An Efficient Recommender System Technique in Social Networks Based on Association Rule Based Mining

Gypsy Nandi

Abstract*: A recommender system is an information filtering system that has become a buzzword in various areas of marketing and research such as movies, music, books, products and research articles. The main role of recommender systems is to guide users on a personal level to provide an optimum set of suggestions based on the users' taste, explicit rating of items, his/her demographic and other related valuable information. In the past decade, several approaches have been discussed for recommendation of items to online users keeping in mind the accuracy of prediction, the cold-start problem and the problem of sparsity. However, most of these existing recommender system techniques have failed to define a proper recommendation model that can be well suited for any online social network and can consider majority of the limitations when modeling real-market recommendations. In this paper, we present a novel efficient recommender system technique RecGyp and many other standard commonly used existing prediction techniques and also perform experimental evaluations to make a comparative analysis among each technique. The experiments carried out on the MovieLens 100K and the Yahoo! Webscope Movie datasets demonstrate the superior nature of the proposed RecGyp technique in solving the scalability issue and accuracy of results for recommending items to users of a social network. In addition to the traditional similarity measurements evaluation, results are also provided for three evaluation metrics, Precision, Recall and ROC, to evaluate the accuracy of all the recommender system techniques.

***Index Terms*: association rule mining, collaborative filtering, online social networks, recommender system**

I. INTRODUCTION

Recommender system (RS) is a software tool that can be used for applications or websites for providing fruitful suggestions to users. RS is often coined with the term e-commerce as recently the impact of RS in the field of e-commerce has seen a massive turn. The bulk amount of rich data generated by the ecommerce sites help a RS to provide as input for processing [13], [14]. These valuable data are mainly retrieved from customer activities such as customers' personal details, the products bought by him/her, the ratings given for a particular product as well as the kind of products viewed by him/her that signifies the taste of the customer. All these valuable inputs are analyzed to produce a catalog of recommended products for each customer. For example, in an online shopping, once a customer has bought a handloom jewelry product from an e-commerce site, immediately the customer will be suggested for a series of similar jewelry items that can match the taste of that customer.

Revised Manuscript Received on July 07, 2019.

Dr. Gypsy Nandi, Dept. of CSE & IT, Assam Don Bosco University, Guwahati, India.

There are, however, several noteworthy challenges met by all RS techniques for generation of top-N recommendations for a customer of a site which can be summarized as follows:

- If we consider a real scenario, a RS will have as its input thousands, lakhs or even millions of distinct products as well as visited customers for an ecommerce site, all of which have to be considered for providing recommendations.
- Another problem, termed as the *'cold-start'* problem, arises for the first-time customers who have not visited the ecommerce site before and hence no information can be fetched based on his/her previous activities to provide some recommendations
- The older customers, on the other hand, may have abundant amount of information stored based on the amount of purchases and ratings made by these frequently visited customers.
- The most challenging task is to generate recommendations in a real-time setup which demands that the RS technique should provide quick results in not more than half a second by also considering an optimum accuracy of recommendations.

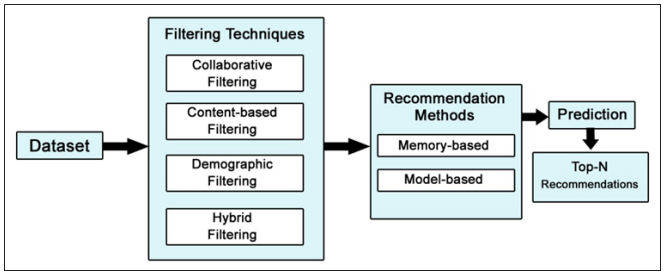

Fig 1. A typical recommender system for generating top-*N* recommendations

Figure 1 illustrates the conventional method used for recommendation of products to customers by providing top-*N* recommendations as output. To do so, the RS will initially take the dataset as input containing customer and product information. This input is then fed to any standard widely-used filtering techniques, such as collaborative filtering, content-based filtering, demographic filtering and hybrid filtering [15] (discussed next). After the process of filtering is over, any memory-based or model-based method of recommendations is applied to make predictions of items for a list of users and finally top-N recommendations are given as output for each user.

A. *Collaborative Filtering*

Collaborative filtering [13], [20] is considered to be the most commonly used filtering techniques for RS. This technique works on the principle of finding the similarity between users or items based on the items ranked or rated in the past. *k*-Nearest Neighbors (*k-NN*) algorithm is one of the most commonly implemented approach for collaborative filtering in RS that analyses the opinion of like-minded users for generating accurate recommendations. *k-NN* algorithms used in RS can be either item-based *k-NN* or user-based *k-NN*, each of which uses different concepts to generate top-*N* recommendations for users. Both item-based and user-based collaborative filtering techniques can use either memory-based approach or model-based approach for making recommendations. Memory-based methods form a matrix consisting of users' ratings for items [5]. This matrix is used to calculate similarity metrics by finding the distance between two items or two users. Model-based methods, on the other hand, build a model by forming a user-ratings matrix that helps in finding the group of similar users. There are many well-known approaches used for model-based methods such as matrix factorization, Bayesian classifiers, genetic algorithms, neural networks, and so on. Other than *k-NN*, association rule mining can also be used as an effective mining method in case of collaborative filtering [20]. In association rule mining, main emphasis is given on generating important relationships between items by focusing on those items that frequently appear together in the dataset being considered. Hence, analysis of relationships between items is done to lessen the data sparsity problem which is a critical issue of neighborhood-based algorithms. In such a case, the ratings or purchases of items by users are considered as transactions that serve as input for the RS. Association rules are formed based on the input provided and ranking of items are done based on the confidence measure.

B. *Content-based Filtering*

Content-based filtering [18] works on the principle of generating recommendations for users based on the study of choices made by the users in the past. For instance, if a user has recently purchased a set of books of type fiction, the RS should recommend more books of type fiction to that user. Thus, content- based filtering pays attention to mainly what items a user has already bought and/or which items the user has recently viewed or ranked. Content-based filtering technique is a domain-dependent technique as it lays emphasis on the attributes of items for generation of predictions [2]. This technique is best-suited for recommendation of documents such as publications, news or web pages. The models used for this work to find the similarity between documents are varied, such as the vector space model which may use Term Frequency/Inverse Document Frequency (TF/IDF) or probabilistic model which may use neural networks, Naïve Bayes classifier or decision trees. Recommendations are generated by learning the underlying model with either machine learning techniques or statistical analysis.

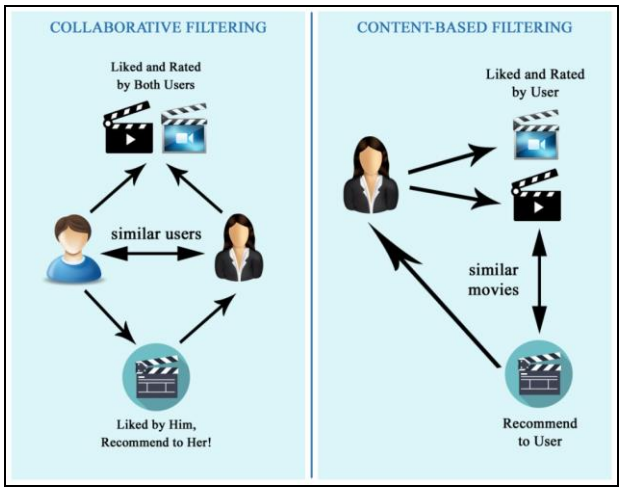

Fig 2. Basic principle behind collaborative filtering and content-based filtering

Figure 2 illustrates the basic principle behind collaborative and content-based filtering techniques for RS. The main advantage of content-based filtering when compared to collaborative filtering approach is that recommendations are generated for users even without considering the profile of other users. That is, this approach does not rely on other users' ratings of items for providing recommendations. It can manage generating results with just the basic amount of information that can be easily retrieved for each item. However, the greatest challenge in content-based filtering technique is the task of the system to congregate information about each user as well the items' metadata and model these descriptive features to satisfy the challenges of recommending items based of the congregated information. That is, for recommendations using content-based filtering, rich description of items is required along with well-organized user profile [24]. This is often termed as limited content analysis. Hence, it can be concluded that the effectiveness of content-based filtering solely depends on the availability of descriptive data.

C. *Demographic Filtering*

Demographic filtering [17], [22] technique, as the name suggests, use various demographic information of users to generate recommendations. A dataset may contain various demographic information of users, such as gender, age, area code, education, occupation, income level and marital status. These information are content-rich and helps in forming clusters of users to which similar items can be recommended. The main task in case of demographic filtering technique is to categorize users that are demographically similar to a targeted user for which recommendations are to be made. However, extracting demographic information can be a challenging task as the users are usually not interested to mention or reveal all their demographic information in social apps or websites. Figure 3 illustrates how RS work based on the concept of demographic filtering approach. Initially, as can be seen from Figure 3, the target user for which recommendations are to be generated is clustered with the group of other similar users by matching similar demographic-based data found from user profiles. Also, the neighborhood of the target user is matched with the similar users based on the

availability of the ratings of items. Finally, recommendation of top-N items are provided to target user using the user demographic similarity results as well as the selected neighborhood.

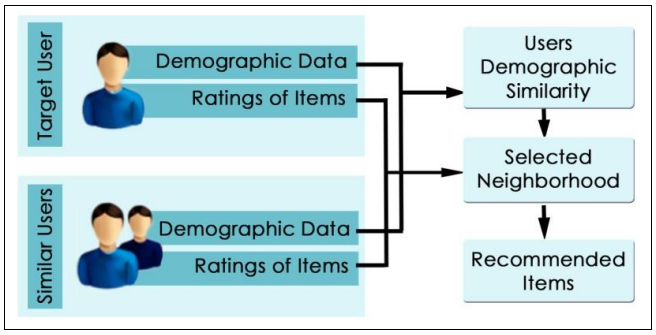

Fig 3. Recommendation of items using demographic-based approach [27]

D. *Hybrid Filtering*

The recent development in the growth of RS has revealed that hybrid approaches of RS can be more effective in some cases for accurate recommendations. Hybrid filtering [4], [25] approach has become a common thread in RS research to enable recommendation techniques to be combined to attain peak performance. This is so because all the standard RS techniques have their own strengths and weaknesses which can be studied in-depth to combine the strengths of each individual RS techniques in several ways. The most usual approach in hybrid filtering is to combine collaborative filtering with some other filtering technique in an attempt to avoid the 'ramp-up' problem [16], [21] faced in collaborative filtering approach. The incredible growth in the use of online sites has paved a way for the researchers to carry out in-depth studies in social network analysis and mining. The recommender system techniques used in social networks is one such vital and hot area of research that is has gained tremendous popularity and new approaches are hence emerging for better results. Our proposed work in this paper can be summarized as follows:

- This paper initially discusses the four standard filtering techniques used in recommender systems to generate the top-N recommendations for users. These filtering techniques are the most conventional and common method used when dealing with large datasets of social networking sites.
- The paper then discusses about several conventional approaches used in RS for finding a similarity measure between two users or two items. A brief discussion on the *weighted sum* method is done for prediction computation of items for targeted users. Also, the various standard evaluation metrics have been explained that allow results to be tested to provide directions for further improvement
- The paper also discusses a new recommender system technique namely "*RecGyp"* that aims to provide significantly better results in terms of more correct recommendations for users. Several extensive experiments are performed on two real-time datasets (*MovieLens 100K* and *Yahoo! Webscope Movie*) to produce results that prove that the "*RecGyp"* technique can prove more efficient in dealing with accuracy, sparsity and run-time complexity issues.
- Finally, we conclude that the novel "*RecGyp"* recommender system technique can be considered as a base model for providing recommendations to users and further studies can be carried out in this area to deal with complex networks.

The rest of the paper is organized as follows. In Section 2, a discussion on standard similarity measures, prediction computation and evaluation metrics are given. Section 3 discusses about the novel recommender system technique *RecGyp* which aims to provide better result than the standard existing recommender system techniques. Section 4 illustrates the experimental results by comparing the various predictions of recommendations made by the existing techniques with the novel technique. A conclusion of the paper and discussion on the scope for future work is given in Section 5.

II. RECOMMENDER SYSTEM COMPUTATIONAL TECHNIQUES

There are many standard similarity measures, prediction computation and evaluation metrics used for recommender system techniques. The effectiveness of a recommendation algorithm depends on the similarity measures and the computational techniques used which need to be thoroughly analyzed.

A. *Similarity Measures Computation*

There are several conventional approaches used in collaborative filtering of RS for finding a similarity measure (SM) between two users or two items. The similarity value that ranges between -1 and +1 indicates the amount of strength of connections between two users or two items. A value of -1 indicates that the two users or the two items are totally dissimilar. A value of 0 indicates that the two users or the two items are independent of each other. A value of +1 indicates that the two users or the two items are totally similar to each other. Few such similarity measures include *Pearson Correlation Coefficient, Cosine, Adjusted Cosine, Constrained Correlation* and *Mean Squared Difference*.

i. **Pearson Correlation Coefficient (PCC) Similarity Measure** - The PCC is the most extensively used SM and it is considered as a benchmark for collaborative filtering [7]. This SM can be used for finding the similarity between two users or two items. The similarity between two users, u_1 and u2, is determined using the formula as shown in equation 1. Here, $r_{u1,i}$ and $r_{u2,i}$ denote the ratings given by user u1 and user u_2 for item 'i' respectively. Also, $\bar{r}_{u1}$ and $\bar{r}_{u2}$ denote the average ratings of items for users u1 and u2 respectively.

$$sim(u1,u2) = \frac{\sum_{i \in I}(r_{u1,i} - \bar{r}_{u1})(r_{u2,i} - \bar{r}_{u2})}{\sqrt{\sum_{i \in I}(r_{u1,i} - \bar{r}_{u1})^2} \cdot \sqrt{\sum_{i \in I}(r_{u2,i} - \bar{r}_{u2})^2}} \quad (1)$$

ii. **Adjusted Cosine Similarity Measure** - Adjusted Cosine SM [7] is a modified version of vector-based similarity. In this measure, the similarity between two users, u1 and u2, is calculated as shown in

iii. equation 2. Here, $\bar{r}_i$ indicates the average rating of item 'i'.

$$sim(u1,u2) = \frac{\sum_{i \in I}(r_{u1,i} - \bar{r}_i)(r_{u2,i} - \bar{r}_i)}{\sqrt{\sum_{i \in I}(r_{u1,i} - \bar{r}_i)^2} \cdot \sqrt{\sum_{i \in I}(r_{u2,i} - \bar{r}_i)^2}} \quad (2)$$

iv. Cosine Similarity Measure Cosine Similarity [7] is also known as vector-based similarity where the two items or the two users are assumed as vectors. In Cosine SM, the similarity between two users, u1 and u2, is determined as shown in equation 3. Here, r_i indicates the average rating of item i.

$$sim(u1,u2) = \frac{\sum_{i \in I}(r_{u1,i} \cdot r_{u2,i})}{\sqrt{\sum_{i \in I}(r_{u1,i})^2} \cdot \sqrt{\sum_{i \in I}(r_{u2,i})^2}} \quad (3)$$

B. Prediction Computation

The final step in collaborative filtering approach of RS is calculation of predictions. Generally, prediction of ratings of a user-item pair is done using the *weighted sum* method. In this method, all the list of items similar to a target item is chosen and then from that list of items, only those items are considered which have been rated by the active user. Finally, a prediction of rating $P_{u,i}$ is calculated for a non-rated item *i* for the active user *u* by considering the sum of similarities of all items N as shown in equation 4. The predicted values using this weighted sum method serve as an output for a RS technique.

$$P_{u,i} = \frac{\sum_N (S_{i,N} * R_{u,N})}{\sum_N (|S_{i,N}|)} \quad (4)$$

C. Evaluation Metrics Computation

Research for quality measures of output generated by a RS has been extensively carried out in parallel with research in RS techniques. Evaluation metrics make it possible for comparison of numerous solutions generated by different RS techniques for the same dataset and evaluation of the most accurate results. All these evaluation metrics allow results to be tested which, in turn, provide directions for further improvement. The quality measures that are most often used by RS are prediction evaluations, evaluations for recommendation as sets, and evaluations for recommendations as ranked lists. Evaluation metrics are mainly classified into prediction accuracy metrics, classification accuracy metrics and rank accuracy metrics [1].

i. Prediction Accuracy Metrics - Prediction Accuracy Metrics, such as Mean Absolute Error (MAE), Root of Mean Square Error (RMSE), and Normalized Mean Average Error (NMAE), refer to the amount of accuracy achieved by a RS in predicting the actual rating of items. A brief description of each of these prediction accuracy metrics is given below:

- **Mean Absolute Error (MAE)** - MAE computes the average deviation in the predicted rating versus the true rating. The lower the value of MAE, the more correctly the RS predicts user ratings. Let *r(i,s)* be the true ratings of items, and $r^p(i,s)$ be the ratings predicted by a RS. Let $W = \{(i,s)\}$ be a set of user-item pairs for which the predictions are made by a RS. Then, the mean absolute error, denoted by $|\bar{E}|$, is defined as shown in equation 5.

$$MAE = |\bar{E}| = \frac{\sum_{(i,s) \in W} |r^p(i,s) - r(i,s)|}{|W|} \quad (5)$$

- **Root Mean Square Error (RMSE)** - RMSE, also known as root mean square division (RMSD), is a variant of MAE. RMSE corresponds to the sample standard deviation of the differences between predicted values of items and observed values of items and places more emphasis on larger deviation. RMSE is defined as shown in equation 6.

$$RMSE = |\sqrt{\bar{E}^2}| = \sqrt{|\bar{E}^2|} = \sqrt{\frac{\sum_{(i,s) \in W} (r^p(i,s) - r(i,s)^2}{|W|}} \quad \mathbf{(6)}$$

- **Normalized Mean Average Error (NMAE)** - NMAE normalizes MAE by using the range of available rating values. Let us consider r_{min} and r_{max} to be the smallest and the largest possible ratings respectively. Then, the normalized mean average error is defined as shown in equation 7.

$$NMAE = \frac{|\bar{E}|}{r_{max} - r_{min}} = \frac{1}{|W|} \frac{\sum_{(i,s) \in W} |r^p(i,s) - r(i,s)|}{r_{max} - r_{min}} \quad (7)$$

ii. Classification Accuracy Metrics - Classification accuracy metrics, such as Precision, Recall and Receiver Operating Characteristic (ROC), refer to the frequency with which a RS makes correct and incorrect decisions regarding choice of items. Under this category, we can classify each recommendation as:

- **TP (True Positive)** – when an accurate item is recommended to user
- **TN (True Negative)** – when an inaccurate item is not recommended to user
- **FN (False Negative)** – when an accurate item is not recommended to user
- **FP (False Positive)** – when an inaccurate item is recommended to user

- **Precision** – Precision is a measure of quality or exactness. It specifies the proportion of relevant recommended items from the total number of items actually recommended for a user. For a RS, a precision score of 1.0 for a user means that every item predicted for recommendation belongs to the actual list of recommended items for a user. The measure of precision can be calculated as shown in equation 8.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

- **Recall** – Recall is a measure of quantity or completeness. It specifies the proportion of relevant recommended items from the total number of relevant items actually available in the test dataset. The measure of recall can be calculated as shown in equation 9.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

The precision and recall values depend a lot on the number of rated items per user and so these values

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

should not be assumed as absolute measures of evaluation measurements of RS. These two evaluation metrics can rather be used as an effective measure to compare different RS algorithms for the same dataset. F-measure is yet another evaluation measure used in RS that allow combining of the measures of precision and recall by means of the relation shown in equation 10.

$$\text{F-measure} = \frac{2.precision\ .\ recall}{precision + recall} = \frac{2.TP}{2.TP+FN+FP} \qquad (10)$$

- **Receiver Operating Characteristic** (ROC) – ROC is a graphical measure that uses two metric, namely the *True Positive Rate (TPR)* and the *False Positive Rate (FPR)*. The more the area under ROC curve, the more accurately the RS predicts user ratings. The measures of TPR and FPR are calculated as shown in equations 11 and 12:

$$TPR = \frac{TP}{TP+FN} \qquad (11)$$

$$FPR = \frac{FP}{FP+TN} \qquad (12)$$

On the vertical axis of the ROC graph, the TPR values are plotted which indicates the number of items recommended related to the total number of relevant items. On the other hand, the corresponding FPR values are plotted on the horizontal axis which indicates the ratio between positively misclassified items and all the non-relevant items.

iii. Rank Accuracy Metrics

Rank accuracy metrics, such as the prediction-rating correlation and half-life utility measure, assess the accuracy of the ordering of items performed by the RS. Thus these metrics evaluate the ability of a RS in recommending an ordered list of items to a user, giving primary importance to the order in which items are recommended. For example, if a RS recommends items I_2, I_6 and I_4 in the order specified, however the testing dataset reveals the order to be I_6, I_2 and I_4. In such a case, though all the items were correctly predicted, a penalty is taken into consideration for not predicting the correct order of items.

III. THE NOVEL *RECGYP* RECOMMENDER SYSTEM TECHNIQUE

In this section, a novel approach for RS has been proposed, namely the *RecGyp* technique, and an analysis of the steps of the proposed algorithm is made to understand the detailed approach of the technique. Also, a comparative analysis of this novel technique is made with some of the standard existing collaborative filtering techniques of RS.

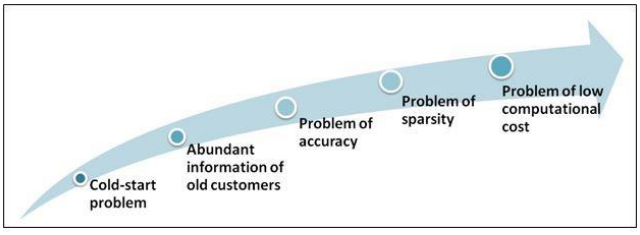

Fig 4. Various limitations of recommender system models

The recent developments in standard recommender system techniques have emphasized on providing recommendations based on several hybrid approaches. However, to the best of our knowledge, none of these existing RS techniques have defined a proper recommendation model that can be well suited for any OSN and can consider majority of the limitations (shown in Figure 4) when modeling real-market recommendations.

A. *Outline of the RecGyp Recommender System Technique*

Keeping in mind the sparsity and accuracy issues while dealing with bulk amount of items in a dataset for an e-commerce site, the novel *RecGyp* RS technique is a novel approach proposed that uses clustering and association rule mining to develop an efficient RS. The concept of clustering has been used keeping in mind the items that can be grouped based on the types and features of items [23]. Thus, clustering helps in classifying datasets and forming groups of items that may form "similar tastes". For example, if we use movies as items for our sample data, all the items can then be grouped according to a range of features or attributes such as, genres, director, actor, etc. Given in Figure 5 are the four clustered lists of movies that are formed based on their genres. The four clusters are named as C_1, C_2, C_3 and C_4 that broadly consist of drama-centered family movies, action- centered adventure movie, children movie and crime related movies, respectively. In the figure, a total of eighteen genres of movies are considered (as found in *MovieLens* dataset [11]) and similarities of genres are limitized within these four clusters that are formed. Now, based on the ratings of movies provided in the dataset for a particular user, the top-three genres are chosen, say, *animation*, *romance* and *comedy*. Since these three genres fall in clusters C_1 and C_3, only those non-rated movies that belong to these two clusters for the user are considered. This, in turn, reduces the number of search for all non-rated items for a user that exist in the entire dataset.

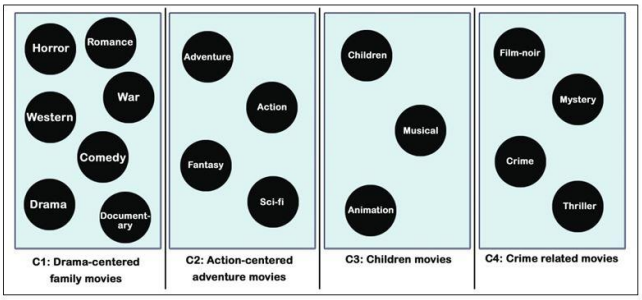

Fig 5. Clusters formed based on genre similarities of movies

Also, as has been previously discussed, association rule mining is considered as an effective mining method in case of collaborative filtering. In such a case, the ratings or purchases of items by users are considered as transactions that serve as input for the RS. Combining clustering and association rule mining results in a novel RS approach named *RecGyp* that has been experimented and the comparative results have been discussed in the section 4.

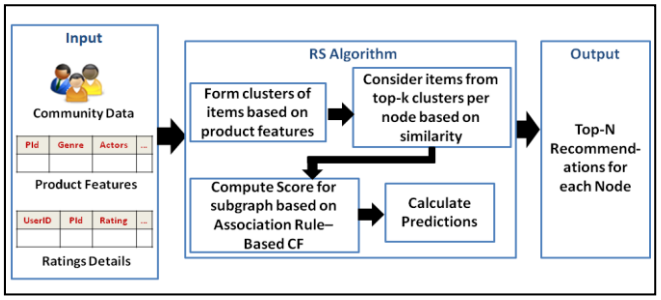

Fig. 6. The proposed framework for generating top-*N* recommendations in *RecGyp* technique

As can be seen from Figure 6, several related datasets - the *community data*, *product features* and *explicit rating details* - are to be provided as input for our novel *RecGyp* algorithm. In order to deal with accuracy, clusters of items are formed based on available item features. To resolve the low computational cost issue, the *RecGyp* algorithm considers items from top-*k* clusters per node based on similarity. Here, the minimum value of *k* is 2 and the maximum value of *k* can be one less than the total number of clusters. Next, a computation of similarity and prediction scores is done for only the smaller-sized sub- graph compared to the entire social network graph that is served as input for the algorithm. This lessens the sparsity problem which is a major issue in dealing with recommender systems. Hence, the main aim of our *RecGyp* algorithm is to handle three major issues related to recommender system techniques in OSNs namely, accuracy, low computational cost and sparisty.

B. *The RecGyp Algorithm*

Algorithm 1 explains the novel *RecGyp* approach that uses the concept of clustering and quantitative association rule mining to display top-*N* recommendations. Initially, the lists of items are considered to form clusters $C=\{C_1,C_2,C_3, \ldots, C_n\}$ (as explained in section 3.1). The items are clustered based on one or two particular features provided in the dataset. For instance, all the movie items can be clustered based on genres as explained in Figure 3. Then, as mentioned in Step 1 of Algorithm 1, for an active user *U* for whom recommendations are to be made, find the top-*N* features based on the rated items. For instance, let us assume that *animation* type of movies is mostly rated by user *U*, followed by *romance* and *comedy*. Thus, the top-3 rated genres for user *U* are *animation, romance and comedy*. These top-3 names of genres are stored in a list *L1*. (Step 3). Next, the corresponding clusters are found in which these top-*N* genres belong to, namely clusters C_3 and C_1, and these chosen clusters are stored in set *C* (Steps 4-5). Now, only those non-rated items of the active user are stored in a list *L2* that belong to either cluster C_3 and C_1 (Steps 6-7). Steps 1 to 7 of the algorithm focus on the clustering approach and by doing so, the number of non-rated items for an active user *U* that is chosen for consideration in the subsequent steps is greatly reduced. This leads to low computational cost compared to many standard RS techniques. The ratings predicted by following association rule mining, as mentioned in Algorithm 1, are the assumed ratings used to form a matrix of values for all user-item pairs. The resulting matrix is named as *'pseudo fully filled'* ratings matrix and it is used to find the actual predicted values of non-rated items for an active user by using a neighborhood based method. This resultant matrix generated solves the sparsity problem of the input dataset as all the values are now filled up in the user-item matrix that is being considered. Lastly, the *weighted sum* method (mentioned in equation 4) can be used for calculation of predictions for ratings of items for an active user. These predicted values calculated using the weighted sum method serves as an output for the novel *RecGyp* RS technique.

Algorithm 1: *RecGyp* Recommender System

1. **For** each active user *U* and rated item *j* with rating R_{uj}
   Count the type of feature that *j* belongs to
2. **End For**
3. Store the top-n features in list *L1*
4. **For** each feature *f* in L1
   Store corresponding cluster *C* which belongs to
5. **End For**
6. **For** each non-rated item *I1* ∈ *C*
   Store in list *L2 item I1*
7. **End For**
8. *Compute equally sized intervals* Inte{r_1, r_2, …, r_n}
9. **For** *each active user* U *and rated item* I2 *with rating* R_{UI2}
   Inte(R_{UI2}) ← interval that R_{UI2} belongs to
10. **For** *each active user* U *and each non-rated item* I1 *(I1* ∈ L2*)*
    For each two items I1 and I2 *(I1* ≠ *I2)*
    For each rating *R1* ∈ *Inte*{r_1, r_2, …, r_n}
    Compute $conf_{I1,I2}$
    End For
    End For
11. **End For**
12. **For** *each active user* U *and non-rated item* I1 *(I1* ∈ L2*)*
    Calculate prediction value r_o = *arg max* $\sum conf_{Ir}$
13. **End For**

In the few steps of the algorithm (Steps 8-13), the concept of association rule mining for collaborative filtering is used. In Step 8 of Algorithm 1, the ratings are mapped and discretized to transactions. For this, the entire rating range of items are divided into equally sized intervals *Inte*$\{r_1, r_2, \ldots, r_n\}$. This can be done by using standard strategies like equi-width binning strategy to divide the value of ratings into equal-sized intervals. Then, for each rated item for a user *U*, *Inte*($R_{a,j}$) is calculated based on the interval that $R_{a,j}$ belongs to (Step 9). The confidence value between a non-rated item *I1* and an item *I2* for an active user U is then calculated and stored (Steps 10-11). Here, the measure of confidence gives the similarity value between two items *I1* and *I2*. This is done by considering each rating interval for the non-rated item *I1* and measuring the confidence between *I1* and a rated item *I2* for user U. Lastly, (in Steps 12-13) the prediction value r_0 for an active user *U* and a non-rated item *I* is calculated based on the grand confidence score as shown in equation 13.

$$r_0 = \arg\max_{r} \sum conf_I \qquad (13)$$

C. *Complexity Analysis of the RecGyp Algorithm*

The novel *RecGyp* RS algorithm consequently gives a much faster execution time compared to the user-based collaborative filtering algorithms and the association-rule based algorithm. This is so because the *RecGyp* algorithm considers study of only few clusters of items for a particular user or node. This results in consideration of only limited number of non-rated items out of several innumerable non-rated items to be considered for a particular user. Hence, the *RecGyp* algorithm gives a better time complexity compared to the other standard existing RS techniques. In fact, experimental results carried out for several RS techniques indicate that the running time performance of the proposed technique is much faster compared to several other discussed existing RS techniques. Hence, the proposed *RecGyp* algorithm largely solves the scalability issue and also yields better results in recommending non-rated items for the RS problem.

IV. EXPERIMENTAL EVALUATIONS

This section discusses about the various experiments conducted for recommender systems using two real-time datasets. We have provided the data set information and the detailed results to arrive at a conclusion by comparing various such results.

A. *Datasets Used*

For carrying out the experiments, two publicly available real-world datasets were used that consists of ratings of movies by users. The two datasets are the *MovieLens 100K* dataset [11] gathered in the *GroupLens* Research Project of the University of Minnesota and the *Yahoo! Webscope* movie rating dataset [12]. The statistical information of both the datasets is summarized in Table 1.

Table 1. Statistics of the *MovieLens 100K* and the *Yahoo! Webscope Movie* datasets

Dataset	*MovieLens100K*	*Yahoo! Webscope Movie*
#Users	943	7,642
#Items	1,682	11,818
#Ratings	1,00,000	2,21,367
Range of Ratings	1-5	1-13
#genres for movies	18	20

B. *Experimental Results*

Experiments have been conducted using the above mentioned *MovieLens 100K* dataset and the *Yahoo! Webscope movie* dataset, for various RS techniques namely, the user-based collaborative filtering using *Pearson correlation coefficient (PCC) method*, the user-based collaborative filtering using *Adjusted Cosine method*, the user-based collaborative filtering using *Cosine method*, the *Association Rule-based (ARB) RS technique* and the novel *RecGyp technique*. Recommendation of top-*N* items for 100 random users from the dataset has been targeted and accordingly all results are generated for 100 such users. To conduct all these experiments, each of the three entire dataset was separated into training and testing datasets consisting of 60% and 40% non-redundant records respectively. Also, it has been checked that all the core nodes are included in the training dataset (here *'core'* refers to those users who have provided a minimum of five explicit ratings of movies). The three evaluation metrics, *Precision*, *Recall* and *ROC*, have been used to evaluate the accuracy of all the RS techniques. These metrics are mainly used as an effective measure to compare different RS algorithms for the same dataset. Figure 7 shows the precision results of several experiments conducted using the *MovieLens 100K* dataset and the *Yahoo! Weboscope movie* dataset. Again, Figure 8 shows the recall results of several experiments conducted using the *MovieLens 100K* dataset and the *Yahoo! Weboscope movie* dataset. It can be observed from all the results that the values of *Precision* and *Recall* for top-N recommendations (values of *N* are varied from 5, 10 and 15) of the novel *RecGyp* RS technique is more than those of *Pearson Correlation Coefficient (PCC)*, *Adjusted Cosine*, *Cosine*, and *Association Rule-based (ARB)* methods. This means that the *RecGyp* RS technique provides better accuracy than the rest of the standard existing RS techniques mentioned above. Thus, from the experimental results performed on both the *movie* datasets, it can be concluded that the proposed *RecGyp* RS technique has the potential to provide better performance than the four other discussed existing state-of-the-art RS techniques.

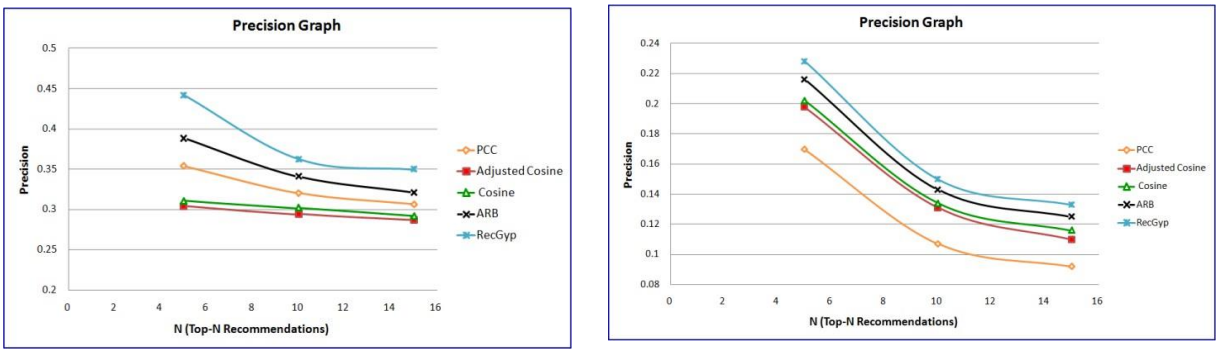

Fig 7. Precision results for 100 random users of (a). *MovieLens 100K* dataset (b). *Yahoo! Weboscope movie* dataset

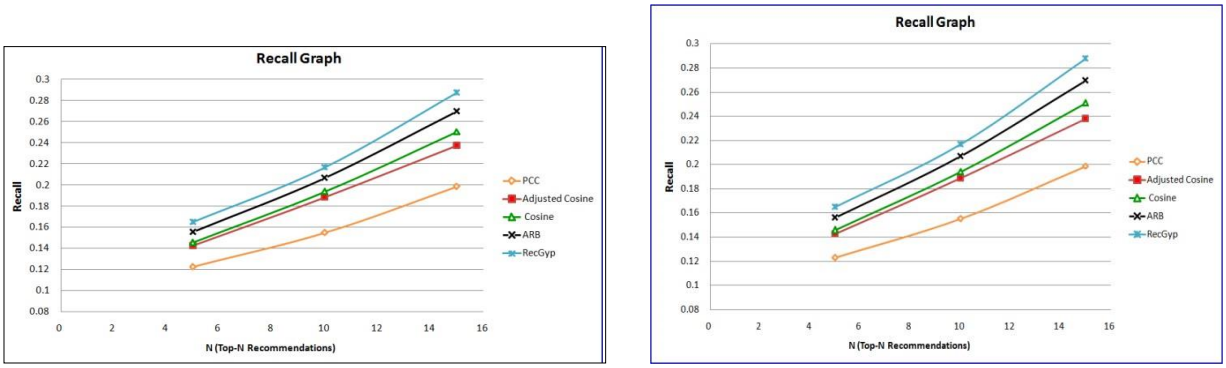

Fig 8. Recall results for 100 random users of (a). MovieLens 100K dataset (b). Yahoo! Weboscope movie dataset

Table 2. F1-measures for top-*N* recommendations of various RS techniques for the *MovieLens 100K* dataset

			Top-5 Recommendations			Top-10 Recommendations			Top-15 Recommendations		
Sl No	**RS Technique Used**		**P**	**R**	**F1 Score**	**P**	**R**	**F1 Score**	**P**	**R**	**F1 Score**
1	User-based Collabo-rative Filtering using:	Pearson Correlation method	0.354	0.101	**0.158**	0.320	0.183	**0.235**	0.306	0.262	**0.282**
2		Adjusted Cosine method	0.304	0.087	**0.133**	0.294	0.168	**0.212**	0.287	0.245	**0.263**
3		Cosine method	0.310	0.088	**0.136**	0.302	0.172	**0.219**	0.292	0.250	**0.269**
4	Quantitative Association Rule-Based RS		0.388	0.111	**0.172**	0.341	0.195	**0.246**	0.321	0.275	**0.295**
5	*RecGyp*		0.442	0.126	**0.197**	0.362	0.207	**0.264**	0.350	0.299	**0.324**

It can also been seen that the *MovieLens 100K* dataset contains comparatively more explicit ratings of items as compared to the *Yahoo! Weboscope movie* dataset. This means that the latter dataset is much more sparse than the *MovieLens 100K* dataset. However, the novel *RecGyp* RS technique proved suitable and more accurate for both densely-edged and sparsely-edged ratings. This technique also proved efficient in dealing with the other major issues related to recommendations of items in social networks such as performance consideration and scalability issue. Accordingly, the values of F1-measure based on the precision and recall values calculated for different RS techniques are tabulated in Table 2 for *MovieLens 100K* dataset and Table 3 for *Yahoo! Weboscope movie* dataset. The F1-measure is a measure of a test's accuracy and is interpreted as a weighted average of the Precision (P) and Recall (R)

values. It reaches its best score at 1 and worst score at 0. As can be seen from both the *Tables*, the novel clustered and association rule-based RS, namely *RecGyp*, outperforms the other existing standard RS techniques based on the highest values of F1-score.

Table 3. F1-measures for top-*N* recommendations of various RS techniques for the *Yahoo! Webscope movie* dataset

Sl No	RS Technique Used		Top-5 Recommendations			Top-10 Recommendations			Top-15 Recommendations		
			P	R	F1 Score	P	R	F1 Score	P	R	F1 Score
1	User-based Collabo-rative Filtering using:	Pearson Correlation method	0.170	0.123	**0.1427**	0.107	0.155	**0.1262**	0.092	0.199	**0.0629**
2		Adjusted Cosine method	0.198	0.143	**0.1661**	0.131	0.189	**0.1549**	0.110	0.238	**0.1505**
3		Cosine method	0.202	0.146	**0.1695**	0.134	0.194	**0.1581**	0.116	0.251	**0.1586**
4	Quantitative Association Rule-Based RS		0.216	0.156	**0.1812**	0.143	0.207	**0.1688**	0.125	0.270	**0.1711**
5	*RecGyp*		0.228	0.165	**0.1915**	0.150	0.217	**0.1773**	0.133	0.288	**0.1819**

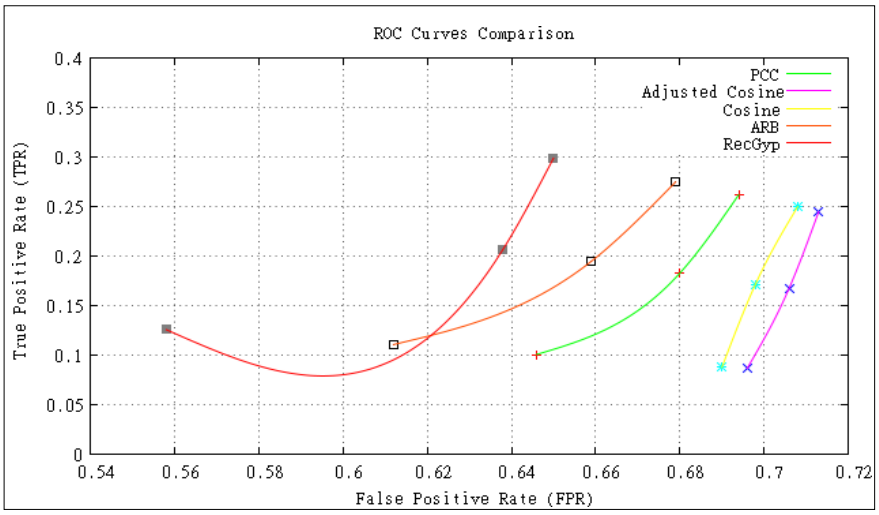

Fig 9. Comparing ROC curves for various RS techniques for the *MovieLens 100K* dataset

Figure 9 and Figure 10 illustrate the ROC curves for the experimented standard RS techniques (discussed in the previous sections) for the *MovieLens 100K* and *Yahoo! Weboscope movie* datasets respectively. The ROC curve uses two values, namely the *False Positive Rate (FPR)* on the x-axis of the graph and the *True Positive Rate (TPR)* on the y-axis of the graph. The TPR value indicates the ratio between the total number of accurate items predicted compared to the total number of items available in the test dataset. The FPR value indicates the ratio between the total number of inaccurate items predicted compared to the total number of items recommended. As mentioned before, the more the area under ROC curve, the more accurately the RS predicts user ratings. It can be clearly seen from both the figures (Figure 9 and Figure 10) that the novel *RecGyp* technique covers more area under the ROC curve and this indicates more accuracy of predictions for recommending top-*N* items for users in a social network.

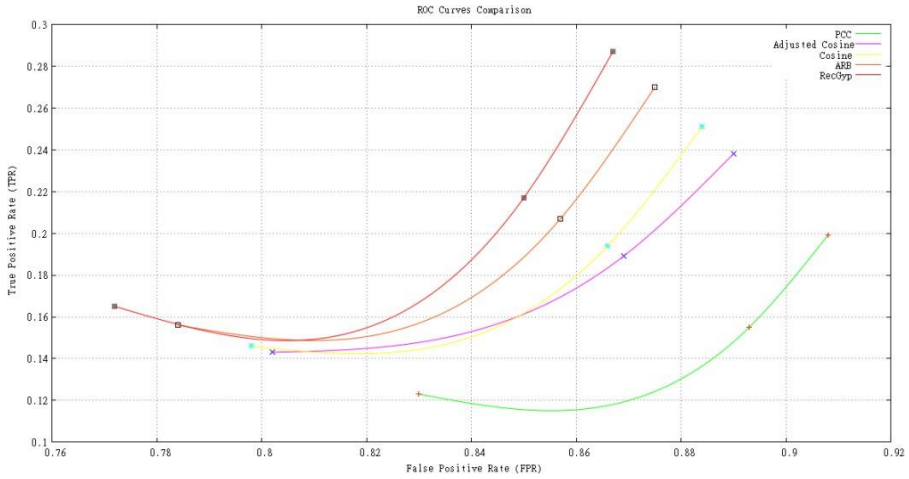

Fig 10. Comparing ROC curves for various RS techniques for the *Yahoo! Weboscope movie* dataset

This paper has provided three primary research contributions:

- The sparsity issue which is a major challenge in building a recommender system has been considered in the novel *RecGyp* recommender system technique.
- Combining the clustering approach with association rule mining has improved the accuracy of results generated by the novel *RecGyp* recommender system technique to a great extent.
- With an aim to achieve representative results, experiments have been carried out on two different types of datasets (*MovieLens 100K* and *Yahoo! Weboscope movie*) which consist of an adequate volume and variety of data in order to produce trustworthy comparative results and overall inferences.

V. CONCLUSIONS

It can be concluded that a RS technique can provide more accuracy of results if we do not solely rely on the explicit ratings provided by users on items. Rather, for a recommendation strategy, it is needed to also explicitly study the features of items and/or users provided in the dataset. We believe that the RS techniques discussed in this chapter is considered as basic, standard state-of-the art techniques for studying the concept of RS in social networks, especially for e-commerce sites and it will help a researcher studying in this field to get a preliminary idea about the same. By adding the concept of clustering in our novel proposed *RecGyp* algorithm, we prove that the problem of generating recommendations can be solved with a low computational cost by considering limited target items. For future work, some comparative study of more recent standard RS algorithms can be done that will result in more accuracy of results and can also handle the *cold-start problem* with low computational cost. Also, hybrid techniques need to be considered that can handle the sparsity issue of RS and build a model that can provide accurate and fast recommendations. Recent studies show that integrating the hierarchy of user or item preferences [17], group recommender systems [19] and interactive recommender systems [6] can also increase the performance of online recommender systems which can also be taken into consideration.

REFERENCES

1. Aggarwal C.C. (2016) An Introduction to Recommender Systems. In: Recommender Systems. Springer, Cham
2. Wang, Suhang, et al. "Exploring Hierarchical Structures for Recommender Systems." IEEE Transactions on Knowledge and Data Engineering 30.6 (2018): 1022-1035.
3. Felfernig, Alexander, et al. Group Recommender Systems: An Introduction. Springer International Publishing, 2018.
4. Aslanian, Ehsan, Mohammadreza Radmanesh, and Mahdi Jalili. "Hybrid recommender systems based on content feature relationship." IEEE Transactions on Industrial Informatics(2016).
5. Wang, Donghui, et al. "A content-based recommender system for computer science publications." Knowledge-Based Systems, Elsevier 157 (2018): 1-9.
6. Faridani, Vahid, Mehrdad Jalali, and Majid Vafaei Jahan. "Collaborative filtering-based recommender systems by effective trust." International Journal of Data Science and Analytics, Springer 3.4 (2017): 297-307.
7. Bobadilla, Jesús, Francisco Serradilla, and Jesus Bernal. "A new collaborative filtering metric that improves the behavior of recommender systems." Knowledge-Based Systems 23.6, pp. 520-528, 2010.
8. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez, "Recommender Systems Survey", in Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.
9. Avazpour, Iman, et al. "Dimensions and metrics for evaluating recommendation systems." Recommendation systems in software engineering. Springer Berlin Heidelberg, pp. 245-273, 2014.
10. Schröder, Gunnar, Maik Thiele, and Wolfgang Lehner. "Setting goals and choosing metrics for recommender system evaluations." CEUR Workshop Proc. vol. 811, 2011.
11. The MovieLens Dataset, Available at http://grouplens.org/datasets/movielens/, accessed November 2016.
12. Yahoo!: Webscope movie data set (Version 1.0), http://research.yahoo.com/, Accessed on 02 January 2017.
13. J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez, "Recommender Systems Survey", in Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.
14. Aggarwal C.C. (2016) An Introduction to Recommender Systems. In: Recommender Systems. Springer, Cham
15. Yang, Dong-Hui, and Xing Gao. "Online retailer recommender systems: a competitive analysis." International Journal of Production Research 55.14 (2017): 4089-4109, Taylor and Francis.
16. Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE Transactions on Knowledge & Data Engineering 6 (2005): 734-749.
17. L. Safoury and A. Salah, "Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System", in Lecture Notes on Software

Engineering, vol. 1, No. 3, pp. 303-307, 2013.
18. P. Lops, M. de Gemmis and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends", Recommender Systems Handbook, Springer, pp. 73-105, 2011.
19. Aggarwal, Charu C. "Content-based recommender systems." Recommender systems. Springer, Cham, 2016. 139-166.
20. X Sun, F Kong, and Hong Chen, "Using Quantitative Association Rules in Collaborative Filtering", Lecture Notes in Computer Science, Springer Berlin/Heidelberg, pp. 822 – 827, 2005.
21. Burke, Robin. "Hybrid recommender systems: Survey and experiments." User modeling and user- adapted interaction 12.4, pp. 331-370, 2002.
22. He, Chen, Denis Parra, and Katrien Verbert. "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities." Expert Systems with Applications, Elsevier 56 (2016): 9-27.
23. Al-Shamri, Mohammad Yahya H. "User profiling approaches for demographic recommender systems." Knowledge-Based Systems, Elsevier 100 (2016): 175-187.
24. Colombo-Mendoza, Luis Omar, et al. "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes." Expert Systems with Applications, Elsevier 42.3 (2015): 1202-1222.
25. Y. Shih, D. R. Liu, "Hybrid recommendation approaches: collaborative filtering via valuable content information", in Proceedings of the 38th Hawaii International Conference on System Sciences, IEEE, pp. 1-7, 2005.
26. da Silva, Edjalma Queiroz, et al. "An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering." Expert Systems with Applications, Elsevier 53 (2016): 204-218.
27. Isinkaye, F. O., Y. O. Folajimi, and B. A. Ojokoh. "Recommendation systems: Principles, methods and evaluation." Egyptian Informatics Journal 16.3, pp. 261-273, 2015.
28. L. Safoury and A. Salah, "Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System", in Lecture Notes on Software Engineering, vol. 1, No. 3, pp. 303-307, 2013.

AUTHORS PROFILE

Dr. Gypsy Nandi is an Assistant Professor in the Department of Computer Science & Engineering and IT, Assam Don Bosco University, India. Her area of interest includes Social Network Mining, Data Analytics, and Machine Learning. She has a number of research publications in reputed Scopus and ESCI journals. She has successfully mentored many winning teams of the National level *Smart India Hackathon*. She also has successfully carried out two sanctioned consultancy-based government projects funded by AICTE and UNDP. She has co-authored a book on *"Soft Computing – Fundamentals and Practical Approaches"* published by Studium Press Pvt. Limited.