Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Algorithm for movie recommendation system using collaborative filtering

Nisha Bhalse [a,*], Ramesh Thakur [b]

[a] IET, DAVV, Indore, MP, India
[b] IIPS, DAVV, Indore, MP, India

ARTICLE INFO

Article history:
Received 26 December 2020
Accepted 10 January 2021
Available online xxxx

Keywords:
Recommendation system
Collaborative filtering
SVD
Cosine similarity
Top n recommendation list

ABSTRACT

Recommender systems are information filtering system that predicts the rating for users and items, basically from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. Most traditional recommender systems lack accuracy in the case where data used in the recommendation process is sparse. This study addresses the sparsity problem and aims to eliminate it through a singular value decomposition collaborative filtering approach applied to a web-based movie recommendation system. This paper proposed a movie recommendation system whose primary objective is to suggest a recommender list through singular value decomposition collaborative filtering and cosine similarity. We enhance the model with a factorization form, which greatly reduces the model's number of parameters with a controlled complexity. This paper proposed a movie recommendation system whose primary objective is to suggest a recommender list through singular value decomposition collaborative filtering and cosine similarity. The present work improves these approaches by taking the movies' content information into account during the item similarity calculations. The proposed approach recommends the top n recommendation list of movies to users on user's interest preferences that were not already rated. Graphically shows the percentage of already viewed movies by user and movies recommended to User.

Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

1. Introduction

The recommendation systems have become an important research area; the first paper on the collaborative filtering approach appeared during the mid-1990 s [1,2]. Both industry and academia have worked together to develop new approaches and contribute to the recommender systems improved quality during the last decade.

A recommendation system is one type of information filtering system, which filters items by User's interests. In recent decades, the recommendation system became an inalienable part of e-commerce and social websites due to the problem overload. In an era of information overloading, recommendation systems have developed for discovering the interesting item according to the User's preference or choice [3,4]. It was applied in many areas, such as online learning, e-commerce, etc. Examples of such applications are YouTube, Amazon.com, MovieLens, Netflix, Facebook, etc.

Fig. 1 shown the recommendation system (R.S.) is classified into three categories content-based (C.B.) approach, collaborative filtering (C.F.) approach, and hybrid approach. The content-based approach is made on the User's profile preferences and descriptions of items [5]. The User has recommended the items according to user ratings, including the number of times they clicked on different items or maybe even liked them. Collaborative filtering algorithm is collected and analyzes enormous information on all user's behavior and preferences and then predicts the item; users will like a similar item to other similar users [6]. The hybrid recommendation is a combination of these two approaches [7].

The C.F. based algorithm is categorized in the memory-based C.F. and the model-based C.F. [8]. The memory-based using chronological data to find a similar item. Memory-based CF recommendation is an actual execution in the e-commerce industry due to its reasonable and ease of implementation. However, this method is suffering from data sparsity, which occurs mainly as users rate only a few products from a large number of available products. Furthermore, when it increases the users and items,

* Corresponding author.

https://doi.org/10.1016/j.matpr.2021.01.235
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

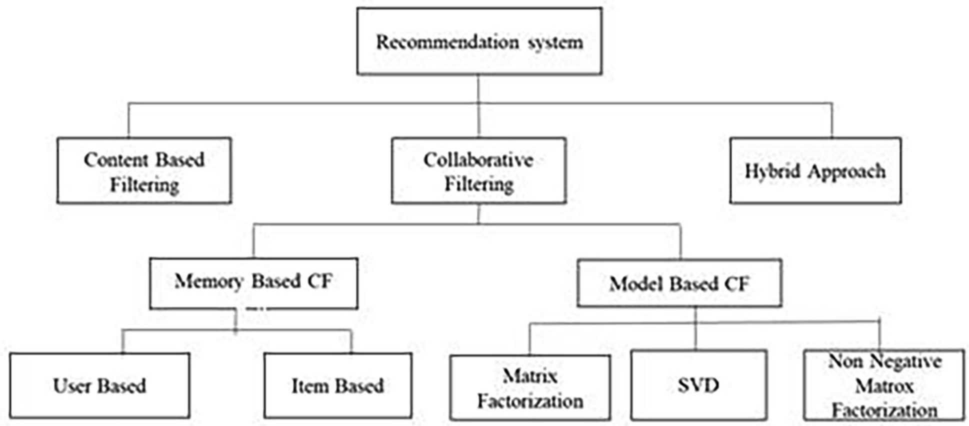

Fig. 1. Classification of recommendation system [5].

the computational complexity also increases, leading to low scalability becoming difficult to make recommendations [9].

The model-based C.F. recommendation has been introduced to overcome the challenges of memory-based C.F. These techniques discover the rating pattern from historical data and give highly accurate and effective recommendations from some sample data. Clustering model [10] and matrix factorization [11] are familiar techniques used in model-based approaches.SVD, SVD++, and ALS are popular M.F. techniques that have gained importance in the Netflix price challenge's recommender system.

Collaborative filtering has effectively used techniques in recommender systems, and the content of recommended items are independent. Recommender systems take input from different sources to make recommendations. The most common way of them is to collect inputs through the user's feedback. Shopping and movie websites are collect ratings provided by users, and it is called explicit feedback—another way to build recommendations through users' implicit feedback [11]. The central issue of collaborative filtering is how to handle massive data to make precise recommendations, and another challenge is data sparsity; the user ratings show the interest are very sparse as compared to users and item are present in the recommender system, and one more problem is a cold start problem, the items which have no rating is available then how to recommend to users.

This paper proposed an algorithm for a recommendation system using collaborative filtering that predicts the top *n* recommendation list of movies to the active User by using singular value decomposition and cosine similarity for computing similarity between users, utilizing publicly available MovieLens datasets. The main contribution of this paper is as follows:

a) It interprets the user-item rating matrix to overcome sparsity and cold start, and scalability issues.
b) It converts the sparse matrix of user-item into the dense matrix of user item.
c) To make recommendations relevant, using the cosine similarity between the users.
d) Finally, based on the above analysis, the paper presents a movie recommendation system that predicts the active User's top n relevant product list.

The Ringo video recommendation system is a web application that generates videos, movies, and music recommendations for users[12]. Tapestry is the first experimental mail in which to implement collaborative filtering techniques for the recommendation system. In the recommendation system in which user preferences are extracted from the ratings, these are explicitly or implicitly given by the users. The recommendations' primary aim is available for the user to spend some more time watching the video according to user choice. A personalized recommendation system in the recommendation is generated; for example, the YouTube recommendation system combines the User's behavior with website and association rules of video [13]. Any video that the User has explicitly liked added to the playlist or given a rating that acts as a seed of recommendation factor.

The proposed Multilinear Interactive Matrix Factorization Algorithm (MLIMF) is a user interaction model and associates each event to its final decisions. This model worked on pairwise interaction and some empirical factors [14]. It computed the similarity between two n-dimensional vectors based on their angle. The cosine-based computation is used for text mining and information and compared two text documents. The User u and v between computed the similarity define as follows [16]:

$$Sim(u,v) = \frac{u,v}{|u|,|v|} = \frac{\sum_i r_{u,i} \cdot r_{v,i}}{\sum_i r_{u,i}^2 * \sum_i r_{v,i}^2}$$

The similarity scores are calculated using the cosine similarity method, which expressed similarities between users and items. These scores are the basis for generating recommendations based on users or items.

The remainder of the paper structure is as follows. The proposed recommendation system algorithm is described in section II. The experimental setup and results are described in Section III. Section IV, in which the conclusion of experimental work and future work is described. Proposed Methodology of Recommendation System

2. Proposed methodology of recommendation system

We proposed a collaborative filtering algorithm for the recommendation system. The algorithm worked on singular value decomposition and cosine similarity algorithm. It predicts the top n products list recommended to the active User. The proposed methodology initially constructs a user-item utility matrix: We collect the rating information about the User after browsing and purchasing behavior, then clean and convert the data into input data called the user-item utility matrix. The sparse matrix is firstly normalized, and then it becomes dense. This dense matrix applies singular value decomposition, which reduced the matrix according to the latent factors. The similarity between users computes through a cosine similarity algorithm. Though cosine similarity

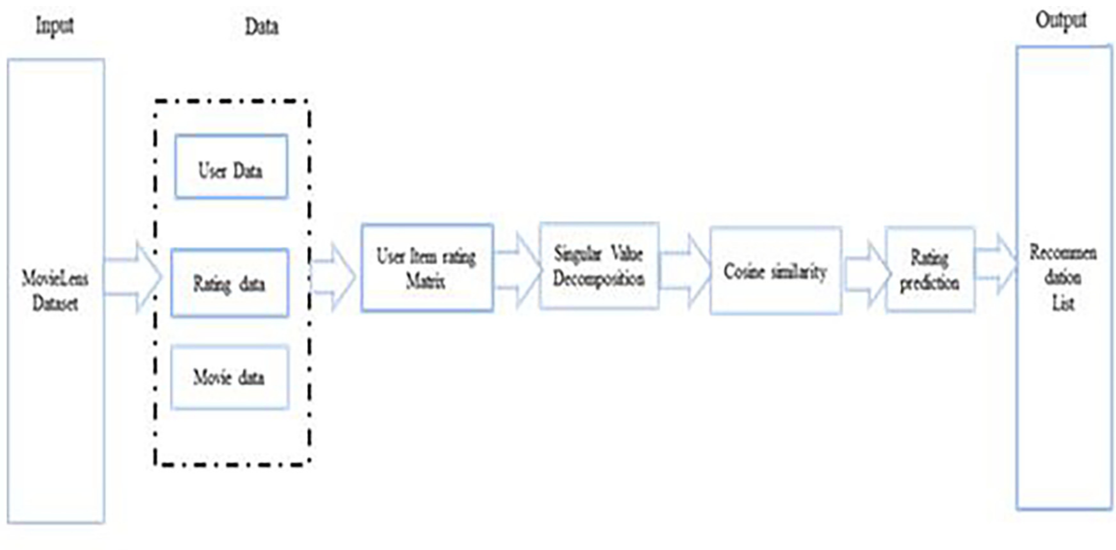

Fig. 2. Structure of proposed movie recommendation system.

Table 1
Used symbols.

Symbol	Definition
U	Users set
I	Items set
$_R$mxn	User item matrix
u	User
ua	Active User
i	Item
uij	User i rating on item j
N	Number of items
K	The dimension of latent factors

active User searching neighbors on the preferences of the rating information of same rated items by users and then recommended the top *n* product list those are not rated by active User. Fig. 2 showed the structure of our proposed recommendation system.

Secondly, similarity calculation: Similarity calculation between users is the common technique of C.F. recommendation technique. The proposed method in which the co-measure algorithm is used for measurement and then sort the User's list toward the active User.

Thirdly, neighborhood election: Based on a similarity ranking among users, k-nearest neighbors are used for implementing the predicted set, or similarity thresholds are set, and users exceeding the threshold for selection of active User's neighbors.

Finally, prediction of ratings and movie recommendation: Once the active User has obtained the nearest neighbor set, the weight of similarity score are obtained for unrated movies and generate a top n movie to the active User.

2.1. Proposed algorithm and basic notation

We introduce the preliminaries of model-based singular value decomposition of the collaborative filtering approach. Table 1 shows the symbol and their representation used in this paper.

The proposed algorithm for collaborative filtering recommendation system. The algorithm shows in the following steps:

Algorithm:. *Input: Active user* u_a
R: The user-item utility rating matrix
Output: N items to be recommended.

Step 1: Compute the mean rating of matrix R.

Step 2: Compute the user vector and item vector's inner product by equation (1) R = U $\sum$ V^T.

Step 3: Use equation (2) for calculating the User's similarity and generated the matrix of similarity indicated by

$$sim(u,v) = \frac{u,v}{|u|,|v|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sum_i r_{u,i}^2 * \sum_i r_{v,i}^2}$$

Step 4: Sort the similarities score of matrixes in descendent order.

Step 5: Obtain the neighbor set for active Users based on user similarity score.

Step 6: For predicting rating using equation $(u,v) = \bar{R}_u + \frac{\sum v \in K\, sim(u,v)(r_{v_i} - r_v)}{\sum v \in K|sim(u,v)|}$

Step 7: Generate the recommendation List.

2.2. Singular value decomposition (SVD)

SVD is one technique of matrix factorization used in model-based collaborative filtering recommendation system. The SVD reduced the feature of dataset and space dimension from N dimension to K dimension (where K < N) [15].

SVD is an algorithm that breaks a matrix A into an approximation of the original matrix A. Mathematically; it decomposes A into a diagonal matrix and two unitary matrices:

$$R = U \sum V^T \tag{1}$$

Where R is an m $\times$ n rating matrix, U is an m $\times$ r orthogonal left singular matrix that represents user "features" matrix, Σ is an r $\times$ r is the singular values of a diagonal matrix, and V$^{.T.}$ is an r $\times$ n diagonal right singular matrix, which indicates the similarity between items and latent factors.U represents how many users "like" each feature and V$^{.T.}$ represents how relevant each feature is to each movie. The SVD decreased dimension of rating matrix *A* by extracting its latent factors and map each User to each item is called *r*-dimensional latent space. Fig. 3 shown the mapping relationship between user and item. [15].

SVD is used in model-based collaborative filtering recommendation systems; it involves user set, item set, and user preferences on items, which are often represented by the [user, item, rating].

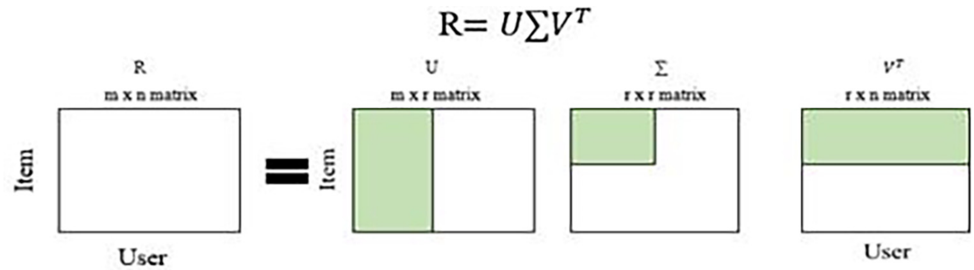

Fig. 3. Singular value decomposition matrix [15].

The rating matrix R, R ∈ R, containing m × n where m is number of Users and n is the number of items obtained, and each rating rij characterizes as a user I prefer on item j.

2.3. User similarity computation

When we build a movie recommendation system framework, it is most important to measure User's similarity. User's similarity is measured through the commonly used cosine similarity algorithm in C.F. recommendation algorithms [17]. If the User is not rated on an item, then the user rating is set to 0. Let *U= {u1, u2, u3, u4,………Um}* and *I= {i1, i2, i3, i4,…………In}* is user set and item set. The rating matrix is represented as $R = [r_{ui}]^{m*n}$. The r_{ui} expressed the rating on item i by User u. The cosine similarity equation is following [17]:

$$sim(u,v) = \frac{u,v}{|u|,|v|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sum_i r_{u,i}^2 * \sum_i r_{v,i}^2} \quad (2)$$

2.4. Neighborhood selection

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. We segregate the similar item and make a set, according to active user rating, use a k nearest neighbor algorithm, and obtain a prediction.

2.5. Rating prediction

The rating prediction computed on an unrated User item, we obtain the neighbors set of k items for active user u_a. In equation (2), we measure the similarity between users. Then, construct a set for the first k users near the active User according to user similarity values. The predicting value is determined by the following equation [18]:

$$P(u,i) = \bar{R}_u + \frac{\sum v \in K\, sim(u,v)(r_{v_i} - r_v)}{\sum v \in K|sim(u,v)|} \quad (3)$$

2.6. Top n movie recommendation list

For every User, predicted rating P (u, i) is calculated for each item i ε I not rated by the active User. After getting the neighbor set of active users, we use a similarity score as a weight to predict an unrated item from the active User. Recommendations are generated by computing the sum of predicted ratings on items similar weighted by the similarity score and normalized by using the sum of the similarity values and getting a Top n movie. These movie lists are recommended to the active User.

3. Experiment

3.1. MovieLens dataset

We use the dataset of MovieLens in the conduct of experimental work. MovieLens dataset is publicly available on the MovieLens group [19]. The dataset includes movie ratings and many more fields. Each record of data represents movie ratings by different users. The movies.csv, ratings.csv, and user.csv files are used in our Recommendation System. These datasets contain the 100,837 ratings applied over 9743 movies. User reviews are presented on the rating scale through 1 to 5 numbers. Number 1 shows least liked, and number 5 shows most liked. Table 2 shown the description of the movielens benchmark dataset.

3.2. Experimental platform

Our experiments were implemented in python 3.8 using Jupiter notebook. We run the Collaboratory experiments that allow everyone to write and executing Python code via browser, and it is particularly suitable for data analysis and machine learning. Colab has no set required for use, and it is hosted Jupyter notebook service, while it provided free access for computing resources, for example, GPUs, etc.

3.3. Building movie recommendation system

3.3.1. Pre-Processing

We are firstly transforming movie rating into a user-item matrix; it is called a rating utility matrix.

Each matrix cell is filled with a rating that is provided by the User. The sparse matrix in which many cells are empty for not rated by the User to that movie. Collaborative filtering algorithm worked on a dense matrix. So the sparse matrix is converted into a dense matrix by using the normalization. The empty cells represent new users, new movies, and not rated by anyone. The positive users towards the movie will always be rated the 4 or 5 rating and negative rating rates1or2. Therefore, we need to normalize the ratings for item and user bias. This can be done by taking the mean normalization. Fig. 4 shown the transform the dataset into user-item matrix.

3.3.2. Singular value decomposition

After the pre-processing of data, we need to start the process of model building. SVD is a widely used technique in collaborative filtering. The user ratings are human-generated features of the

Table 2
Description of the MovieLens benchmark dataset.

Name	User	Movies	Rating
ml-latest-small	610	9743	100,837

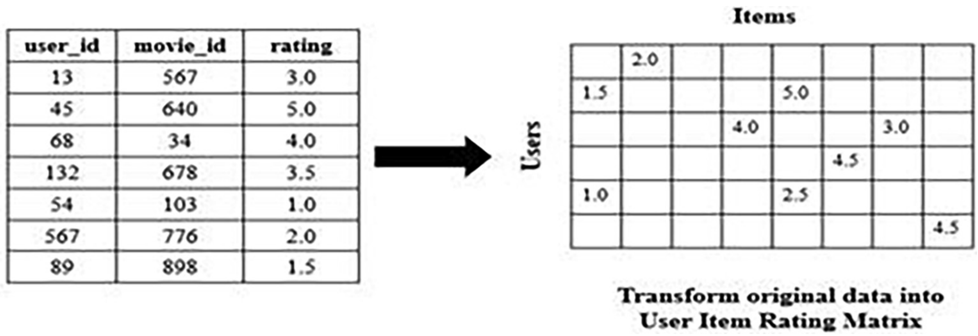

Fig. 4. Transform dataset into User -item rating matrix.

movie. However, certain features are not directly observable, but they are necessary for rating predictions. The set of hidden features are called Latent features. We don't know specifically what each latent feature represents. Still, one feature can be assumed to represent that a user likes a comedy film, and another latent feature may reflect that the User likes animation movies and so on.

3.3.3. Prediction & computation of neighborhood

When we have applied the SVD on a normalized matrix, then a reduced matrix is generated. We computed the similarity between other Users and Active Users by equation (2). Similarities between users follow the same pattern of rating. The users have a similarity score is equal to or greater than the threshold value they are neighbors of active User.

Once neighbourhood users are found, their ratings are aggregated to form a predicted score value of rating to active User. Through equation (3) to calculate the predicted score value on unrated items of active user u_a, we get the k set of neighbors of active user u_a. First, k users, those prediction scores near the active User, have constructed the neighbor set. And we compute the predicting value.

3.3.4. Top n recommendation generation

Experiments on the ML-latest-small dataset are implemented in this section. Our proposed algorithm predicts the top *n* most similar movies recommended for active Users, those movies have not been rated or seen by the active User. The selection of nearest neighbors is an important factor in the quality of the recommendations. In experiments, nearest neighbor k ranges from 10 to 30. Nowadays, the cosine similarity method is a widely used algorithm for calculating the similarity between users. Fig. 5 shows the Top 10most similar movies are recommended to the active User 305; results show the Movie Id, the movie's title, and movie genres.

Fig. 6 shows the graphical representation of movie genres and percentage of already viewed movies by the User and movies recommended to the User.

4. Conclusion and future work

The collaborative filtering algorithm is generally used for the recommendation system. With the use of technology on a large scale, so the volume of information has generated every day, it's getting difficult to find the appropriate information according to the User's choice. User's preferences on items are stored in the rating matrix; it is used to make the relationship between item and User to find the relevant item. Thus, the Collaborative filtering algorithm has faced a large dataset and sparseness in the rating matrix.

This paper proposed an algorithm for collaborative filtering recommendation system and applied it in the movie recommendation system. This personalized recommendation system uses the singular value decomposition algorithm and User-based co-coin similarity algorithm; these recommend the top *n* movies to the active User. The SVD can handle the massive dataset and sparseness of the rating matrix. A cosine similarity algorithm is used to measure the similarity between users. Those users rated the same movie they have similar choice or preferences and proposed an algorithm that recommended only not rated movies with a similar choice to active Users.

```
# Top 10 movies that User 305 hopefully will enjoy
predictions
```

movieId	title	genres
2005	Goonies, The (1985)	Action|Adventure|Children|Comedy|Fantasy
3967	Billy Elliot (2000)	Drama
104879	Prisoners (2013)	Drama|Mystery|Thriller
7458	Troy (2004)	Action|Adventure|Drama|War
2278	Ronin (1998)	Action|Crime|Thriller
59369	Taken (2008)	Action|Crime|Drama|Thriller
105504	Captain Phillips (2013)	Adventure|Drama|Thriller|IMAX
2746	Little Shop of Horrors (1986)	Comedy|Horror|Musical
3471	Close Encounters of the Third Kind (1977)	Adventure|Drama|Sci-Fi
40819	Walk the Line (2005)	Drama|Musical|Romance

Fig. 5. Recommended Top 10 movies for active user.

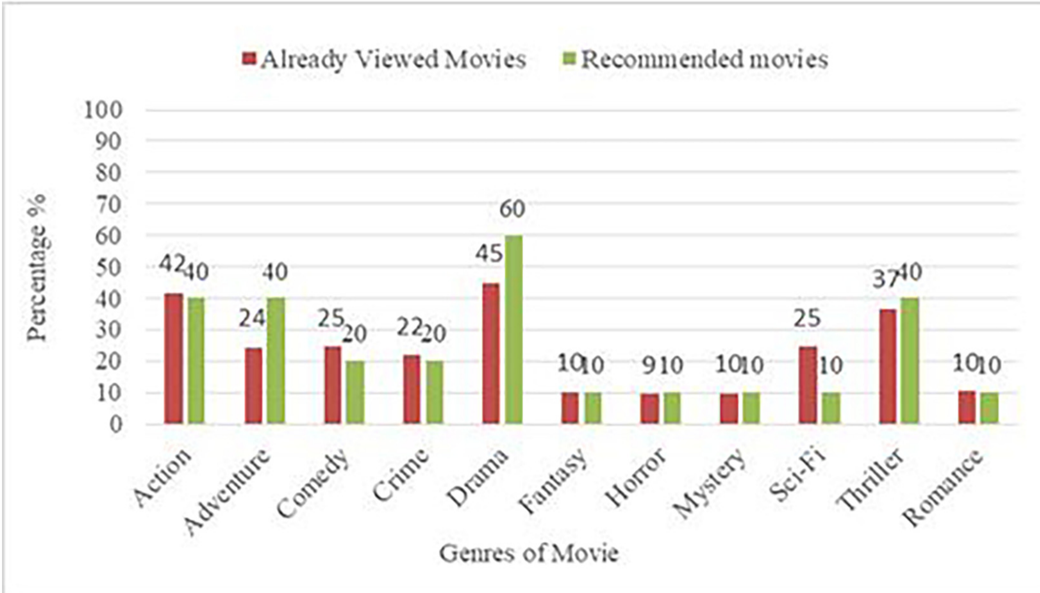

Fig. 6. Percentage of already viewed movies by User and movies recommended to User.

Our proposed algorithm is recommended to top n movies to the User's interest's active user base. This makes our approach usable in real-world scenarios. Future work can be extended by evaluating the proposed algorithm on evaluation parameters.

CRediT authorship contribution statement

N. Bhalse: Investigation, Methodology, Software, Supervision, Validation, Writing draft & Editing, Conceptualization, Formal analysis. **R. Thakur:** Project Administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. Hill, L. Stead, M. Rosenstein, G. Furnas(1995). Recommending and Evaluating Choices in a Virtual Community of Use. Conference on Human Factors in Computing Systems.
[2] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, Proc. Comput. Supported Cooperative Work Conf. (1994).
[3] G. Linden, B. Smith, J. York, Amazon.com recommendations: Item-to-item collaborative filtering, IEEE Internet Comput.. 7 (1) (2003) 76–80.
[4] Shuiguang Deng, Dongjing Wang, Ying Li, A recommendation system to facilitate business process modeling, IEEE Trans. Cabernets 47 (6) (2017) 1380–1394.
[5] L. Yao, Q.Z. Sheng, A.H.H. Ngu, J. Yu, A. Segev, Unified collaborative and content- based web service recommendation, IEEE Trans. Services Comput. 8 (3) (2015) 453–466.
[6] B. Guo, S. Xu, D. Liu, L. Niu, F. Tan, Y. Zhang. (2017).Collaborative filtering recommendation model with user similarity filling.IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC) pp. 1151-1154.
[7] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (1) (2013) 109–132.
[8] J.S. Breese, D. Heckerman, C. Kadie(2013).Empirical analysis of predictive algorithms for collaborative filtering.Vol. 7(7), pp. 43–52.
[9] J. Herlocker, J.A. Konstan, J. Riedl, An empirical analysis of design choices in neighborhoods-based collaborative filtering algorithms, Information Retrieval 5 (4) (2002) 287–310.
[10] J. Das, P. Mukherjee, S. Majumder, P. Gupta (2014).Clustering-based recommender system using principles of voting theory.International Conference on Contemporary Computing and Informatics (IC3I), Mysore, pp.230-235.
[11] R. Bell, Y. Koren, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[12] A.B. Barragáns-Martínez, E. Costa-Montenegro, J.C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, A. Peleteiro, A hybrid content based and item-based collaborative filtering approach to recommend T.V. programs enhanced with singular value decomposition, Inf. Sci. (Ny) 180 (22) (2010) 4290–4311.
[13] Luis G. Perez, Francisco Chiclana, Samad Ahmadi(2011).A Social network representation for Collaborative Filtering Recommender Systems.International Conference on IntelligentSystems Design and Applications (ISDA) pp 438-443.
[14] L. Yu, C. Liu, Z. Zhang, Multi-linear interactive matrix factorization, Knowl.-Based Syst. 85 (2015) 307–315.
[15] M.W. Berry, Large-scale sparse singular value computations, Int. J. Supercomput. Appl. 6 (1) (1992) 13–49.
[16] G. Adomavicius, A. Tuzhilin, Toward then extends generation of recommender system.A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[17] F.O. Isinkaye,Y.O. Folajimi, B.A. Ojokoh (2015). Recommendation systems: Principles, methods and evaluation. Egypt. Informat. J.
[18] Guangxia Xu, Zhijing Tang, Chuang Ma,Yanbing Liu,Mahmoud Daneshmand (2019).A Collaborative Filtering Recommendation Algorithm Based on User Confidence and Time Context.Hindawi Journal of Electrical and Computer Engineering. Article ID 7070487, 12 pages.
[19] MovieLens,http://www.grouplens.org/node/73.

Further Reading

[1] H. Nakayama, A. Hattori, Incremental learning and forgetting in RBF networks and SVMs with applications to financial problem.Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2003, pp. 1109–1115.