OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Challenges in Recommendation System and Handling Sparsity Problem

Meenal Naranje
Prof. Dr. MandarSohani
[1]M.E., Department Of Computer Engineering, VIT, Mumbai, India.
[2]Prof. Of Department Of Computer Engineering, VIT, Mumbai, India..

ABSTRACT: Collaborativefiltering is one of the most successful techniques that attempts to recommend items (such as music, movies, web sites) that are likely of interest to the people. However, Existing CF technique may work poorly due to the sparse attribute inherent to the rating data. In this paper, a new mechanism that combines the user-based rating and item attribute-based is presented. First, we find out the similarities using Euclidean distance, Cosine similarity and Pearson Correlation Coefficient between the critic(user) to the movie(item). Second with the help of formula we calculate the estimated value for sparse data. Case studies show that our approach contributes to estimation of the unrated blank data for sparse matrix. The filling-in accuracy is also acceptable and reasonable.
KEY WARDS:Recommendation System, Collaborative Filtering, Content Based Filtering, data- sparcity, Shilling Attack.

Date of Submission:07-05-2019 Date of acceptance: 24-05-2019

I. INTRODUCTION

As we know the word recommendation means suggestion or proposal to the best course of action,thus in the same way recommendation system works by providing suggestions to the users that are best suited by their preference and demands. Hence it becomes most popular area for researchers and widely used to recommend a product to user that are most appropriate [1][2] and gained popularity. Data Mining is a process of acquiring meaningful data from a huge number of data [1]. Nowadays, Internet has become the lead source for everyone for getting information about anything. The web is the place where billions to millions information sources are available as per user need. As there are huge amount of data are available over web, the problem of information overload appears. Because of Information overload the mining results can take more time to search the required. To overcome information overloading problem, information filtering systems are required. Recommendation is subsystem of information filtering.

Facebook uses a recommender system to suggest recommendation for user about the people they may know. This is happened by Recommendation System where the system is trained based on users personal data, mutual friends, where you went to school, where you worked and mutual networks etc and provide suggestion.

Netflix also makes the use of recommendation System for providing us movie recommendations depending on our criteria/ taste we have already fill out while registering or while rating the particular movie or drama series etc. Thus it filter out through the thousands of Criterias to get a better idea of what you might like to watch.Factors that Netflix algorithm uses to make such recommendations include:

- The genre of movies and TV shows available
- Your streaming history, and previous ratings you've made.

The combined ratings of all Netflix members who have similar tastes in titles to you.

II. TYPES OF RECOMMENDATION SYSTEMS

There are basically three types of Recommendation systems content-based filtering, collaborative filtering and hybrid filtering technique.

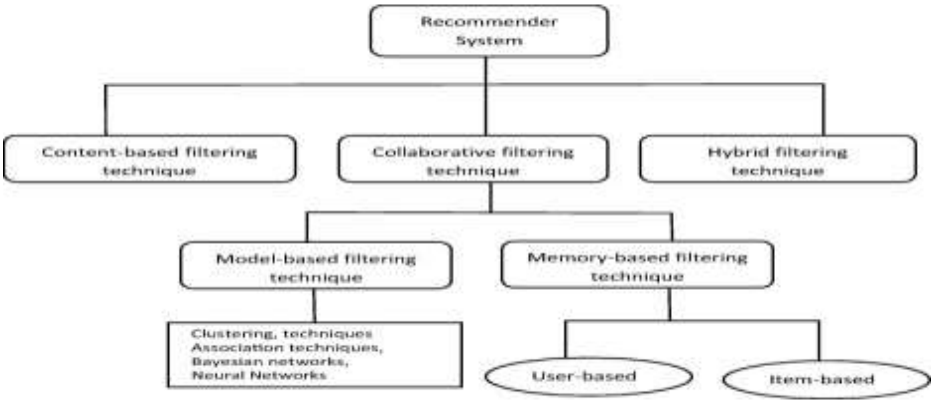

Figure 1: Types of Recommendation System

Among these Collaborative Filtering technique has very high popularity because of their high effectiveness as it creates better suggestions than others.

2.1 Content Based Filtering

Content based recommendation system provides prediction based on user or item information and past interests of user. Content-based filtering method examines users past interests for particular item. Upon examines the user interests, the system provide recommendation for the items that have highly similar kind of features related to user interest or items accessed in past.

Figure 2: Content Based Recommend System

2.2 Collaborative Filtering

This technique analyzes a large amount of data collected from user responses to an item as rating in past and recommends items to user. Here, analyzing item content is not necessary and information is shared between two users so that can provide surprising recommendation which user may pretend to be interested. The base of this method depends on relationship between user and items and also on rating feedback matrix where each element representing a specific rating on a specific items [6].

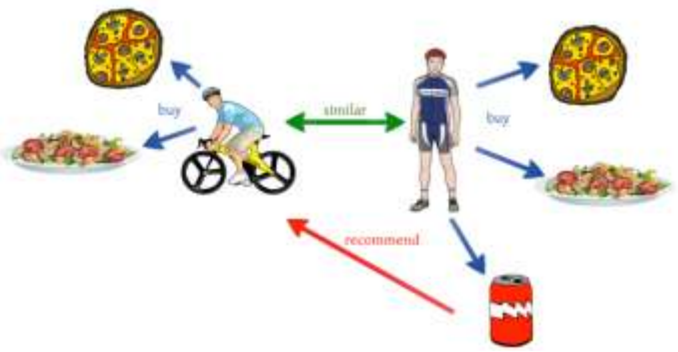

Figure 3: Collaborative Recommendation System

This Collaborative filtering approach is mainly classified into two types they are Model based approach and Memory based approach. The memory-based approaches are one of the most popular methods of applying CF. Memory based collaborative filtering technique approach [7], [8] we will be using the user item rating matrix in order to calculate the ratings that are not rated by the user based up on similar items or users. Hence this finding up of similar users or items can be done in two methods of them the first is Item based collaborative filtering technique and the next one is User based collaborative filtering technique. The first one Item based collaborative filtering approach technique [8], is used for prediction of the unknown ratings for the user for an item based up on the similar items for the item for which we are predicting. The next User based collaborative filtering approach technique is used to calculate the prediction of the unknown ratings for the user for an item based up on the similar users of the user for which we are predicting. The opposite for Memory based is the Model based approach. The main theme of this model based approach is to create a model that uses the ratings in the user item rating matrix directly and then instruct the model using the available information and then used for prediction purpose.

2.3 Hybrid Recommendation Systems

Combining collaborative filtering and content-based filtering could be more effective in some cases. Hybrid approaches can be implemented in several ways, by making content-based and collaborative-based predictions separately and then combining them, by adding content-based capabilities to a collaborative-based approach (and vice versa), or by unifying the approaches into one model. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommendation systems such as cold start and the sparsity problem.

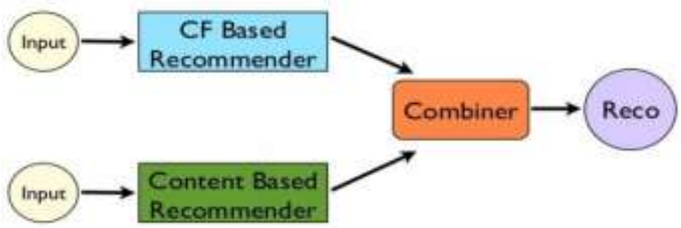

Figure 4: Hybrid Recommendation System

III. LITERATURE REVIEW

While developing Movie Recommendation System, Lakshmi Tharun Ponnam et al. (2016) suggests a system which makes use Item-based Collaborative Filtering Technique for better effect. In this itembased Recommendation process, they generally look at ratings given to similar items on Netflix dataset. In this approach they have used items that are most similar to the current item for which they predict the rating by using the item similarity weights and using the K most similar items and predict the unknown rating. Then recommend the top N items having highest predicted rating as recommendations to the user.

Elena Shakirova (2017) proposed a Music Recommendation System where she makes use of collaborative filtering methods and evaluation metrics to estimate effectiveness of recommender systems. She had prepared a theoretical basis for the implementation of collaborative filtering techniques for a music recommender system and plan to vary different parameters such as similarity measure, scoring function, ranking aggregation and Evaluation metrics to improve the effectiveness of recommender systems. Here also some problems of collaborative filtering such as Cold-Start, Sparcity, Shilling attack hence Parmar Darshana (2018) suggests a Music Recommendation System based on Content based and collaborative filtering for providing solution to the cold start problem [2].

Chaloemphon Sirikayon et al. (2018) suggests another Recommendation System which is used inschool, colleges and university Libraries i.e. Library Book Recommendation System. For generating book recommendations in library Contains 4 steps in user-based collaborative filtering to make a prediction for each student. Initially there is no rating score from student in library for borrowing, Book borrowing records with time stamps are used to construct rating matrix. Besides, matrix factorization technique is also adopted in order to solve sparseness in data and high dimensionality. After that they calculate similarities and based on similarity prediction is done. Books with 3 highest prediction scores are then recommended to the active students. The results show that the accuracy of our recommended books was acceptable and can help library to increase the book utilization.

Chengchao Yu et al. (2018) have provided a model for recommender system for ordering platform in restaurants. They proposed an improved Collaborative Filtering algorithm based on historical order data of restaurants. In general, it is difficult to collect the necessary information about foods and customers in the recommendation system for their ordering system. Therefore, the recommended algorithm based on collaborative filtering is the suitable approach because it makes use of past user activities. They designed and implemented a recommender system for food dishes based on the improved Collaborative Filtering algorithm, which includes modules such as rule generation, incremental learning, recommendation, food's hotness degrading. The recommender system has been successfully served on Zhuoji's ordering platform and has got great feedback. Here they mostly focus on accuracy of recommendations. Thus it not only made full use of historical orders, but also took greater advantage of real-time orders to improve the accuracy of recommendation

IV. CHALLENGES

The web recommender System suffers from many challenges such as Lack of Data, Changing Data, Changing User Preferences, Unpredictable Items, Scalability, Privacy protection. Some of them are: cold-start problem, data-sparsity, shilling Attacks, scalability, Over-Specialization etc.[10]

4.1 Cold-start problem

This problem occurs when new user enters in the system or new item are added to the catalogue. Hence don't have enough previous rating related to that item. In such cases, neither the taste of the new user can be predicted nor can the new item be rated or purchased by user leading to less accurate recommendations. It is a bit difficult to recommend items to new users as the system don't have any information related to his past purchases or it might be possible that he has not rated any item yet so his taste is unknown to the system.

4.2 Data-sparsity

In any recommender system, it is impossible to assume that each user will rate every item present in the system. Consider, you have an online shop that has a huge amount of users and items. If a user purchased few items from the shop and has rated any of them. It is even possible that some users will not rate any item. Then, it will lead to the problem of Sparsity. Also, we can say that Sparsity is the problem of lack of knowledge. This further weakens the recommendations

4.3 Shilling Attacks

It is difficult to spot malicious or unreliable user and their associated ratings. It may happen that malicious user or competitor enters into a system and starts giving false ratings on some items-popularity or to decrease its popularity. It may also happen that people may give positive rating for their item and negative rating for their competitor's item. Thus such attack can break the trust on Recommendation System as well as decrease the performance and quality of Recommender system

4.4 Scalability

It becomes very difficult for typical recommender system to process large scale data and as the numbers of users and items grows the system needs more resources in order to give the most accurate recommendations to the users. Most of resources are used in the purpose of determining users of similar tastes, and items with similar attributes. It is one of the problems found in collaborative filtering approach.

4.5 Latency Problem

Recommendation system faces latency problem when new items are added frequently to the database, where the recommender suggests only the already rated items as newly added items are not yet rated. Collaborative Filtering can reduce waiting time but may introduce overspecialization.

4.6 Over-Specialization

This is one of the most common problems faced by the content-based recommendation system. This occurs when recommender system suggests the item and the preferences which the user has already used in the past leads to over specialization. A good recommender system must suggest diverse items which content-based system lacks. It gives nothing "surprised". It controls the users from discovering something new and different. Users are recommended items they are already familiar with.

V. HANDLING SPARCITY PROBLEM

When there are no rating scores from users is given to all item, sparsity occurs for rating matrix. Matrix factorization is well known technique using for solving sparsity problem in recommendation system. The model maps both users and items to joint factor space of dimensionality, such that user-item interactions are modeled as inner product on that space. The result from matrix factorization is a set of factors for each user and item score in certain dimension.

	Item1	Item2	Item3	Item4
User 1		4.5	2.0	
User 2	4.0		3.5	
User 3		5.0		2.0
User 4		3.5	4.0	1.0
User 5	4.0		3.5	

Rating matrix

=

	Feature1	Feature2
User 1	4.7	0.5
User 2	0	5.3
User 3	5.3	0
User 4	4.2	2
User 5	0	5.3

User matrix

×

	Item1	Item2	Item3	Item4
Feature 1	0	0.9	0.3	0.2
Feature 2	0.7	0	0.7	0

Item matrix

Figure 5:Example of Matrix Factorization Result

Hence we adopted matrix factorization method to explore the latent factors that defers to the relation of users interest and relevant item. And once we calculate the similarity between users, based on their similarity score most similar 'n' user is selected and the rating for each item for active user are predicted by using scores of item. And item with highest prediction scores are then recommended to the active user. thus we could calculate similarity scores for each user. However, only few similarity scores were obtained because the rating matrix was very sparse as shown in Matrix.

To deal with sparsity problem, we filled the missing values with zeros and then apply matrix factorization to decompose it. Then, similarity scores between the user and every other user were calculated as shown in Table I, and Table IIshows an example of similarity scores for student U39 obtained by various approach. The result is slightly different for Pearson correlation and cosine similarity methods.

Table I: Example of similarity scores for User U_{39} without matrix Factorization

Top N neighbors	Similarity Calculation Method		
	Pearson	Cosine	Euclidean
1	325: 0.6324	325: 0.6325	325: 0.1695
2	337: 0.3700	337: 0.3701	301: 0.1351
3	N/A	N/A	8209: 0.1351
4	N/A	N/A	8196: 0.1351
5	N/A	N/A	8195: 0.1351
6	N/A	N/A	8194: 0.1351
7	N/A	N/A	8191: 0.1351
8	N/A	N/A	8188: 0.1351
9	N/A	N/A	8186: 0.1351
10	N/A	N/A	8183: 0.1351

Table II: Example of similarity scores for User U_{39} with matrix Factorization

Top N neighbors	Similarity Calculation Method		
	Pearson	Cosine	Euclidean
1	325: 0.9928	3686: 0.9452	325: 0.2261
2	337: 0.9907	337: 0.9452	301: 0.2133
3	318: 0.9907	351: 0.9452	351: 0.2109
4	351: 0. 9907	2223: 0.9452	282: 0.2092
5	3686: 0. 9907	318: 0.9452	2184: 0.2085
6	2223: 0.9907	301: 0.9451	941: 0.1904
7	301: 0.9898	282: 0.9302	1000: 0.1787
8	282: 0.9599	325: 0.9265	1779: 0.1767
9	5971: 0.8514	2184: 0.9156	7562: 0.1665
10	1930: 0.8360	941: 0.8966	3694: 0.1649

Various similarity calculation methods are available but most Euclidean Similarity, Pearson similarity and Cosine similarity are most commonly used similarity methods.

5.1 Euclidean distance

The Euclidean distance is already familiar to you from 2- and 3-dimensional geometry. The Euclidean distance r2(x, y) between two 2-dimensional vectors x = (x1, x2) T and y = (y1, y2) T is given by:

$$r_2(x,y)=\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = \sqrt{\sum_{i=1}^{2}(x_i - y_i)^2} \quad \text{.................................(1)}$$

Generalising to higher dimensions, the Euclidean distance between two d-dimensional vectors x1 = x12, x12, x13, . . . , x1d T and x2 = x21, x22, x23, . . . , x2d T is given b

$$r_2(x_1,x_2)=\sqrt{\begin{matrix}(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 \\ + \ldots\ldots + (x_{1d} - x_{2d})^2\end{matrix}} = \sqrt{(\sum_{j=1}^{d}(x_{1j} - x_{2j})^2} \quad \text{...(2)}$$

It is often the case that we are not interested in the precise distances, just in a comparison between distances. For example, we may be interested in finding the closest point (nearest neighbour) to a point in a data set. In this case it is not necessary to take the square root

$r_2(x_1,y_2) = |(x_{11} - x_{21})|+|(x_{12} - x_{22})| + \cdots + |(x_{1d} - x_{2d})| = \sum_{j=1}^{d} |x_{1j} - x_{2j}|$ ....(3)

The notation |a| indicates the absolute value of a. More generally it is possible to use other powers, giving rise to a more general form (known as the p-norm or L p –norm).

$r_p(x_1,x_2)=\left(\sum_{j=1}^{d} |x_{1j} - x_{2j}|^p\right)^{\frac{1}{p}}$ ....(4)

Finally we need to convert our distance measure into a similarity measure by applying following formula:

$\mathrm{Sim}(x,y) = \frac{1}{1+r2(x,y)}$ ....(5)

5.2 Cosine Similarity

This method is also most commonly used method in collaborative filtering in recommender systems. Cosine similarity finds how two vectors are related to each other using measuring cosine angle between these vectors. The major drawback with cosine similarity is that it considers null preferences as negative preference.

$s(x,y)= \frac{x\ .\ y}{||x||||y||} = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1}(x_i)^2} \times \sqrt{\sum_{i=0}^{n-1}(y_i)^2}}$ ....(6)

Consider the following Matrix:

Table III: Ratings given to six movies by six film critics

	AntMan	Black Panthor	Aqua Man	Hancock	DeadPool	Forest Gump
Jessica	3	7	4	9	9	7
Ema	7	5	5	3	8	8
Micheal	7	5	5	0	8	4
Sarah	5	6	8	5	9	8
Danial	5	8	8	8	10	9
David	7	7	8	4	7	8
User 2	X	6	9	X	X	6

Consider a new user (user 2) who has not seen Hancock, Australia or Milk, but has supplied ratings to the other three films given in the table, thus creates sparsity in the matrix and this is then removed by calculating the similarity between different users/critic and applies estimation formula shown below.

1. By Euclidean distance

Table IV: Euclidean distance for each critic to the user2

Critic	R2(critic, user2)
Jessica	5.2
Ema	4.6
Micheal	4.6
Sarah	2.2
Danial	3.7
David	2.4

Once we get Euclidean distance then by using formula(5) we have to calculate the similarity which is shown below

Table V: Similarity calculation for each critic to the user2

Critic	sim(critic, user2)
Jessica	0.16
Ema	0.18
Micheal	0.18
Sarah	0.31 (**Highest similarity)**
Danial	0.21
David	0.29

2. By Cosine Similarity

Table VI. Similarity calculation for each critic to the user2

Critic	sim(critic, user2)
Jessica	0.57
Ema	0.64
Micheal	0.59
Sarah	0.73 (**Highest similarity)**
Danial	0.70
David	0.70

5.3 Pearson Correlation Coefficient
5.3.1 Normalisation

So far our estimate of similarity has been based on the Euclidean distance between feature vectors in a review space. But this distance is not well normalised. For example, two critics may rank a set of films in the same order, but if one critic gives consistently higher scores to all movies than the other, then the Euclidean distance will be large and the estimated similarity will be small. In the data we have been working with (Table III) some critics do give higher scores on average: the mean review ratings per critic range from 4.8 to 8.0. One way to normalise the scores given by each critic is to transform each score into a standard score1 . The standard scores are defined such that the set of scores given by each critic have the same sample mean and sample standard deviation. We first compute the sample mean and sample standard deviation for each critic. Consider an M-dimensional feature vector corresponding to critic c, xc = (xc1, xc2, . . . , xcM) , where xcm is critic c's rating for movie m. We can compute the sample mean x¯c and sample standard deviation sc for critic c as follows2 :

$$\overline{x_c} = \frac{1}{M}\sum_{m=1}^{M} x_{cm} \quad \text{...(7)}$$

$$s_c = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(x_{cm} - \overline{x_c})^2} \quad \text{...(8)}$$

We then use these statistics to normalise xcm (the critic c's score for movie m) to a standard score

$$z_{cm} = \frac{x_{cm} - \overline{x_c}}{s_c} \quad \text{..(9)}$$

The z scores for a critic c are normalised with a mean of 0 (obtained by subtracting the mean score from the xc scores) and a sample standard deviation of 1 (obtained by dividing by the sample standard deviation of the xc scores). Thus using these normalised scores for each critic removes the offset effect of differing means and the spread effect of differing variances.

5.3.2 Pearson Correlation Coefficient

To estimate the correlation between two sets of scores we use the Pearson Correlation Coefficient. There are many other ways that we could measure similarity. Rather than considering the distance between feature vectors as a way to estimate similarity, we can consider the correlation between the critics score. Figures 2 plot films in terms of the ratings of two specified critics, along with a best fit straight line.

If the ratings of the two critics are closely related (similar) then the best-fit line will (almost) touch every item, if the films are generally far from the best fit line then the review scores are not well associated(dissimilar).

To estimate this we first normalise the scores for each critic to stand scores, using equations (7), (8) and (9). We can then compute the Pearson correlation coefficient between critics c and d, rcd as:

$$r_{cd} = \frac{1}{M-1}\sum_{m=1}^{M} z_{cm}\, z_{dm} \quad \text{...(10)}$$

$$r_{cd} = \frac{1}{M-1}\sum_{m=1}^{M}\left(\frac{x_{cm} - \overline{x_c}}{s_c}\right)\left(\frac{x_{dm} - \overline{x_d}}{s_d}\right) \quad \text{...(11)}$$

If zcm tends to be large when zdm is large and zcm tends to be small when zdm is small, then the correlation coefficient will tend towards 1. If zcm tends to be large when zdm is small and zcm tends to be small when zdm is large, then the correlation coefficient will tend towards −1. If there is no relation between critics c and d, then their correlation coefficient will tend towards 0.

Once we have calculate the similarity for each critic to the user2 we then find out the estimated value to put in the user item rating matrix to overcome sparsity problem by using following formula.

$$sc_a(m) = \frac{1}{\sum_{c=1}^{C} \mathrm{Sim}\,(\overline{x_a}, \overline{x_c})}\sum_{c=1}^{C} \mathrm{Sim}(\overline{x_a}\,,\ \overline{x_c})\ .\, x_{cm}$$

Estimated value calculation for above matrix using Euclidean distance and cosine similarity is shown below.

Table VI:. Estimated value calculation using Euclidean Distance similarity

		AntMan		Hancock		DeadPool	
	Similarity	Score	Sim * Score	Score	Sim * Score	Score	Sim * Score
Jessica	0.16	3	0.48	9	1.44	9	1.44
Ema	0.18	7	1.26	3	0.54	8	1.44
Micheal	0.18	7	1.26	0	0.00	8	1.44
Sarah	0.31	5	1.55	5	1.55	9	2.79
Danial	0.21	5	1.05	8	1.68	10	2.10
David	0.29	7	2.03	4	1.16	7	2.03
Total	1.33		7.63		6.37		11.24
Estimated Score			5.7	4.7		8.45	

Table VIII: Estimated value calculation using Cosine similarity

		AntMan		Hancock		DeadPool	
	Similarity	Score	Sim * Score	Score	Sim * Score	Score	Sim * Score
Jessica	0.57	3	1.71	9	5.13	9	5.13
Ema	0.64	7	4.48	3	1.92	8	5.12
Micheal	0.59	7	4.13	0	0.00	8	4.72
Sarah	0.73	5	3.65	5	3.65	9	6.57
Danial	0.70	5	3.50	8	5.60	10	7.00
David	0.70	7	5.32	4	3.04	7	5.32
Total	3.99		22.79		19.34		33.86
Estimated Score			5.7	4.8		8.48	

Time complexity for Euclidean Distance, Pearson Correlation coefficient and Cosine Similarity are O(n), O(2n), O(3n) respectively[12].

Advantages And Disadvantages

Matrix Factorization method requires less time to calculate similarity as they fill the missing values with zeros and then apply matrix factorization to decompose it. Whereas when we are calculating similarity using some prediction we requires few steps thus requires more time. Matrix Factorization method is comparatively less accurate as we are simply putting random value (0) and then calculating similarity but when we take extra efforts to fill sparse data with more predicted value we achieve more accuracy.

VI. CONCLUSION

Over the last decades recommender systems emerged as a significant information filtering system and still there are lots of improvements needed. It uses several techniques for recommendation which includes content-based, collaborative and hybrid methods. However in spite of all this advancement, some challenges are still need to be overcome. We reviewed various problems that are faced by techniques followed by recommender system and we proposed a novel mechanism to filling in the unrated rating in sparse matrix with higher accuracy.

REFERENCES

[1]. ChaloemphonSirikayon; PanitaThusaranon; PiyalakPongtawevirat; "Collaborative Filtering based library book recommendation system", 2018 sth International Conference on Business and Industrial Research (ICBIR), 2018.

[2]. ParmarDarshana; "Music recommendation based on content and collaborative approach & reducing cold start problem", 2018 2nd International Conference on Inventive System and Control (ICISC), 2018.

[3]. LjupchoRechkoski; Vangel V. Ajanovski; MarijaMihova; "Evaluation of grade prediction using model-based collaborative filtering methods" ; 2018 IEEE Global Engineering Education Conference (EDUCON), 2018.

[4]. Shakila Shaikh; SheetalRathi; PrachiJanrao; "recommendation system in E-commerce Websites: A Graph Based Approached"; 2017 IEEE 7th International Advance Computing Conference (IACC), 2017.

[5]. Chengchao Yu; Qingshi Tang; Zheng Liu; Bin Dong; Zhihua Wei; " A Recommendar System for Ordering Platform Based on improved Collaborative Filtering Algorithm"; 2018 International Conference on Audio, Language and Image Processing(ICALIP), 2018.

[6]. Lakshmi TharunPonnam; Sreenivasa Deepak Punyasamudram; Siva NagarajuNallagulla; Srikanth Yellamati; "Movie recommender system using item based collaborative filtering technique"; 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), 2016.

[7]. Bansari Patel; Palak Desai; Urvi Panchal; "Methods of Recommender System: A review"; 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017.

[8]. F.O. Isinkaye, Y.O. Folajimi and B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, vol.16,pp.261-273, 2015.

[9]. Elena Shakirova; "Collaborative Filtering for Music Recommender System"; 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus); 2017.

[10]. ohamedBobenia; FaycalM'hamadBouyakoub; AbdelkarBelkhir; "A Collaborative recommendation system for location based social network", 2018 International Symposium on Programming and Systems (ISPS), 2018
[11]. Sarika Jain; Anjali Grover; Praveen Singh Thakur; Sourabh Kumar Choudhari; "Trends, problems and solutions of recommender system",International Conference on Computing, Communication & Automation; 2015
[12]. Ali SeyedShirkhorshidi; Saeed Aghabozorgi; Teh Ying Wah; " Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data" RESEARCH ARTICLE PLOS|ONE December 11, 2015
[13]. Hiroshi Shimodaira "Similarity and recommender systems "

Meenal Naranje" Challenges in Recommendation System and Handling Sparsity Problem" International Journal of Modern Engineering Research (IJMER), vol. 09, no. 3, 2019, pp 07-15