Filtering

K. M. Kavin Chendhur[1*], V. Priya[2], R. Mohana Priya[3], S. Lavanya Lakshmi[4]
[1,2,3,4]*Department of Computer Science and Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi, India*

***Abstract*: In the day-to-day rapidly growing internet world, where the number of choices and data are abundant, there is an essential need to filter, prioritize and efficiently provide relevant information to the users in order to reduce the problem of information overload. Recommender solve this problem by searching through large volume of dynamically generated information to supply users with personalized content and services. Collaborative filtering is one of the best known and most extensive techniques in the recommendation system. Its basic idea is to predict which items the user would be interested in on the basis of their preferences. Recommendation systems using collaborative filtering can provide accurate prediction when sufficient data is provided, as this technique is based on the preference of the user. However, their precision is often comparable to increasingly confused and computationally costly calculations. To improve the execution time and accuracy of the prediction problem, this paper proposes the user preference based improved collaborative filtering approach. The system provides efficient book recommendation to the user when compared to other state of art techniques.**

***Keywords*: Book recommender system, Improved collaborative filtering, Mini batch gradient descent algorithm, Threshold values, User based collaborative filtering.**

1. Introduction

The online recommendation system has become a trend which offers a simpler and faster way to purchase items and purchases with an ease of making orders from home. Online book selling internet sites now-a-days is competing with one another by considering many attributes. A recommendation system is one of the strongest tools to increase profits and retaining buyers. User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. In this user-based filtering, the users are selected for their similarity to the active user. This similarity is determined by matching users who have posted similar ratings. Based on the previous similarity, it's presumed that future likes and dislikes also will be similar. Our Book Recommender System is based on the following observations: (1) Users who interact with items in a similar manner (for example, buying the same products or viewing the same articles) share one or more hidden preferences. (2) Users with shared preferences are likely to respond in the same way to the same items.

2. Literature Survey

[1] The paper on Books and Movies Recommendation and Rating Prediction Based on Collaborative Filtering Networks shows the outcome rundown of 10 books and films which clients may likewise been too recommended to the user by collaborative filtering.[2] Book Recommendation Based on Library Loan Records and Bibliographic Information describes in order to show the effectiveness of using (a) library loan records and (b) information about book contents as a basis for book recommendations, various data into a support vector machine (SVM) were entered and used to recommend books to subjects and evaluations of the recommendations that were given. [3] Personalized book recommendation based on ontology and collaborative filtering algorithm paper discusses the necessity of collaborative recommendation in digital library, introduces main methods and technology based on collaborative filtering recommendation system.

[4] In the work on Collaborative Filtering for Implicit Feedback Datasets, we studied collaborative filtering on datasets with implicit feedback, which is a very common situation. One of our main findings is that implicit user observations should be transformed into two paired magnitudes: preferences and confidence levels [5]. In Hybrid Book Recommender System Based on Table of Contents (ToC) and Association Rule Mining paper, they have discussed about the designing of a hybrid book recommender system which uses book contents, item-item CF approach and association rule mining to recommend more accurate and relevant books meeting the needs of the book reader.

[6] The Literature Survey on Collaborative filtering based online recommendation system analyzes the recommendation systems based on collaborative filtering and the two techniques namely item-based and user-based approach. It also presents a survey of various state of the art techniques for recommendation systems and highlights the best techniques to generate accurate results [7]. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering focuses to solve the problems of scalability and

*Corresponding author: kavin.chendhur@gmail.com

sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology. [9] The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one.

[8] The research paper on Movie recommendation system using collaborative filtering and k-means the purpose of this research is to develop a movie recommender system using collaborative filtering technique and k-means. This paper considers the users m (m is the number of users), points in n dimensional space (n is the number of items) and it presents an approach based on user clustering to produce a recommendation for the active user by a new approach.[9] Recommendation system using Collaborative filtering, this literature and their system improves the execution time and accuracy of the prediction problem, this paper proposed item-based collaborative filtering applying dimension reduction in a recommendation system [10]. Book recommendation system based on combine features of content-based filtering, collaborative filtering and association rule mining. paper presents book recommendation system based on combined features of content filtering, collaborative filtering and association rule mining.

[11] The literature survey on Item-Based Collaborative Filtering Recommendation Algorithms shows that Recommender systems are being stressed by the huge volume of user data in existing corporate databases. In this paper they presented and experimentally evaluated a new algorithm for CF-based recommender systems. Their results show that item-based techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations. [12] The literature survey on K-means clustering based solution of sparsity problem in rating-based movie recommendation system proposes to solve the sparsity problem in movie recommendation system that has the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. [13] A new collaborative filtering algorithm using K-means clustering and neighbors' voting, this paper considers the users are m (m is the number of users) points in n dimensional space (n is the number of items) and represents an approach based on user clustering to produce a recommendation for active user by a new method. It uses k-means clustering algorithm to categorize users based on their interests.

[14] This research work on Mining top k-sequential rules, proposes Top Seq. Rules, an efficient algorithm for mining the top-k sequential rules from sequence databases, where k is the number of sequential rules to be found and is set by the user [15]. Transitive node similarity for link prediction in social networks with positive and negative links, defines a basic node similarity measure that captures effectively local graph features and also exploits global graph features introducing transitive node similarity [16]. KA survey of book recommender systems categorizes the systems into six classes, and highlighted the main trends, issues, evaluation approaches and datasets [17]. FUCL mining technique for book recommender system in library service presents a book recommendation system for university libraries to support user interests which are related in the same topic and faculty. The main motive of this research is to develop the technique which recommends the most suitable books to users according to the faculty of the user profile with book category, and book loan.

[18] Simulation System of Book Goods Recommender Based on K-means Clustering Algorithm focuses on a new method is given based on user clustering and association rule mining. First, k-means algorithm is used to finish customer segmentation according to customer's bought records, so that birds of a feather flock together. Then, each cluster's personal rules are built by applying association rule mining on the cluster's purchase database. Finally, recommendations are made by applying cluster's personal rules on customer's bought records [19]. Book recommender system using fuzzy linguistic quantifier and opinion mining presents a feature-based opinion extraction and analysis from customers' online reviews for books using Ordered Weighted Aggregation [20] Hierarchical clustering for collaborative filtering recommender systems. In International Conference on Applied Human Factors and Ergonomics proposes a Recommender System based in Agglomerative Hierarchical Clustering for Collaborative Filtering.

3. Proposed System

The objective is to recommend books based on the reading history of user and to improve the precision of book recommendation to the user by improving the computational methods in collaborative filtering algorithm. Recommendation systems based on collaborative filtering usually results in accurate prediction when sufficient data or information is provided. User based Collaborative Filtering is successful to predict the customer behaviour and activities which may involve user interests. By improving the threshold values in collaborative filtering will filter out a greater number of users as well as books which are relevant and hence the recommendations can be given more accurately as a greater number of relevant data were provided. The four main modules to be executed for building our recommendation system are Data Pre-processing, Implementation of Collaborative Filtering Algorithm using Tensor Flow, Training with the dataset and Testing.

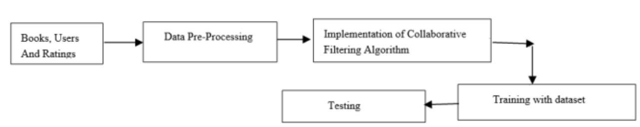

Fig. 1. Working flow of the recommender system modules

4. Data Set

The "Book Crossing Dataset" is the dataset used. It was compiled by Cai-Nicolas Ziegler in 2004 and consists of three tables for users, books and ratings. Explicit ratings are given on a scale of 1 to 10 (with higher values denoting greater appreciation), while implicit ratings are given a value of 0. This dataset Contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit /implicit) about 271,379 books.

Link to dataset files:

http://www2.informatik.uni-freiburg.de/~cziegler/BX/

Data Pre-processing: Data pre-processing enhances the quality of data to proceed the extraction of required insights from the data. This technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training model is the first most important step in our Book Recommender System. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. The data pre-processing steps does the following steps: (a) Merge user, rating and book data. (b) Remove unused columns. (c) Filtering books that have had at least 25 ratings. (d) Filtering users that have given at least 20 ratings.

A. *Implementation of Collaborative Filtering Model*

Collaborative filtering approaches build a model from user's past behaviour (i.e., items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that user may have an interest in. Collaborative filtering (CF) approaches overcome some of the limitations of content-based ones. Items for which the content is not available or difficult to obtain can still be recommended to users through the feedback of other users. CF ones can also recommend items with very different content, as long as other users have already shown interest for these different items.

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. Many websites use collaborative filtering for building their recommendation system. In user-based filtering, the users are selected for their similarity to the active user. This similarity is determined by matching users who have posted similar ratings. Based on the previous similarity, it is presumed that future likes and dislikes will also be similar. Once the pre-processed data is loaded into the project, the collaborative filtering model of book recommender system is built.

Typically, the workflow of a collaborative filtering algorithm is as follows:

1. A user expresses his or her preferences by rating items (e.g., books, movies) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item).
4. We look for users who share the same rating patterns with the active user (the user whom the prediction is for) and use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.
5. Build an item-item matrix determining relationships between pairs of items and infer the tastes of the current user by examining the matrix and matching that user's data.

Our technique is based on the following observations that, users who interact with items in a similar manner (for example, buying the same products or viewing the same articles) share one or more hidden preferences and users with shared preferences are likely to respond in the same way to the same items. Using Tensor Flow which is an open-source package that makes building, evaluating, and serving sophisticated recommender models easier as well as efficient in training recommendation models that jointly optimize multiple recommendation objectives.

The process in the Tensor Flow for building Collaborative filtering Book Recommender model is described as follow:

Normalizing the rating feature: Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between0 and 1. The formula is as follows: (value – min / max-min).

Building user, book matrix: We have already loaded training set through pandas with three columns: user, book and rating. Rows in the matrix correspond to users and columns to books, therefore entries correspond to ratings given by users to books.

Defining Network parameters: We will set up some network parameters, such as the dimension of each hidden layer and initialize the Tensor Flow placeholder. Weights and biases are to be randomly initialized.

Defining the model: We will define the model and the predictions by building the encoder and decoder functions. Auto encoders are unsupervised learning neural networks, they try to reconstruct input data at the output, this means that they learn a compressed representation of the input, and they use that to reconstruct the output. Defining loss function and optimizer, minimize the squared error, and defining the evaluation metrics: Once the structure of a neural network has been defined, we need a loss function. A loss function quantifies how much worse is our estimate on the current example, using the current parameters for the model. At the end of this module, the model based on collaborative filtering is defined using Tensor Flow and will be ready for training.

B. *Training with the dataset*

The Recommender model is to be trained with the existing filtered data of books and users. We proceed to split training data into batches, and we feed the network with them. And, then we can train our model with vectors of user ratings, each vector represents a user and each column a book, and entries are ratings that the user gave to book. For training the dataset, we prefer to use the mini batch gradient descent algorithm.

C. *Mini Batch Gradient Descent Algorithm*

Mini batch algorithm is the most favourable and widely used algorithm that makes precise and faster results using a batch of ‘m’ training examples. In mini batch algorithm rather than using the complete data set, in every iteration we use a set of ‘m’ training examples called batch to compute the gradient of the cost function. Mini-batch gradient descent is a variation of the gradient descent algorithm that splits the training dataset into small batches that are used to calculate model error and update model coefficients. It seeks to find a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent algorithms.

Batch gradient descent is the deterministic variant where we update the parameters with respect to the loss calculated on all training examples. The Mini batch gradient descent algorithm is as follows:

Initialize w: =0m−1, b:=0w:=0m−1,b:=0

for epoch e∈[1,..E]e∈[1,...,E]:

for every ⟨x[i],y[i]⟩∈D⟨x[i],y[i]⟩∈D:

Compute prediction y^[i]:=h(x[i])y^[i]:=h(x[i])

Compute loss
L:=1n∑ni=1L(y^[i],y[i])L:=1n∑i=1nL(y^[i],y[i])

Compute gradients
Δw:=−∇L[i]w,Δb:=−∂L∂bΔw:=−∇L[i]w,Δb:=−∂L∂b

Update parameters w:=w+Δw,b:=+Δb

For our Book Recommender system, we can now initiate the training our model with the Mini Batch Gradient Descent algorithm implemented using Tensor Flow. Firstly, the training data is split into batches, and we feed the network with them. We train our model with vectors of user ratings, each vector represents a user and each column a book, and entries are ratings that the user gave to books. The training model for 100 epochs with a batch size b of 35 would be consuming enough memories, so that the complete training set will feed our neural network 100 times, every time using 35 users. Finally, we must make sure to remove user’s ratings in the training set. This is to ensure that our system must not recommend books to a user in which has been already rated.

D. *Testing*

The process of this testing is to ensure that the individual components that comprise our recommender algorithms have been thoroughly tested in order to avoid bugs caused by tricky data structure errors. Finally, we will see how well our model performs by picking a User at random and seeing what books we can suggest to them. Testing with the Reader with User ID: 55927and we fetched the top 10 recommendations for the user. Based on the sorted normalized prediction scores, the above book recommendations are being given to the User with ID: 55927.

```
In [36]: top_ten_ranked.loc[top_ten_ranked['User-ID'] == 55927]
Out[36]:
```

	User-ID	Book-Title	Book-Rating
3632060	55927	The Secret Life of Bees	0.046972
3628837	55927	Harry Potter and the Chamber of Secrets (Book 2)	0.046207
3628841	55927	Harry Potter and the Prisoner of Azkaban (Book 3)	0.044000
3629406	55927	Life of Pi	0.042050
3627702	55927	Bridget Jones's Diary	0.042030
3628843	55927	Harry Potter and the Sorcerer's Stone (Harry P...	0.039456
3631984	55927	The Red Tent (Bestselling Backlist)	0.038370
3628839	55927	Harry Potter and the Goblet of Fire (Book 4)	0.037684
3628840	55927	Harry Potter and the Order of the Phoenix (Boo...	0.035098
3632690	55927	Where the Heart Is (Oprah's Book Club (Paperba...	0.034645

Fig. 2. Testing with user ID:55927

5. Results and Discussions

Our Book Crossing dataset comprises three CSV files of data for Users, Books and Ratings respectively. The Books data set provides book details. It includes 271,360 records and 8 fields like the ISBN, book title, book author, publisher details. The User data set provides the user demographic information. It includes 278,858 records and 3 fields: user id, location and age. The Ratings data set provides a list of ratings that users have given to books. It includes 1,149,780 records and 3 fields: User ID, ISBN, and Book Rating.

To build an effective Collaborative Filtering model, the filtering of data by means of users who have given similar ratings to the same books, thus creating a link between users, to whom books that were reviewed in a positive way will be suggested. It is indeed to build an efficient CF model with the usage of threshold values to filter the books that have had at least a particular threshold value rating. We have observed the functioning of the book recommender system with ten different sets of values for two parameters i.e., books that had been given at least ‘n’ ratings and secondly, the users who have rated at least ‘m’ books. The combinations of these (n, m) threshold values that were taken for the study are: (0,0), (10,5), (25,20), (20,60), (30,30), (50,40), (80,75), (100,100), (250,300) and (1000,500). Out of executing the recommender system with all the mentioned threshold values, it was observed that three distinct cases were prominent in determining the best threshold value for the recommender system.

Case 1: Threshold (50,40) - Filtering books that have had at least 50 ratings and users who have given at least 40 ratings. We finally got a result of filtered data of 2444 unique books out of total 271,360 data and the User data being filtered to 1260 out of a total of 278,858 unique users. The drawback observed in this case is that the number of users and books are way too refined and thus the model is built with sparse (or) with a selective number of users and books. In addition to that, there arises a conflicted situation in which recommendations are provided to only those users who have rated more books and the books which are only highly rated.

Case 2: Threshold (10, 5) - Filtering books that have had at least 10 ratings and users who have given at least 5 ratings. The filtered data of 17446 unique books out of total 271,360 data and the User data being filtered to 14322 out of a total of 278,858 unique users was fetched as a result. The drawback that we observed on execution is that, the number of users and books are way too big in data and thus the model is built with many association links between multiple users and books. Moreover, the Memory Error was observed due to the increased count of user data affecting the building of an efficient Collaborative Filtering Model.

Table 1
Evaluation metrics for threshold (50,40)

S. No.	User ID	Precision	Recall	F-Score
1	638	0.6	1.0	0.75
2	507	NA	NA	NA
3	487	NA	NA	NA
4	254	0.8	1.0	0.88
5	243	0.9	1.0	0.9470
6	8936	0.6	1.0	0.750
7	278582	NA	NA	NA
8	55927	0.6	1.0	0.750
9	2313	NA	NA	NA
10	189835	NA	NA	NA

Table 2
Evaluation metrics for threshold (10,5)

S. No.	User ID	Precision	Recall	F-Score
1	638	0.5	1.0	0.66
2	507	0.7	1.0	0.8
3	487	0.8	1.0	0.9
4	254	0.8	1.0	0.9
5	243	0.7	1.0	0.75
6	8936	0.6	1.0	0.7
7	278582	0.7	1.0	0.799
8	55927	0.6	1.0	0.75
9	2313	0.6	1.0	0.750
10	189835	0.5	1.0	0.6

Case 3: Threshold (25,20) - Filtering books that have had at least 25 ratings and filtering users who have given at least 20 ratings. Our final dataset contains 3,192 users for 5,850 books. And each user has given at least 20 ratings and each book has received at least 25 ratings. The execution time was better when compared to Case 2 and the Memory Error did not arise for this set of threshold values. The recommendations of books to the Users and the evaluation metrics were good for this particular threshold value among all the other sets of thresholds.

Table 3
Evaluation metrics for threshold (25,20)

S. No.	User ID	Precision	Recall	F-Score
1	638	0.6	1.0	0.75
2	507	0.7	1.0	0.8235
3	487	0.9	1.0	0.947
4	254	0.8	1.0	0.88
5	243	0.9	1.0	0.9470
6	8936	0.6	1.0	0.750
7	278582	0.8	1.0	0.88
8	55927	0.6	1.0	0.750
9	2313	0.7	1.0	0.8
10	189835	0.6	1.0	0.7

Thus, the top 10 books for the user are recommended based on the rating history of the user. With the execution and observations made on various set of values, it is determined that the results for the threshold of books that have had at least 25 ratings and users that have given at least 20 ratings, resulted in the best recommendations. This recommender system makes a considerable impact on providing better accuracy of prediction and much faster execution time in comparison with the traditional UBCF methods.

To determine accuracy of the working of our Book Recommender system, we are calculating the evaluation metrics as follows: (a) Precision, (b) Recall, (c) F-score (F-Measure). Precision is a metric that quantifies the number of correct positive predictions made, therefore, calculates the accuracy. It is calculated as the ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted. Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. The F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

The book reading history and the recommended books are compared and analysed for each user. For instance, it was observed that the User ID: 487 had given higher ratings for the books that came under the category of Ballad, Contemporary Romance, Fiction, Biography and Mystery and the most of the books that was fetched as the Top 10 book recommendations for this user were found to be in the same category. Similarly, the evaluation metrics are calculated for the users based on their reading, rating history with the recommended outputs for thresholds (25,20), (50,40) and (10,5) are tabulated.

This recommender system makes a considerable impact on providing better accuracy of prediction and much faster execution time in comparison with the traditional methods. It results in enhancing the quality of the recommendation system using collaborative filtering models. In addition to that, we have observed that improving the working of Collaborative Filtering model by means of building effective associations between users, books and ratings is highly important. With the execution and observations made on various sets of values, it is determined that the outputs for the threshold of books that have had at least 25 ratings and users that have given at least 20 ratings, resulted in the best recommendations.

A. *Cosine Similarity Calculation*

Cosine Similarity calculation uses the book rating information to process the comparability between Users and Books, which is utilized foe making efficient recommendations. For example, in User based models, the estimation of recommendation to User 'u' provides for books 'i' is determined as a conglomeration of some comparable users' evaluating of the books.

$$r_{u,i} = aggr_{uI \in U} \, r_{uI,i}$$

Where U denotes the set of top ‘N’ users that are almost like to user ‘u’ who rated item ‘i’.

The cosine-based approach which is the cosine-similarity between two users x and y is calculated as:

$$\boldsymbol{simil(x, y)} = \boldsymbol{\cos}(\vec{x} \cdot \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| * \|\vec{y}\|} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

Table 4
Results

ISBN	Book Title	Book Author	Year of Publication	Publisher	Cosine Similarity
0385470819	A Time to Kill	John Grisham	1993	Dell	0.778015
0452282152	1st to Die: A Novel	James Patterson	2002	Warner Vision	0.947023
0446610038	Girl with a Pearl Earring	Tracy Chevalier	2001	Plume Books	0.884574
0804106304	The Joy Luck Club	Amy Tan	1994	Prentice Hall (K-12)	0.892124

Where Ix, y is the set of items rated by both user x and user y.

An advantage of cosine similarity is its low-complexity, especially for sparse vectors. Only the non-zero dimensions need to be considered. The book ratings of the book "The Rainmaker" is taken as Vector 1 and the Vector 2 is taken the ratings of the books "A Time to Kill", "1st to Die: A Novel", "Girl with a Pearl Earring" and "The Joy Luck Club".

We have calculated the cosine similarity between these books, referenced from our base paper, compared the evaluation metrics tabulated the results as follows. It is observed that the Cosine Similarity factor was comparably high and efficient for our Book Recommender System.

6. Conclusion

This paper presented an overview on book recommender system using improved collaborative filtering.

References

[1] Sivabalaselvamani, D., Vidhyasree, M. S., Pavithra, M. P., Soundarya, M. G., & Preethika, M. M. Books and Movies Recommendation and Rating Prediction Based On Collaborative Filtering Networks.
[2] Tsuji, K., Takizawa, N., Sato, S., Ikeuchi, U., Ikeuchi, A., Yoshikane, F., &Itsumura, H. (2014). Book recommendation based on library loan records and bibliographic information. Procedia-social and behavioral sciences, 147, 478-486.
[3] Cui, L., Wang, C., & Yang, B. (2014). Personalized book recommendation based on ontology and collaborative filtering algorithm. The Open Cybernetics & Systemics Journal, 8(1).
[4] Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE International Conference on Data Mining, pp. 263-272.
[5] Ali, Z., Khusro, S., & Ullah, I. (2016, May). A hybrid book recommender system based on table of contents (toc) and association rule mining. In Proceedings of the 10th International Conference on Informatics and Systems, pp. 68-74.
[6] Khan, B. M., Mansha, A., Khan, F. H., & Bashir, S. (2017, December). Collaborative filtering based online recommendation systems: A survey. In 2017 International Conference on Information and Communication Technologies (ICICT), pp. 125-130.
[7] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. JSW, 5(7), 745-752.
[8] Phorasim, Phongsavanh & Yu, Lasheng. (2017). Movies recommendation system using collaborative filtering and k-means. International Journal of Advanced Computer Research. 7. 52-59.
[9] Lee. Y. (2015) "Recommendation system using collaborative filtering". Master's Projects.
[10] Tewari, A. S., Kumar, A., & Barman, A. G. (2014, February). Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining, 2014
[11] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, Proceedings of the 10th, 2001 Computing Conference (IACC), pp. 500503.
[12] Shrivastava, R., & Singh, H. (2017). K-means clustering based solution of sparsity problem in rating based movie recommendation system. International Journal of Engineering and Management Research (IJEMR), 7(2), 309-314.
[13] Dakhel, G. M., & Mahdavi, M. (2011, December). A new collaborative filtering algorithm using K-means clustering and neighbors' voting. In 2011 11th International Conference on Hybrid Intelligent Systems (HIS), (pp. 179-184).
[14] Fournier-Viger, P., & Tseng, V. S. (2011, December). Mining top-k sequential rules. In International Conference on Advanced Data Mining and Applications, (pp. 180-194). Springer, Berlin, Heidelberg.
[15] Symeonidis, P., Tiakas, E., & Manolopoulos, Y. (2010, September). Transitive node similarity for link prediction in social networks with positive and negative links. In Proceedings of the fourth ACM conference on Recommender systems, pp. 183-190.
[16] Alharthi, H., Inkpen, D., & Szpakowicz, S. (2018). A survey of book recommender systems categorizes the systems into six classes, and highlighted the main trends, issues, evaluation approaches and datasets. Journal of Intelligent Information Systems, 51(1), 139-160.
[17] Jomsri, P. (2018). FUCL mining technique for book recommender system in library service Procedia Manufacturing, 22, 550-557.
[18] Rong, L. I, Simulation System of Book Goods Recommender Based on K-means Clustering Algorithm [J]. Computer Simulation, vol. 6, 2010.
[19] Sohail, S. S., Siddiqui, J., & Ali, R. (2016, September). Book recommender system using fuzzy linguistic quantifier and opinion mining. In The International Symposium on Intelligent Systems Technologies and Applications, pp. 573-583, Springer, Cham.
[20] Chalco, C. I., Chasi, R. B., & Ortiz, R. H. (2018, July). Hierarchical clustering for collaborative filtering recommender systems. In International Conference on Applied Human Factors and Ergonomics, pp. 346-356, Springer, Cham.