Electronic Letters on Computer Vision and Image Analysis 19(3):18-37, 2020

A Review of Movie Recommendation System: Limitations, Survey and Challenges

Mahesh Goyani [*] and Neha Chaurasiya[+]

[*] *Department of Computer Engineering, Government Engineering College, Modasa, Aravalli, India*
[+] *Department of Computer Engineering, Government Engineering College, Modasa, Aravalli, India*

Received 6th of May 2020; accepted 26th of august 2020

Abstract

Recommendation System is a major area which is very popular and useful for people to take proper automated decisions. It is a method that helps user to find out the information which is beneficial to him/her from variety of data available. When it comes to Movie Recommendation System, recommendation is done based on similarity between users (Collaborative Filtering) or by considering particular user's activity (Content Based Filtering) which he wants to engage with. To overcome the limitations of collaborative and content based filtering generally, combination of collaborative and content based filtering is used so that a better recommendation system can be developed. Also various similarity measures are used to find out similarity between users for recommendation. In this paper, we have surveyed state-of-the-art methods of Content Based Filtering, Collaborative Filtering, Hybrid Approach and Deep Learning Based Methods for movie recommendation. We have also reviewed different similarity measures. Various companies like facebook which recommends friends, LinkedIn which recommends job, Pandora recommends music, Netflix recommends movies, Amazon recommends products etc. use recommendation system to increase their profit and also benefit their customers. This paper mainly concentrates on the brief review of the different techniques and its methods for movie recommendation, so that research in recommendation system can be explored.

Key Words: Recommendation System, Hybrid Filtering, Matrix Factorization, SVD, Similarity Measures.

1 Introduction

Recommendation System is an information tool which helps users to find out the items which they want from the large no of items available [1], [2]. Main goal of recommendation system is to forecast the rating which a specific user gives to an item. It helps the user to find the best solution from the available list of items. Many companies use recommendation system so that they can serve their user and raise their profit like Netflix, YouTube, Amazon and others Still now it is a good topic of research because to find what the user wants from available resource is a big challenge, as our choice keeps on changing with time. Nowadays what we purchase online is recommendation. For example, if we want to buy books, listen music, watch movies etc there is one

Correspondence to: <mgoyani@gmail.com>

Recommended for acceptance by <Youssef EL HABOUZ>
https://doi.org/10.5565/rev/elcvia.1232
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

recommendation system that is working in background which suggest the user based on his previous actions [3]. Many platforms like Netflix which suggest movies, Amazon which suggest products, Spotify that suggest music, LinkedIn that is used for recommending jobs or any social networking sites which suggest users, all these work on recommendation system [4], [5]. By using these recommendation engine users can easily find out what he wants according to his/her choice. So to build an effective recommender system is also a challenge because user's preference keeps on changing with time.

1.1 Applications

Recommendation System is a vast area which is used everywhere in every field. People use recommendations as it saves time, so it plays a vital role in various areas. It is used in many real life applications like Entertainment, E-Commerce, Services, Social Media etc. [6]. In Entertainment area recommendation system is widely used in watching movies or listening music or any TV program. When we talk about E-Commerce field, Amazon is the world's largest shopping site. Some use it for purchasing books, for buying any household products or any products, some use it for clothing. So this way whole world is dependent on these E-commerce sites for one or the other work. Some other E-Commerce sites like Flip cart, E bay, Myntra, Shop clues etc. also provides recommendations. Some other applications of recommendation system are listed below:

- Movie Recommendation: Netflix uses algorithm for recommending movies according to their interest. Other such platforms that provide recommendations include hotstar, sonyLIV, voot, ALTBalaji etc [7].

- Music Recommendation: Pandora generates a radio station. It uses the properties of songs to recommend other songs. Other medium in this field that suggest music recommendation are Spotify, JioSavan, Gaana etc [8].

- News: Various applications that provide news recommendation can be Google News, Apple News(integrated into IOS and macOS), Flip board, Feedly, Tweet Deck, Pocket, Mix, Zig, News360. All these suggest news, articles, blog post, content from top publishers etc [9].

- Fashion: People can buy various clothing items of their choice. This section include various shopping sites like Myntra, Amazon, Club Factory, SHEIN, Lime Road, Flip cart and others [10].

- Travel service: Recommendation helps here to suggest various travelling sites to safeguard journey. This includes Road trippers which leads you plan any road trip with ease. Using Hooper, users can input their travel plans, and the app will tell them when is the best time to book their flight [11].

1.2 Challenges

There are various challenges faced by Recommendation System. These challenges are Cold Start problem, Data Sparsity, Scalability.

Cold Start Problem: It needs enough users in the system to find a match. For instance, if we want to find similar user or similar item, we match them with the set of available users or items. At initial stage for a new user, his profile is empty as he has not rated any item and the system do not know about his taste, so it becomes difficult for a system to provide him recommendation about any item. Same case can be with new item, as it is not rated by any user because it's new for the user. Both these problem can be resolved by implementing hybrid techniques [12].

Data Sparsity: The user or rating matrix is very sparse. It is very hard to find users that have rated the same items because most of the user does not rate the items. So it becomes hard to find set of users who rate the items. To give recommendation is really tough when there is less information about any user [13].

Scalability: Collaborative Filtering use massive amount of data to make reliable better which require more number of resources. As information grows exponentially processing becomes expensive and inaccurate result from this Big data challenge [14].

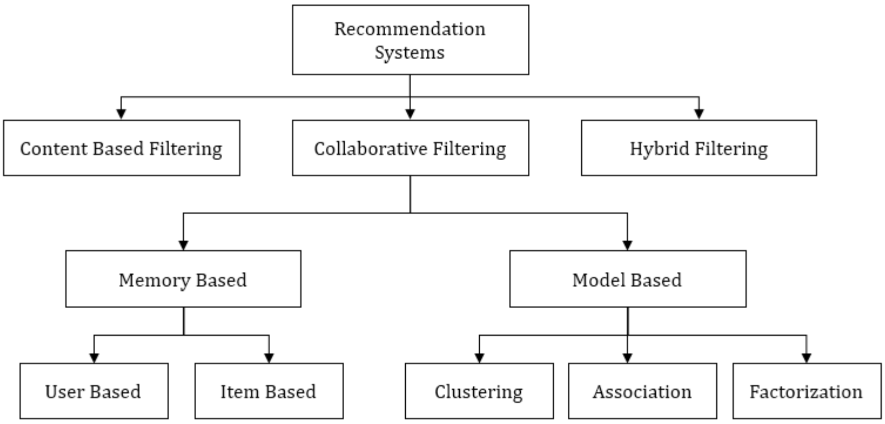

Figure 1: Taxonomy of Recommendation System

Rest sections of this paper are organized as follows: In section 1, types of filtering techniques and their subtypes are discussed with detail survey and comparison. Along with this, similarity measures and evaluation metrics are explored. Further, description about dataset is drafted in section 3. Lastly, the article is concluded after discussing the advantage, challenges and limitations of the work with enabling future scope in Section 4.

2 Literature Review

There are three techniques of recommendation system: Collaborative Filtering, Content-Based Filtering and Hybrid Filtering. In Content Based recommender system, user provides data either explicitly (rating) or implicitly (by clicking on a link). The system captures this data and generates user profile for every user. By making use of user profile, recommendation is generated. In content based filtering, recommendation is given by only watching single user's profile. System tries to recommend item similar to that item based on users past activity. Unlike content based, collaborative filtering finds those users whose likings are similar to a given user. It then recommends item or any product, by considering that the given user will also like the item which other users like because their taste are similar [15]. Both these technique have their own strength and weakness so to overcome this, hybrid technique came into picture, which is a combination of both these techniques. Hybrid filtering can be used in various types. We can use content based filtering first and then pass those results to collaborative recommender (and vice-versa) or by integrating both the filter into one model to generate the result. These kinds of modifications are also uses to cope up with cold start, data sparsity and scalability problem. Taxonomy of Recommender System is depicted in figure 1.

Various recommendation systems are surveyed in following section.

2.1 Content-Based Filtering

Content-Based Filtering are also known as cognitive filtering [16]. This filtering recommends item to the user based on his past experience. For example, if a user likes only action movies then the system predicts him only action movies similar to it which he has highly rated. The broader explanation could be suppose the user likes only politics related content so the system suggests the websites, blogs or the news similar to that content. Unlike collaborative filtering, content based filtering do not face new user problem. It does not have other user interaction in it. It only deals with particular user's interest. Content based filtering first checks the user preference and then suggest him with the movies or any other product to him. It only focus on single user's ideas, thoughts and give prediction based on his interest. So if we talk about movies, then the content based filtering technique checks the rating given by the user. The approach checks which movies are given

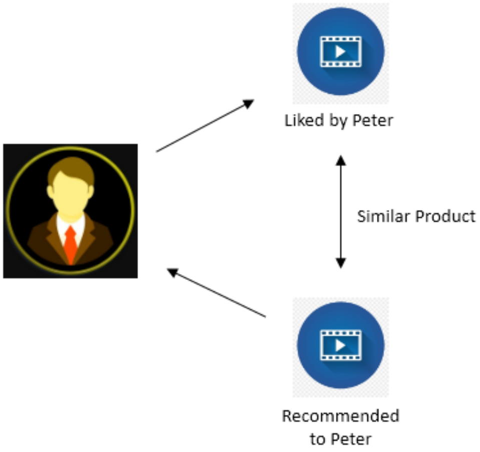

Figure 2: Content Based Filtering

high ratings by the user by checking the genre categories in the user profile. After analysing user profile, the technique recommends movies to user according to his taste. The figure 2 shows us how Content-Based Filtering works.

As shown in figure 2, content based filtering whole process is shown by giving an example of Geometric Shapes [17]. Here in the figure, first an Item Profile is developed based on the liking of the user. Here the user likes circle and triangle of blue colour. Now based on item profile, user profile is build. This user profile is generated by getting the data from item profile. As we can see in item profile, user likes circle and triangle of blue colour so user profile is also having circle, triangle and blue colour. Now we will match this user profile with the collection of different shapes available. In the shapes collection, we have pentagon of blue colour then circle of yellow colour and two square of yellow colour. So here system finds which of these shapes matches with the user profile. So here blue colour pentagon matches with the user's interest.

Jieun Son and Seoung Bum Kim proposed Content-based filtering for recommendation systems using multiattribute networks that contain attribute information about item [18]. By using all attribute based on network analysis various items are recommended and overspecialization problem is resolved. Results show that problems like sparsity and scalability are also addressed when compared with pure Content Based Filtering, Linked Open Data and Feature Weighting. For conducting experiments Movie lens dataset is used, where on a random basis 100 users are taken for experiment purpose and accuracy is also improved when compared with the above mentioned methods.

In the paper, Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems by Yolanda Blanco-Fernandez et al. solved overspecialization problem [19]. For this, hidden semantic association between user and the product are known then applying Spreading Activation technique to detect a node that is strongly connected.

There is no requirement about other user's data to make recommendations. It is easy for content based approach to recommend new items. It provides recommendation to the user with unique taste. There is no first rater problem. It also provide content feature which helps us to explain reason for an item recommended.

To find any particular feature like images or movies of any specific genre, it sometimes becomes a problem. Generally it is referred as overspecialization problem. User is never recommended anything outside user profile. It is easy to miss recommending item to user as there is not enough information about that item.

2.2 Collaborative Filtering

The concept of collaborative filtering was first introduced in 1991 by Goldberg et al. [20]. The Tapestry system applies only to smaller user groups (e.g. a single unit), and has too many demands on the user. As a proto-

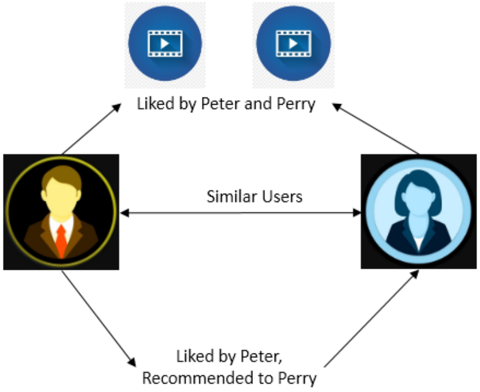

Figure 3: Collaborative Based Filtering

type of collaborative filtering recommendation system, Tapestry presents a new recommendation, but there are many technical deficiencies. Since then, there has been a scoring based collaborative filtering recommendation system, such as Grouplens, which recommends news and films. At present many ecommerce sites have been using the recommendation system such as Amazon, CDNow, Drugstor and Moviefinder etc. There is massive amount of data available. As we all know that today in this busy life no one has time to search hundreds of thousands of item and select the one which is similar to their taste. So collaborative filtering is one of the ways to filter the data and provide the relevant information in which the user is interested in. Collaborative Filtering is one of the most well known techniques for recommending items. This technique suggests relevant item to the user based on neighbour's choice. It first finds out the similarity between the user and his neighbour and then predicts the items. There can be n number of users. This technique finds the similar user from the list of user's. But the similarity between users is found out based the ratings which the users have given to the particular item. This way the approach continues and the desired result is generated. This strategy takes ratings given by user for any item from the large catalog of item catalog of ratings given by the user. This large catalog is referred as *user-item* matrix [21].

Figure 3 explains collaborative filtering with an example of recommending movie to the user. As it is clearly depicted in the diagram that user 1, user 2 and user 3 have rated movies according to their interest. Based on that user- movies matrix is created and any similarity model is applied to find the similarity between them so that recommendation can be given to user 3.

Ching-She Wu et al. have used collaborative filtering both approaches i.e. user based and item based filtering [22]. The authors have used Pearson correlation similarity for finding the similarity between users. Another algorithm named Nearest N User Neighbourhood to obtain N similar users so that they can be grouped together. For finding the item similarity, algorithm known as Log Likelyhood Similarity. The results obtain from the item based recommender, are stored into Hadoop distributed File system (HDFS). The dataset used here is obtained from Yahoo Research Web Scope Database.

Mehrbakhsh Nilashi proposed a recommendation method based on collaborative filtering using ontology and dimensionality reduction technique to improve the sparsity and scalability problem in collaborative filtering [23]. Here experiments are also given which improve the predictive accuracy and throughput of movie recommendation system. Tianqi Zhou et al. used Hadoop programming model to implement recommendation algorithm based on item based collaborative filtering. The Movielens-10M dataset is used which contain 1000 rows of rating [24].

Collaborative Filtering is further divided into two types. They are memory based and model based methods [25], [26].

2.2.1 Memory Based Method

This method is also known as Neighborhood based approach [27]. Memory-based method uses similarity measures calculated from explicit user rating to find neighbour and generate predictions [28], [29]. This type of method sees the user's interest for any item. After analyzing user's view for an item it checks the similar user who also posses the same interest as that user. So finding similar users is done by studying utility matrix. So this type of approach is mainly based on systems memory for getting prediction of similar user. So here the unknown rating of any user can be originated using the user item rating matrix (utility matrix) if we find out similar user. At last recommendation can be given [30].

Memory based approach is further classified into two types. User based approach and Item based approach [31], [32].

(A). User Based Approach This approach is also known as user-to-user filtering. In this technique, a rating matrix of n users and m items is created. For finding recommendation for a new user, this approach finds nearest neighbour using neighbour's previous rating and generates prediction for an item. In other words, recommendation is given by checking which user have similar taste [33]. Similarity between users are found using various similarity measures or by creating clusters.

Hamidreza Koohi and Kourosh Kiani proposed fuzzy C-means clustering to user based CF [34]. The rating matrix is divided into five different training sets. Then clustering techniques: K-means, SOM and fuzzy clustering are applied to create clusters for finding nearest neighbour for a new user to predict his rating and provide recommendation. By the experiments conducted, it is shown that fuzzy c-means gives better performance than K-means and SOM in terms of accuracy. In this paper it is also observed that as the number of cluster increases accuracy decreases. For experiment, 80% data is used for training and 10% for testing.

Ningning Yi et al. proposed a movie recommendation system using graph database [35]. For finding similarity, user based CF is used. As there is sparsity of the data, user item rating matrix is prefilled. The table in database used are u.user (user_id, age, sex, occupation), u.item (movie_id, name, release date, website), u.data (user_id, movie_id, score, timestamp). Different colours are used to distinguish movies. By conducting the experiments, it is seen that for highly recommended movies, node adds yellow edge and the thickness of the edge represent film recommendation. It is observed that as the radius of the nodes increases and as the edges become thick, the score of the movies is increased. For implementation, py1neo is used which is a working library with Neo4j and movielens100k dataset is used.

(B). Item Based Approach This approach is also known as item -to- item filtering. It is used to recommend any item based on related item's rating. By analyzing the rating , the users whose rating are similar for different item, only those items are available for recommendation [36]. It has been widely used by all Web giants including Netflix, YouTube etc.

Gilbert Badaro et al. [37] presented a hybrid approach which is a weighted combination of user based and item based filtering for finding the unknown ratings for an item, so that top ranked item can be recommended. The ratings of ubcf and ibcf are multiplied with the weights assigned to them to predict the rating of an item. The author showed that this combination also helps to overcome cold stat user and cold start item problem. The approach also addresses that data sparsity is reduced and accuracy is improved. For experiment purpose movie lens dataset is used.

2.2.2 Model Based Method

In Model-based method it develops a user model using ratings of each user to evaluate the expected value of unrated items [38], [39]. This method generally uses machine learning or data mining algorithm to create a model. The model is developed using the utility matrix which is build using rating given by user for any item. The model is trained by getting the information from the utility matrix. Now this model is trained using the given data to generate prediction for the users [40], [41].The model based approach is further classified into various categories. They are as Association rule mining, Decision Tree, Clustering, Artificial Neural Network, Regression etc. There are various examples working on model based approach. Some of them are Latent

Semantic methods like Latent Semantic Analysis and Latent Semantic Indexing and Dimensionality Reduction techniques like Singular Value Decomposition (SVD), Matrix Factorization etc. Model based techniques are use to solve sparsity problem occurring in recommendation system.

(A). Matrix Factorization Matrix Factorization [42], [43] is the most powerful model which generally users and movies in the matrix form. It represents rows as users or movies as columns. Other way is also used. It can also represent movies as rows and columns as users. Generally, not all the users give ratings to any item. So when we create a matrix that matrix is known as sparse matrix. Explicit user comes under this category as every user does not give rating. When we talk about implicit ratings, rating of any item is calculated by seeing what user has purchased in past. Based on which purchased history ratings are given. Matrix Factorization works using the below given formula.

$$r_{ui} = q_i^T \times p_u \tag{1}$$

Here q_i tells us about the i-th item user likes or not. It gives positive or negative feedback by assigning positive or negative value. p_u tells us about the user who is interested in particular item, r_{ui} tells us about the ratings user u gives to an item i.

Now first we have to find out factors of q_i and p_u .We have to factorize q_i and p_u in such a way that we can create user to item matrix.

We have to find factors in a way that they are very close to the actual value. So there are very less chances of error.

The formula to find out minimum values for q_i and p_u is given below.

$$min \sum_{(u,i \in k)} (r_{ui} - q_i^T \times p_u)^2 + \delta(||q_i||^2 + ||p_u||^2) \tag{2}$$

Here k is set of (u, i) pair. By using this formula, user to item matrix can be find out and we have to minimize δ in equation 2 so that error can be reduced. For minimizing value, the well known methods Stochastic Gradient Descent and Alternating Least Square are used. These two methods can be used to minimize the above equation.

In the paper named, Explainable Matrix Factorization for Collaborative Filtering is proposed by Behnoush Abdollahi and Olfa Nasraoui to compute top n recommendation [44]. The authors showed that if j item is explainable for user i then their representation in the latent space should be close to each other. Cosine similarity is used for finding nearest neighbours. The experiment gives the idea that this method perform well for generation top k recommendation.

The authors Dheeraj Bokde et al. surveyed Matrix Factorization model in Collaborative Filtering Algorithm. Matrix factorization models like SVD, PCA, PMF and NMF are discussed. The experiments concluded that Stochastic SVD increases accuracy and preciseness of user based and item based CF. It also reduces the computation cost of both CF algorithm [45].

(B). Singular Value Decomposition Singular Value Decomposition is one of the models of matrix factorization [46], [47]. This method takes A as input data matrix and disintegrates it into product of three different matrixes described as follows.

$$A_{[m \times n]} = U_{[m \times r]} S_{[r \times r]} (V_{[r \times n]})^T \tag{3}$$

A is input data matrix of size $m \times n$ e.g. m users and n movies. U is matrix of size $m \times r$ that stores left singular vectors. S contains singular values of size $r \times r$ diagonal matrix which has zeros everywhere except on its diagonal. So all these nonzero values are singular values and they are arranged in decreasing order so that largest value comes first. V is a right singular vector of size $n \times r$.

The equation 3 can be represented in compact way as follow:

$$A_k = U_k \times S_k \times V_k^T \tag{4}$$

A_k represents the closest linear approximation of the original matrix A with reduced rank k. Once this transformation is completed, users and items can be thought of as points in the k-dimensional space. SVD property states that U, S, V are unique in which U and V are column orthonormal i.e. $U = 1$, $V^T V = 1$ and S is diagonal whose entries are positive singular values and sorted in decreasing order.

M.G. Vozalis and K.G. Margaritis used SVD with demographic data [48] and applied it on user based and item based collaborative filtering. The result conducted in the paper concluded that user based CF do not produce better result and item based CF gives less error rate. It is also seen that by using SVD in recommender system tackle scalability and data sparsity problem and also improve accuracy.

There is no need of any feature selection for any particular item. As it works with any kind of item. This becomes the biggest advantage of collaborative filtering.

It usually faces first rater problem, as it cannot recommend any unrated item. It also faces popularity bias problem. It becomes a case when user may not like an item but due to its popularity that item is recommended to the user.

2.3 Hybrid Filtering

This filtering is an information filtering system that takes ratings of the movies as input from the users and then apply the collaborative and content based filtering and generate recommendation list [49]. It is a combination of the two technique i.e. collaborative filtering and content based filtering. When only the single method i.e. the collaborative filtering or content based filtering alone cannot solve the problem then hybrid filtering concept comes into picture. By using hybrid filtering many problems of collaborative filtering and content based filtering can be resolved. The problem like cold start problem in collaborative filtering is a major challenge in it. So if we apply content based filtering and then use collaborative filtering can be a solution to it. So making it hybrid can resolve the problem.

S.M.Ali et al. proposed hybrid model ,which uses genomic tags of movies with content based filtering concept [50]. It uses Principal Component Analysis with Pearson Correlation technique to reduce the tags which are redundant, so that computation complexity can be reduced. Here Movie lens dataset is used which was released on October 18, 2016.

F.Deng et al. defined an approach to calculate user's potential preference based on hybrid features like user generated features, image visual features and transforming user item rating into hybrid feature ratings [51]. The experiments performed here shows that proposed work gives better result on sparse dataset and has higher efficiency on large dataset. In [52] C. Yang et al. put forth a hybrid approach based on social similarity and item attribute. The author used collaborative filtering method combined with social similarities and genres of the movie. He used BPR-MF model to solve the sparsity problem. The proposed method works in two stages. First the BPR-MF model is used to obtain the candidate set using the ratings by training the training dataset. After finding the candidate set, the unknown ratings are predicted using the existing ratings. Then the ratings are sorted and final candidate set for each user is obtained. Each set has several top items. In the second stage the movies are recommended to user using feature selection TF-IDF method used for finding the similarity between users and movie lens dataset is used here. The result shows that by using BPR-MF shows more accurate result rather than collaborating filtering. In [53] the idea used by Priscila Valdiviezo and J. Bobadilla is that they took various ratings of the user and the demographic information like age, gender and occupation and combined them into one matrix model. Then collaborative filtering is used to find out the missing ratings. The main idea used here is to improve the overall rating prediction. Three different datasets like Movie Lens (100k), Book Crossing (BX) and Film Trust. Here MAE (Mean Absolute Error) is used to measure the performance of the proposed approach. The data sparsity problem is also solved by using demographic feature of user and the item. In [54] R. Bharti and D. Gupta have proposed a system which uses content based filtering for new users and collaborative filtering for the old user and for finding the similarity the author have used cosine and Pearson similarity and finally hive is used for storing user and movies details in the database. For experiment purpose movie Lens dataset is used.

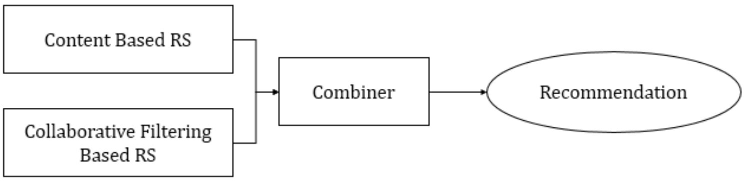

Figure 4: Hybrid Filtering

There are various categories of hybrid filtering. Hybrid Filtering is clearly understood by understanding the figure given below. As it is shown in the figure, first input is applied to collaborative filtering and content-based filtering, and then their final outcome is combined to get hybrid filtering.

2.3.1 Weighted hybridization

In this technique of hybridization, at first outcomes of the items which are recommended is generated from the list the recommendation techniques used in the particular system. The system named P- Tango known as Personalized Tango [55] used this hybrid concept. This P-Tango system consist of front end, back end and database. The user access the front end through the web browser, the back end download the articles and makes prediction. Generally the collaborative and content based filtering and weighted hybridization is applied. In short, weighted hybridization implement collaborative and content based filtering individually and then finally combine their predictions.

2.3.2 Switching hybridization

As the name suggest, switching or changing of recommendation technique is done. This switching is done based on the current situation of the system. So criteria are decided by the system to switch between the two recommendations systems. This switching approach is generally used to avoid ramp-up problem. But both, collaborative and content based filtering both faces *new user* problem. This technique was implemented by a system called Daily Learner, which uses content based filtering at start and then uses collaborative filtering. This switching hybridization method is used by [56] to deal with different cold start problems. The author used content based CAMF-CC and demographic based CAMF-CC to provide a way to cope up with cold start problem.

2.3.3 Cascade hybridization

In this technique the recommendation by one technique is generated and that recommendation results are refined or filtered by another recommendation technique to improve the recommendation system. A music recommender system developed by [57]. It acts like a middleware system which provide facility to digital audio/music libraries. It uses content based filtering then collaborative filtering to provide recommendations. This system gives recommendation of music from the same genre according to the user's query and also considers previous user's preference as well as other user's preference by collaborative filtering. The restaurant recommender EntreeC, is also a cascaded knowledge-based and collaborative recommender.

2.3.4 Mixed hybridization

As the name suggest it uses more than one recommender at a time to provide large number of recommendation together. When users wish to get immense number of recommendation simultaneously then this *mixed* recommender is used. One example showing mixed type is ProfBuilder recommender system [58] which works as an agent based recommender system for a Web site. ProbBuilder first collect the site usage information to find user's interest and a base for collaborative filtering. Then it helps the user to find relevant page of their choice by using both content based and collaborative filtering.

2.3.5 Feature Combination

Here at first some features are produced by one recommendation technique and then they are applied to another recommendation technique. So basically this approach provides a merger for collaborative and content based filtering. This way features from different recommender engine are combined and fed into one single recommendation algorithm [59].

2.3.6 Feature Augmentation

Here, the output of one recommender engine is used as input to another recommender system. The first recommender system uses ratings and extra information which are fed to the second system as input. Now the second system takes this information and also add some additional function to generate recommendations [59]. Feature Augmentation works better than Feature Combination because it adds some extra feature to the earlier recommender engine.

2.3.7 Meta Level

This is also one of the ways to combine two systems and provide recommendations. Internal or base model is generated by the recommender engine and then that model is applied as a whole to the other system. It can be a misconception that Meta level and feature augmentation are same but they are totally different from one another. In feature augmentation, some additional information with the available data are fed to the system while in meta level the model generated by one system is itself a source to be applied to the other system [60]. Here there is no need to provide extra data with the model.

It is used to deal with cold start problem, sparsity problem and grey sheep problem. Implementation cost is increased and increased complexity is also a factor.

2.4 Deep Learning Based Approaches

Jeffrey Lund and Yiu-Kai Ng [61] propsed adopted deep learning based approach in which authors have used autoencoder based collaborative filtering system. Prediction error in the propsed method has been minimized using regularization.

Traditional content based and collaborative filtering techniques require prior information like user history and habits. Performance of such systems highly depends on the initial knowledge about user. S. Kumar et al. [62] proposed sentiment analysis based hybrid approach for movie recommendation. From micro bogging websites, tweets have been collected about movie and the sentiments of the users are analyzed and the cold start problem has been nicely addressed. J. K. Leung et al. have studied the effect of user's mood on recommendation system. Authors have assigned emotional tags to movies by auto detection of affective attributes. Text based Emotion Detection Recognition model has been trained using tweet text and the leaned model is used to detect affective features [63].

T. Singh et al. [64] have used real time multi-lingual tweets to improve the performance of recommendation system. Sentiment analysis is applied on the tweets, and tweets are classified using RNN. The system achieves promising results on real time tweets.

Deep learning based approach has been proposed by J. Bobadilla et al. [65] to improve the performance of collaborative filtering. Non linear relation between predictions, reliabilities and accurate recommendation has been extracted using deep learning. The proposed architecture achieves the superior performance through three abstraction layers: real prediction errors, prediction errors and predicted ratings.

2.5 Comparison of State-of-the-Art Methods

Various techniques working on collaborative and content based filtering are discussed in the literature survey. In this particular subsection, comparison of the some of the recent techniques used for recommendation is surveyed as shown in Comparative Study

Table 1: Comparison of State-of-the-Art methods

Ref	CF/CB/H	Dataset	Method	Similarity Measures	Performance
[50]	H	Movie lens	PCA	Cosine Similarity	Cutoff = 0.5 will leave some useful tags Cutoff = 0.3 will lose some important tags
[51]	H	Movielens-1k	HFB-KNN, CFB-KNN, User KNN		Rating changes from 10 to 80 HFB-KNN, CFB-KNN performs better
[22]	CF	From, Yahoo Research Web scope database	NNUN, Log Likely hood	Pearson Correlation	AAD =0 .Prediction Accuracy = 100%
[23]	CF	Movie lens	PNN, SVD		The sparsity level of the Movie Lens dataset is 93.7% the sparsity level of the Yahoo! Web scope R4 dataset is 99.8%
[24]	CF	Movielens-10M	User Based CF, Item Based CF		HDFS makes the performance better.
[52]	H	Movie lens	BPR-MF,TF-IDF	Cosine similarity	MAE = 0.817 , RMSE= 1.037
[53]	H	Movielens 100k,Film Trust	Biased Matrix Factorization		For Movie Lens, Precision = 0.8704 Recall =0.107 For Film Trust, Precision= 0.7391 Recall = 0.0051
[54]	H	Movie lens	Map Reduce	Cosine Similarity, Centered Cosine Similarity	Using Hadoop Map Reduce Framework, movie recommendation is done fast though dataset is large.
[18]	CB	Movie lens -1M	CB-MN system, FW		Overspecialization and data sparsity is improved
[19]	CB	IMDB and BBC web server	SA		Overspecialization problem is solved and observed that when no. of cluster increases accuracy decreases.
[34]	User based CF	Movie lens with 6.3% rating available	K-means, SOM, Fuzzy clustering	Pearson Correlation	Fuzzy C-means and max average accuracy-80.44 and Fuzzy C-means and max Pearson accuracy-81.1
[35]	User based CF	Movielens-100K	Neo4j is NoSQL graph database	Euclidean distance	Larger radius of nodes and thicker edge then movie score is high
[37]	H	Movie lens	UBCF and IBCF	Pearson Correlation	Improvement of 13% over UBCF and 16% over IBCF
[44]	CF	Movie lens	EMF, NMF and PMF	Cosine	Average RMSE of EMF is 1.3411
[48]	CF	Movie lens	SVD with UBCF and IBCF	Demographic correlation	IBCF produces less error value than UBCF

2.6 Similarity Measures

There are various similarity measures to find out similarity between user and item. Some of them are discussed here.

2.6.1 Jaccard Similarity

It is the ratio of common items rated by the user to the total number of items rated by both the users. The formula to calculate Jaccard Similarity is given below [66].

$$sim(u,v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{5}$$

Here I_u and I_v are the set of item rated by the user u and v respectively.

2.6.2 Cosine Similarity

Cosine similarity [67], [68] is the measure of similarity between two non-zero vectors in the inner product space. It measures the angle between these two vectors. A cosine of two non-zero vectors can be calculated using dot products of these two vectors:

$$u.v = ||u|| \,.\, ||v|| \,.\, cos\theta \tag{6}$$

Cosine similarity is particularly used in positive space where the result is efficiently bounded in [0, 1]. Thus for two given vectors u and v, the cosine similarity, $cos\theta$ can be computed as the combination of dot product and magnitude of the vectors:

$$sim(u,v)^{cosine} = cos\theta = \frac{\sum u \; v}{||u|| \,.\, ||v||} \tag{7}$$

2.6.3 Pearson Correlation Similarity

It uses Pearson Correlation Coefficient [69], [70] to determine the similarity between users. The higher the coefficient the two users are more closely related. Formula to calculate Pearson Correlation Coefficient is given below,

$$r = \frac{\sum((u-\bar{u})(v-\bar{v}))}{\sqrt{\sum(u-\bar{u})^2 . \sum(v-\bar{v})^2}} \tag{8}$$

2.6.4 Mean Square Distance

Mean square distance [70] is calculated by the ratio of sum of the square of the difference of the items rated by the user to the common items rated by both the users. Then the Mean Square similarity is calculated by subtracting the mean square distance by 1. The formula to calculate the mean square distance is given below.

$$sim(u,v)^{MSD} = 1 - \frac{\sum i \in I_{(u,v)}(R(u,i) - R(v,i))^2}{|I_{(u,v)}|} \tag{9}$$

$R(u,i)$ and $R(v,i)$ are the ratings given by the user u and v to item i, respectively. $I(u,v)$ indicates the co-rated items of users u and v.

2.6.5 Jaccard Mean Square Distance (JMSD)

This is generated by multiplying two similarity measures, i.e. Jaccard similarity and mean square distance similarity measure [70].

$$sim(u,v)^{JMSD} = sim(u,v)^{Jaccard} . sim(u,v)^{MSD} \tag{10}$$

$$sim(u,v)^{JMSD} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} . \left(1 - \frac{\sum i \in I_{(u,v)}(R(u,i) - R(v,i))^2}{|I(u,v)|}\right) \tag{11}$$

2.6.6 Relevant Jaccard similarity (RJaccard)

$$sim(u,v)^{RJaccard} = \frac{1}{1 + \frac{1}{|I_u \cap I_v|} + \frac{|\bar{I_u}|}{1+|I_u|} + \frac{1}{1+|\bar{I_v}|}} \tag{12}$$

where, if $|I_u \cap I_v|$ = 0 then $sim(u,v)^{RJaccard} = 0$

2.6.7 Relevant Jaccard Mean Square Distance (RJMSD)

Relevant Jaccard mean square distance is obtained by multiplying relevant jaccard and mean square distance. The formula to calculate RJMSD is given below [70].

$$sim(u,v)^{RJMSD} = sim(u,v)^{RJaccard} . sim(u,v)^{MSD} \tag{13}$$

$$sim(u,v)^{RJMSD} = \left(\frac{1}{1 + \frac{1}{|I_u \cap I_v|} + \frac{|\bar{I_u}|}{1+|I_u|} + \frac{1}{1+|\bar{I_v}|}}\right) . \left(1 - \frac{\sum i \in I_{(u,v)}(R(u,i) - R(v,i))^2}{|I_{(u,v)}|}\right) \tag{14}$$

where, if $|I_u \cap I_v|$ = 0 then $sim(u,v)^{RJMSD} = 0$

2.7 Evaluation Metrics for Recommendation System

There are various measures to evaluate recommender system. These metrics are a way to find how accurate a recommender system is. Accuracy is must for any recommender system to work. Performance of any model is judge by these metrics. The evaluation metrics are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Precision, Recall, F1-Measure, Aggregate diversity.

2.7.1 Mean Absolute Error

$$MAE = \frac{1}{N} \sum_{(u,i)} |P_{(u,i)} - R_{(u,i)}| \tag{15}$$

Here $P(u, i)$ is the predicted rating for user u on item i , $R(u, i)$ is the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings [71].

2.7.2 Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_{(u,i)} - R_{(u,i)})^2}{N}} \tag{16}$$

Root Mean Square Error (RMSE) [72] puts more emphasis on larger absolute error and when RMSE is low than accuracy of recommender system is better.

2.7.3 Precision

$$Precision = \frac{Currently\ recommended\ items}{Total\ recommended\ items} \tag{17}$$

Precision [73] shows the result that are relevant , i.e. the items correctly recommended by the system.

2.7.4 Recall

$$Recall = \frac{Currently\ recommended\ items}{Total\ useful\ recommended\ items} \tag{18}$$

Recall [73] shows the results which are successfully recommended by the system.

2.7.5 F1-Measure

$$F_1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{19}$$

For finding F1-Measure [73] we have to find precision and recall first.

2.7.6 Diversity

$$Diversity = U_{u \in U}\ L_n(u) \tag{20}$$

Where u is any particular user, U is the total user in the dataset and $L_n(u)$ is the list of relevant items recommended to the user u [74], [75].

3 Dataset Descriptions

A dataset is just a unit to measure information released in a public open data repository. The researchers use the dataset to perform experiments. For this, they divide the dataset into training and testing set to obtain the desired result. A dataset corresponds to one or more database tables where every column of a table represents a particular variable and each row corresponds to a given record of the data. The model is run using the training set which is then compared with the target result. Then testing data is applied to provide estimation to the final model. There are various datasets available for movies which are widely available. The datasets like Movielens100k, Movielens-10M, Movielens-10k etc are used. Other dataset are IMDB, Netflix dataset, Film Trust dataset etc. The movie dataset contains the fields like user_id, item_id, ratings, tags, timestamps etc.

Table 2: Summary of various datasets for Recommendation system

Name	Movies	Ratings	Users	tags	URL
Movielens-100k [52]	9115	1,00,005	671	1197	http://grouplens.org/datasets/movielens/100k/
Movie lens [54]	27,278	9,999,999	69139	-	http://grouplens.org/datasets/movielens/latest/
Movielens-1M [76]	40,000	14 M	1,60,000	670,000	https://grouplens.org/datasets/movielens/
Yahoo Web scope (R4) [22]	11,915	1,11,131	7,641	-	http://webscope.sandbox.yahoo.com
Movielens-100k [53]	1682	1,00,000	943	-	http://movielens.org
Movielens-15M	17,000	10M	1,38,000	4,65,000	https://grouplens.org/datasets/movielens/15m/
Movie lens Full	58,000	1,80,000	600	1,100,000	https://grouplens.org/datasets/movielens/latest/
Movielens-100k	1700	1,00,000	1000	-	http://grouplens.org/datasets/movielens/100k/
Movielens-10M	10,000	10M	71,000	1,00,000	http://grouplens.org/datasets/movielens/10m/

4 Conclusions and Future Scopes

This paper describes different types of filtering techniques. Various uses, advantages, disadvantages are also discussed. To build an efficient recommender system a hybrid combination of different methods of recommendation is must. It is concluded that by using combination of similarity measure a better user similarity can be generated rather than using single similarity measure and efficiency of the system is also increased. One of the fact that similarity measure like RJMSD is evolved by the author and up till now it is only used in movie recommendation. The author also showed that this similarity measure is better than the other in terms of efficiency parameters. Accuracy of any recommender system can be improved if we add extra movie features. Generally, most of the papers have shown the combination of collaborative filtering and content-based filtering. By combining methods the problems related to the two methods are tried to resolve. So hybrid filtering is the most well-known technique in any recommendation system. Because using this helps to build an effective recommendation system.

There are various areas of recommendation system as discussed earlier. Various techniques are also discussed which works in giving recommendation. So, scope of any recommender system is to build a model in such a way that their user gets proper recommendation and efficiency of the system is maintained.

References

[1] J. A. Konstan and J. Riedl, ”Recommender systems: From algorithms to user experience”, *User Modeling and User-Adapted Interaction* 22(1-2):101-123, 2012. DOI: http://dx.doi.org/10.1007/s11257-011-9112-x

[2] R. Katarya and O. P. Verma, ”An effective collaborative movie recommender system with cuckoo search”, *Egyptian Informatics Journal* 18(2):105-112, 2017. DOI: http://dx.doi.org/10.1016/j.eij.2016.10.002

[3] S. H. Min and I. Han, ”Detection of the customer time-variant pattern for improving recommender systems”, *Expert Systems with Applications* 28(2):189-199, 2005.

[4] S. Wattal, Y. Hong, M. Mandviwalla, and A. Jain, ”Technology diffusion in the society: Analyzing digital divide in the context of social class”, *IEEE Proc. of 44th Hawaii International Conference on System Sciences*, 1-10, 2011. DOI: http://dx.doi.org/10.1109/HICSS.2011.398

[5] M. Goldmann and G. Kreitz, ”Measurements on the spotify peer-assisted music-on-demand streaming system”, *IEEE International Conference on Peer-to-Peer Computing*, 206-211, 2011. DOI: http://dx.doi.org/10.1109/P2P.2011.6038737

[6] P. N. V. Kumar and V. R. Reddy, ”A Survey on Recommender Systems (RSS) and Its Applications”, *International Journal of Innovative Research in Computer and Communication Engineering* 2(8):5254-5260, 2014.

[7] C. A. Gomez-Uribe and N. Hunt, ”The Netflix Recommender System: Algorithms, Business Value, and Innovation”, *ACM Transactions on Management Information Systems* 6(4):1-19, 2015. DOI: http://dx.doi.org/10.1145/2843948

[8] Ò. Celma, *Music Recommendation*, Springer, Berlin, Heidelberg, 43-85, 2010. DOI: http://dx.doi.org/10.1007/978-3-642-13287-2_3

[9] F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li ”DRN: A Deep Reinforcement Learning Framework for News Recommendation”, *Proceedings of the World Wide Web Conference*, 167-176, 2018. DOI: http://dx.doi.org/10.1145/3178876.3185994

[10] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, ”Visually-aware fashion recommendation and design with generative image models”, *IEEE International Conference on Data Mining*, 207-216, 2017.

[11] V. C. S. Heung, H. Qu, and R. Chu, ”The relationship between vacation factors and socio-demographic and travelling characteristics: The case of Japanese leisure travellers”, *Tourism Management* 22(3):259-269, 2001.

[12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, ”Recommender systems survey”, *Knowledge-Based Systems* 46(2):109-132, 2013. DOI: http://dx.doi.org/10.1016/j.knosys.2013.03.012

[13] M. Sharma and S. Mann, ”A Survey of Recommender Systems: Approaches and Limitations”, *International Journal of Innovations in Engineering and Technology* 2(2):8-14, 2013.

[14] F. Mansur, V. Patel, and M. Patel, ”A review on recommender systems”, *IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 1-6, 2017. DOI: http://dx.doi.org/10.1109/ICIIECS.2017.8276182

[15] J. Salter and N. Antonopoulos, ”CinemaScreen Recommender Agent:Combining Collaborative and Content-Based Filtering”, *IEEE Intelligent Systems* 21(1):35-41, 2006.

[16] H. Li, F. Cai, and Z. Liao, ”Content-based filtering recommendation algorithm using HMM”, *IEEE Fourth International Conference on Computational and Information Sciences*, 275-277, 2012.

[17] A. Patel, A. Thakkar, N. Bhatt, and P. Prajapati, *Survey and Evolution Study Focusing Comparative Analysis and Future Research Direction in the Field of Recommendation System Specific to Collaborative Filtering Approach*, Springer, Singapore, 155-163, 2019. DOI: http://dx.doi.org/10.1007/978-981-13-1742-2_16

[18] J. Son and S. B. Kim, ”Content-based filtering for recommendation systems using multiattribute networks”, *Expert Systems with Applications* 89:404-412, 2017.

[19] Y. B. Fernandez, J. J. P. Aris, A. G. Dolla, M. R. Cabrer, and M. L. Nores, ”Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems”, *IEEE Transactions on Consumer Electronics* 54(2):727-735, 2008.

[20] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, ”Using collaborative filtering to Weave an Information tapestry”, *Communications of the ACM* 35(12):61-70, 1992. DOI: http://dx.doi.org/10.1145/138859.138867

[21] S. Khatwani and D. M. B. Chandak, ”Building Personalized and Non Personalized Recommendation Systems”, *IEEE International Conference on Automatic Control and Dynamic Optimization Techniques* , 623-628, 2016. DOI: http://dx.doi.org/10.1109/ICACDOT.2016.7877661

[22] C. M. Wu, D. Garg, and U. Bhandary, ”Movie Recommendation System Using Collaborative Filtering”, *IEEE 9th International Conference on Software Engineering and Service Science* , 11-15, 2018.

[23] M. Nilashi, O. Ibrahim, and K. Bagherifard, ”A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques”, *Expert Systems with Applications* 92:507-520, 2018. DOI: http://dx.doi.org/10.1016/j.eswa.2017.09.058

[24] T. Zhou, L. Chen, and J. Shen, ”Movie Recommendation System Employing the User-Based CF in Cloud Computing”, *IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing* , 2:46-50, 2017.

[25] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, ”Recommendation systems: Principles, methods and evaluation”, *Egyptian Informatics Journal* 16(3):261-273, 2015. DOI: http://dx.doi.org/10.1016/j.eij.2015.06.005

[26] T. Bai, J. R. Wen, J. Zhang, and W. X. Zhao, ”A neural collaborative filtering model with interaction-based neighborhood”, *Proceedings of ACM on Conference on Information and Knowledge Management* , 1979-1982, 2017.

[27] L. T. Ponnam, S. D. Punyasamudram, S. N. Nallagulla, and S. Yellamati, ”Movie recommender system using item based collaborative filtering technique”, *IEEE International Conference on Emerging Trends in Engineering, Technology and Science* , 1-5, 2016.

[28] J. Bobadilla, F. Serradilla, and A. Hernando, ”Collaborative filtering adapted to recommender systems of e-learning”, *Knowledge-Based Systems* 22(4):261-265, 2009. DOI: http://dx.doi.org/10.1016/j.knosys.2009.01.008

[29] Xiaoyuan Su and T. M. Khoshgoftaar, ”A Survey of Collaborative Filtering Techniques”, *Advances in artificial intelligence* 2009:1-9, 2009.

[30] Delgado, Joaquin, and N. Ishii, ”Memory-Based Weighted-Majority Prediction for Recommender Systems”, *Res. Dev. Inf. Retr* 1999.

[31] J. Pareek, M. Jhaveri, A. Kapasi, and M. Trivedi, *”SNetRS: Social networking in Recommendation system”*, Springer, Berlin, Heidelberg, 195-206, 2013. DOI: http://dx.doi.org/10.1007/978-3-642-31552-7_21

[32] M. Deshpande and G. Karypis, ”Item-based Top-N Recommendation Algorithms”, *ACM Transactions on Information Systems* 22(1):143-177, 2004. DOI: http://dx.doi.org/10.1145/963770.963776

[33] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, ”A Survey Paper on Recommender Systems”, *arXiv preprint arXiv:1006.5278* 2010.

[34] H. Koohi and K. Kiani, ”User based Collaborative Filtering using fuzzy C-means”, *Measurement: Journal of the International Measurement Confederation* 91:134-139, 2016.

[35] N. Yi, C. Li, M. Shi, and X. Feng, ”Design and Implementation of Movie Recommender System Based on Graph Database”, *IEEE 14th Web Information Systems and Applications Conference* , 132-135, 2017.

[36] C.-H. Piao, J. Zhao, and L. J. Zheng, ”Research on entropy-based collaborative filtering algorithm and personalized recommendation in e-commerce”, *Service Oriented Computing and Applications* 3(2):147-157, 2009.

[37] G. Badaro, H. Hajj, W. El-Hajj, and L. Nachman, ”A Hybrid Approach with Collaborative Filtering for Recommender Systems”, *IEEE 9th International Wireless Communications and Mobile Computing Conference* , 349-354, 2013. DOI: http://dx.doi.org/10.1109/IWCMC.2013.6583584

[38] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, ”A new user similarity model to improve the accuracy of collaborative filtering”, *Knowledge-Based Systems* 56:156-166, 2014.

[39] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, ”Item-based collaborative filtering recommendation algorithms”, *Proceedings of the 10th International Conference on World Wide Web* , 285-295, 2001. DOI: http://dx.doi.org/10.1145/371920.372071

[40] M. Liphoto, C. Du, and S. Ngwira, ”A Survey on Recommender Systems”, *IEEE International Conference on Advances in Computing and Communication Engineering* , 276-280, 2016. DOI: http://dx.doi.org/10.1109/ICACCE.2016.8073761

[41] A. K. S. C. Pradhan and B. S. P. Mishra, ”SVD based Privacy Preserving Recommendation Model using Optimized Hybrid Item-based Collaborative Filtering”, *IEEE International Conference on Communication and Signal Processing* , 0294-0298, 2019.

[42] Y. Koren, R. Bell, and C. Volinsky, ”Matrix Factorization Techniques For Recommender Systems”, *Computer* 42(8):30-37, 2009. DOI: http://dx.doi.org/10.1109/MC.2009.263

[43] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk, ”Major components of the Gravity Recommendation System”, *Acm Sigkdd Explorations Newsletter* 9(2):80-83, 2007.

[44] B. Abdollahi and O. Nasraoui, ”Explainable Matrix Factorization for Collaborative Filtering”, *Proceedings of the 25th International Conference Companion on World Wide Web* , 5-6, 2016. DOI: http://dx.doi.org/10.1145/2872518.2889405

[45] D. Bokde, S. Girase, and D. Mukhopadhyay, ”Matrix Factorization model in Collaborative Filtering algorithms: A survey”, *Procedia Computer Science* 49:136-146, 2015. DOI: http://dx.doi.org/10.1016/j.procs.2015.04.237

[46] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia, ”Spectral analysis of data”, *Conference Proceedings of the Annual ACM Symposium on Theory of Computing* , 619-626, 2001. DOI: http://dx.doi.org/10.1145/380752.380859

[47] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, *”Singular Value Decomposition and Principal Component Analysis”*, Springer, Boston, MA, 91-109, 2005.

[48] M. G. Vozalis and K. G. Margaritis, ”Using SVD and demographic data for the enhancement of generalized Collaborative Filtering”, *Information Sciences* 177(15):3017-3037, 2007. DOI: http://dx.doi.org/10.1016/j.ins.2007.02.036

[49] K. N. Jain, V. Kumar, P. Kumar, and T. Choudhury, *”Movie recommendation system: Hybrid information Filtering System”*, Springer, Singapore, 677-686, 2018.

[50] S. M. Ali, G. K. Nayak, R. K. Lenka, and R. K. Barik, *”Movie Recommendation System Using Genome Tags and Content-Based Filtering”*, Springer, Singapore, 85-94, 2018.

[51] F. Deng, P. Ren, Z. Qin, G. Huang, and Z. Qin, ”Leveraging Image Visual Features in Content-Based Recommender System”, *Scientific Programming* 2018, 2018. DOI: http://dx.doi.org/10.1155/2018/5497070

[52] C. Yang, X. Chen, L. Liu, T. Liu, and S. Geng, ”A Hybrid Movie Recommendation Method Based on Social Similarity and Item Attributes”, *International Conference on Sensing and Imaging. Springer, Cham* , 275-285, 2018.

[53] Priscila Valdiviezo Díaz and J. Bobadilla, ”A Hybrid Approach of Recommendation via Extended Matrix Based on Collaborative Filtering with Demographics Information”, *International Conference on Technology Trends. Springer, Cham* , 384-398, 2018.

[54] R. Bharti and D. Gupta, *”Recommending Top N Movies Using Content-Based Filtering and Collaborative Filtering with Hadoop and Hive Framework”*, Springer, Singapore, 109-118, 2019. DOI: http://dx.doi.org/10.1007/978-981-13-1280-9_10

[55] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, ”Combining Content-Based and collaborative filters in an Online Newspaper”, *Proceedings of ACM SIGIR Workshop on Recommender Systems* , 1999.

[56] M. Braunhofer, V. Codina, and F. Ricci, ”Switching hybrid for cold-starting context-aware recommender systems”, *Proceedings of the 8th ACM Conference on Recommender systems* , 349-352, 2014. DOI: http://dx.doi.org/10.1145/2645710.2645757

[57] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, ”A Cascade-Hybrid Music Recommender System for mobile services based on musical genre classification and personality diagnosis”, *Multimedia Tools and Applications* 59(1):241-258, 2012. DOI: http://dx.doi.org/10.1007/s11042-011-0742-0

[58] A. M and A. Wasfi, ”Collecting user access patterns for building user profiles and collaborative filtering”, *Proceedings of the 4th international conference on Intelligent user interfaces* , 57-64, 1998.

[59] E. Cano and M. Morisio, ”Hybrid Recommender Systems: A Systematic Literature Review”, *Intelligent Data Analysis* 21(6):1487-1524, 2017.

[60] R. Burke, *”Hybrid Systems for Personalized Recommendations”*, Springer, Berlin, Heidelberg, 133-152, 2003. DOI: http://dx.doi.org/10.1007/11577935_7

[61] J. Lund and Y. Ng, ”Movie Recommendations Using the Deep Learning Approach”, *IEEE International Conference on Information Reuse and Integration*, 47-54, Salt Lake City, UT, 2018. DOI: http://dx.doi.org/10.1109/IRI.2018.00015

[62] Kumar, Sudhanshu, Kanjar De, and Partha Pratim Roy, ”Movie recommendation system using sentiment analysis from microblogging data”, *IEEE Transactions on Computational Social Systems* 2020.

[63] Leung, John Kalung, Igor Griva, and William G. Kennedy, ”Making Use of Affective Features from Media Content Metadata for Better Movie Recommendation Making”, arXiv preprint arXiv:2007.00636 (2020).

[64] Singh, Tarana, Anand Nayyar, and Arun Solanki, ”Multilingual Opinion Mining Movie Recommendation System Using RNN”, *First International Conference on Computing, Communications, and Cyber-Security*, Springer, Singapore, 2020.

[65] Bobadilla, Jesus, Santiago Alonso, and Antonio Hernando, ”Deep Learning Architecture for Collaborative Filtering Recommender Systems”, *Applied Sciences* 10(7):24-41, 2020. DOI: http://dx.doi.org/10.3390/app10072441

[66] M. Ayub, M. A. Ghazanfar, M. Maqsood, and A. Saleem, ”A Jaccard base similarity measure to improve performance of CF based recommender systems”, *IEEE International Conference on Information Networking* , 1-6, 2018.

[67] T. Q. Lee, Y. Park, and Y. T. Park, ”A similarity measure for collaborative filtering with implicit feedback”, *International Conference on Intelligent Computing. Springer, Berlin, Heidelberg* , 385-397, 2007.

[68] S. Bansal, C. Gupta, and A. Arora, ”User tweets based genre prediction and movie recommendation using LSI and SVD”, *IEEE Ninth International Conference on Contemporary Computing* , 1-6, 2016. DOI: http://dx.doi.org/10.1109/IC3.2016.7880220

[69] S. M. Choi, S. K. Ko, and Y. S. Han, ”A movie recommendation algorithm based on genre correlations”, *Expert Systems with Applications* 39(9):8079-8085, 2012.

[70] S. Bag, S. K. Kumar, and M. K. Tiwari, ”An efficient recommendation generation using relevant Jaccard similarity”, *Information Sciences* 483:53-64, 2019.

[71] G. Shani and A. Gunawardana, *”Evaluating Recommendation Systems”*, Springer, Boston, MA, 257-297, 2011. DOI: http://dx.doi.org/10.1007/978-0-387-85820-3_8

[72] D. Parra and S. Sahebi, *”Recommender systems: Sources of knowledge and evaluation metrics”*, Springer, Berlin, Heidelberg, 149-175, 2013. DOI: http://dx.doi.org/10.1007/978-3-642-33326-2_7

[73] G. Carullo, A. Castiglione, A. De Santis, and F. Palmieri, ”A triadic closure and homophily-based recommendation system for online social networks”, *World Wide Web* 18(6):1579-1601, 2015. DOI: http://dx.doi.org/10.1007/s11280-015-0333-5

[74] G. Adomavicius and Y. Kwon, ”Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques”, *IEEE Transactions on Knowledge and Data Engineering* 24(5):896-911, 2011. DOI: http://dx.doi.org/10.1109/TKDE.2011.15

[75] G. Adomavicius and Y. Kwon, ”Optimization-based approaches for maximizing aggregate recommendation diversity”, *INFORMS Journal on Computing* 26(2):351-369, 2014. DOI: http://dx.doi.org/10.1287/ijoc.2013.0570

[76] M. F. Aljunid and D. H. Manjaiah, *Movie Recommender System Based on Collaborative Filtering Using Apache Spark*, Springer, Singapore, 283-295, 2019. DOI: http://dx.doi.org/10.1007/978-981-13-1274-8_22