A collaborative filtering-based recommender system alleviating cold start problem

Ishan Rathi, Manoj Sethi

Abstract— Recommender systems are making their presence felt in a number of domains, be it for ecommerce or education, social networking etc. With huge growth in number of consumers and items in recent years, recommender systems face some key challenges. These are: producing high quality recommendations and performing many recommendations per second for millions of consumers and items. New recommender system technologies are needed to scale themselves for new items as well as in new user in the system in order to get high quality recommendations. In this thesis, we focus on collaborative approach-based recommender systems to solve the issue of cold start problem. We have compared multiple algorithms which aim to solve cold start problem and proposed a new hybrid algorithm. This new algorithm is implemented on Movie-Lens 1Million Dataset.

Index Terms— Collaborative, Hybrid algorithm, Recommender systems, Movie-Lens

1. Introduction

With increasing information over the internet there are plenty of options available for the users to choose from. Recommender Systems or Recommendation Systems (RS) filter information to help user identify items suited for their preferences. Items are recommended using past purchases or preference of users.

Content- Based Systems

These systems analyze the features of recommended items. For example, an Amazon prime video user has viewed a number of gangster movies, then "gunslinger" genre movies should be recommended to him from database in the database.

Collaborative-Filtering Systems

Based on similarity metrics between users or items, collaborative filtering systems recommend items. The items which are recommended to a user are the ones that similar users prefer for example a collaborative filtering recommendation system for Amazon prime video tastes could predict which Amazon prime show a user would like given a list user's likes and dislikes.

Given below diagram illustrates the flow of process in this paper.

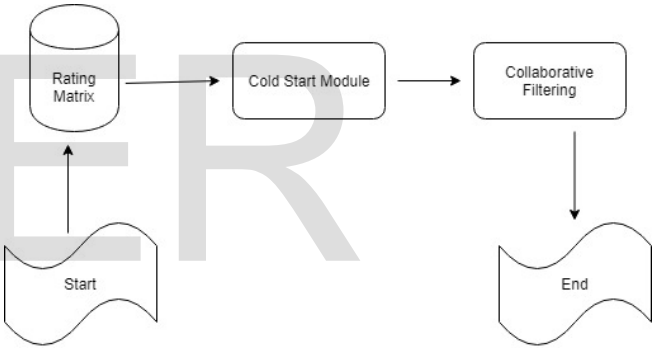

Fig 1. Process Flow

2. Collaborative filtering

Collaborative filtering is one of the most important and popular algorithms that usually predict the rating of the particular user based on similarity between users. Algorithm works on the principal that if some user rated an item with similar rating they might rate other items with similar ratings as well. The similarity between users and/or items is obtained through common similarity measures such as cosine or adjusted cosine, Pearson correlation etc.
One of the benefits of considering like-minded users to make recommendations is that they overcome the "over-specialization" problem. Overly specialized means that the recommended items are always of the same type. Focusing on user ratings rather than content helps avoid such a problem.

The general collaborative filtering framework consists of the following three steps [18]:

a. Data Collection
b. Pre-processing
c. Collaborative

3. Challenges and attack to collaborative filtering

Recommendation system are not perfect and hence are prone to many types of attacks and challenges within themselves. Many of the major issues in CF systems are mentioned below

3.1 Data sparsity problem

CF's principle is to aggregate like-minded user's ratings. However, because of user's absence of knowledge or incentive to rate items the reported user rating matrix is generally very empty.

This issue will prohibit effective recommendations from being made by the CF because the preference of users is difficult to extract.

3.2 Cold start problem

One of the major challenges in RS is to suggest item to user. To recommend item to user the RS needs to look into user's purchase history but in case of a new user no such history is present hence suggestions are inaccurate. In CF based system similarity for new users cannot be established. Some basic solution to such problem involves forcing new user to rate few items before he can start purchasing.

3.3 Changing data set

The database is constantly growing leading to the problem of always changing data set.

3.4 Shilling attacks

In a SA user are inserted into system to provide fake or biased ratings to item, these fake ratings can make even bad products to be highly recommended by system. [7] [9].

3.5 Gray sheep problem

Some users have very different interest when it comes to item ratings and since CF works by comparing similarity of users it becomes difficult to recommend item for a Grey sheep user

3.6 Long tail issue

Recommendation system do not provide users with many options since all recommendations will be based user's purchase history this causes recommendations for only popular products leading to diversity in suggestions problem.

4. Cold start problem

It is hard to recommend new items to user as no information about his previous purchase history is available. There are multiple methods to solve this challenge but all revolve around taking some initial preferences from the new user, without these initial information useful suggestions cannot be made. This problem is commonly known as the cold start problem in recommendation system. When we talk about cold start problem regarding CF system, all suggestions are based on similarity among users/items. One of the Technique to solve cold start problem in CF RS is "ask-to-rate". When a new user registers into the system he is asked rate few items or give a list of his preferences for items. In this way system can gain some initial information to recommend correct items to him. Another solution is, initially a new user is provided with inaccurate suggestion than according to his ratings on those suggestion his new preferences will be determined. In this way accuracy for recommendations will improve gradually.

4.1 Popularity strategy

In this method most popular items from the system are presented to the user. Popularity of an item can be derived as the number of users who have rated the item. The item which is rated by max no. of user can be considered as most popular. It can very well be the case where even the most popular item with the most negative ratings. Popularity of an item can be defined as

$$Popularity(a_t) = |a_t| \quad (1)$$

In this equation $|at|$ refers to the number of users who have rated that item T. This strategy is easy to implement but does not give much useful information as most of user have rated the popular item, so we cannot draw a specific user profile based on most popular item especially in CF systems.

4.2 Random strategy

In this method items are selected in random manner and presented to the user for rating. This approach is not effective as the user may not have any idea about the items he is required to rate. It is one of the basic approaches for cold start problem.

4.3 Pure entropy

Entropy is the measure of randomness in data. In this method we present users with items having mixed ratings so that system is easily able to draw user profile. The items presented are arranged in a descending order of entropy and then presented to the user in non-increasing order. p_i Where p_i is rating for an item. Below is mentioned pseudo code for entropy calculation

$$Entropy(a_t) = \sum_{i=1}^{5} -p_i * \log p_i \quad (2)$$

```
Function Entropy (a_t)
entropy (a_t) = 0
for each item a_t in dataset
        for i as each of the possible rating values // in movielens i = 1 . . . 5
                if a_t's rating= i
                        value[i] += 1 // rating frequencies
        end for
        proportion_i =value[i]   //total number of users who rate at
        entropy (a_t) += proportion_i * Math.log (proportion_i, 2)
end for each
entropy (a_t) = −entropy (a_t)
End
```

4.4 Balanced strategy

In this method we use both popularity and entropy method in combination. Popularity ensures that ratings from users are high and entropy ensures that there is still randomness in items presented for survey

4.5 HELF (Harmonic mean of Entropy and Logarithm of Frequency)

Harmonic mean of entropy and Logarithm frequency. In this strategy harmonic mean of Entropy and Frequency is calculated. Normalization of Entropy and Frequency is also done so that no one factor and dominate the other

HELF value of a_i is calculated as: -

$$HELF_{a_i} = \frac{2 * LF'_{a_i} * H'(a_i)}{LF'_{a_i} + H'(a_i)} \quad (3)$$

Where, LF'_{a_i} is logarithm of the frequency or popularity of a_i and is normalized by the factor $(lg(|U|)) : \mathbf{lg}(|\mathbf{a_i}|) / \mathbf{lg}(|\mathbf{U}|)$, Similarly, $H'(a_i)$ is the entropy of a_i and normalized by a factor of $lg(5) : \mathbf{H}(\mathbf{a_i}) / \mathbf{lg(5)}$

4.6 Item based cold start problem

The new items added to the system are generally excluded while making recommendation and are not presented to new users for initial preference. The reason they are excluded is since these items are new and no user has any preference for it. To solve this problem a set of users can be selected and persuaded to rate these new items.

4.6.1 Market based approach

We consider at a time t there is a set of new items It., We also associate a cost with getting a user's to rate an item. We want to maximize the reach of an item while minimizing the overall cost selecting the users this is known as the market based approach [15]. We consider at a time t there is a set of new items It, also an overall budget is dedicated for all new items. Users are selected on the basis of budget and influence of a user with respect to the item.

Here an earn-per-rating (EPR) list is constructed for each user from which user can choose and provide a rating for a payment. Every new item is placed in a user's EPR list using an item rank. The rank l_{ak}^t for an item i_k in a User (u_a) EPR list. l_a^t Is function of the bid price for an b_k^t and the rate of uptake of the item ru_k^t at a time t .
Rate of uptake is

$$ru_k^t = \left\{ \frac{\sum_{u_a \in U_k^t}^{1} pu(l_{ak}^t)}{\sum_{u_a \in U_k^t} pu(l_{ak}^t)} \right\} \quad (4)$$

This rate of uptake is average appeal of an item while selection by a user. This appeal of an item gives us the new

item which has maximum influence. Thus, new items with high rate of uptake can be used for initial screening of a new user.

4.6.2 IBCTAP algorithm

In IBCTAP algorithm [5] a cluster is formed of items using user-item matrix. The item are partitioned into a group based on similar user liking and we combine this information to build a decision tree to form an association between new and old items. The algorithm IBCTAP has 4 main steps

a) Item clustering

b) Decision tree building

c) New item classifying

d) Ratings predictions

Algorithm: IBCTAP

Input: item-user rating matrix M, number of clusters K, combining coefficient β and item features. Given new item i

Output: $P_{a,i}$: prediction of the new item for the active user

1. Run K-means clustering algorithm on dataset M, break the existing items into K *clusters*.
2. Take the K *clusters* as results and the *item features* as attributes building *decision tree*.
3. Classify i to a certain cluster c according to the *decision tree*.
4. Calculate $P_{a,i}$ using equation (3) by the behaviors of the items in c.

a. Item clustering

K means algorithm is used to form cluster of items which are similar it is often the case when user having a preference of item in one cluster will also have a preference of item in same cluster. Initially k centroid are initialized using K -means algorithm and uses a similarity algorithm to form clusters. Pearson correlation coefficient is used to find similarity between items. Generally, items will have high correlation if user liked both items. Using this assumption all items which are liked a by users with same taste will be same cluster.

b. Decision tree building

Standard Decision Tree building algorithms are used like ID3 and C4.5 which work on the principal of information gain. All items have set of features as described above, for database like Movie Lens 100k each movie item can have features like director, producer, actors etc. A decision tree is built for item based on these attributes the algorithm decides which feature needs to be selected for splitting up the node. The splitting on an attribute is performed only in true and false fashion. Entropy for each item is calculated followed by information gain, if information gain increased the corresponding feature then it is selected for splitting [6]. After each such splitting system proceeds further to see if nodes can be divided or not. Below is a simple example to show a decision tree for such system, where squares show available decisions and circle represents the cluster number.

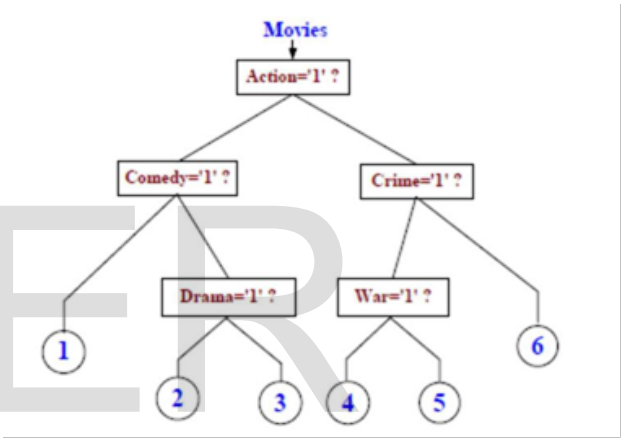

Fig 1: Decision Tree

c. New item classifying

Whenever a new item enters the system which no one has rated yet IBCTAP algorithm classifies it into a cluster using the decision tree and decision algorithm. For example, if an item i arrives in the system it goes to the root of the decision tree. If the item i is a movie of comedy genre it will go to cluster 1. Therefore cluster 1 will contain all movies of comedy genre. In reality this decision tree is not so simple and contain a lot of nodes and clusters as movies can have a lot of item features and separate path for decisions. A worst situation can be when an item i goes to more than one clusters where it becomes difficult to group the new item with already present items.

d. Rating items

Clustering of items with similar features ensures that if a user like one item in that a cluster he will also prefer other

items in the same cluster since they all have similar features. Hence the item is recommended to a user by below equation

$$p_{a,i} = \beta_{r_i}^{-c} + (1-\beta) r^{cf} \; a_i \qquad (5)$$

Where r_i is the mean rating of the active user "a" in that cluster for whom recommendation needs to be made and r^{cf} is the Pearson correlation coefficient among ratings. Where β is the constant for balancing extremely cold start situation. The Pearson correlation coefficient can find similarity between item i and other items for an active user "a".

$$p_{a,i} = \frac{\sum_{j \in I} r_{a,j} \cdot sim(i,j)}{\sum_{j \in I} sim(i,j)} \qquad (6)$$

5. The proposed framework

5.1 Architecture of the proposed system

In the proposed framework a genre-based clustering is used where all users will be clustered based on their preference over the genre. K-means algorithm is used to perform this clustering. Clustering can be done on singular genre or group of genres, here we are forming clusters on a single genre. Dataset used for testing proposed framework is Movie-Lens 1million dataset. It is comprised of movie and movie ratings given by users. In approach used in [6] focuses on alleviating new item cold start problem using K-means on item similarity followed by a decision tree for every item. Algorithm in [6] does not work very well if new user also enters the system. Algorithm fails when items fall into more than one clusters. Proposed Framework works for both new user in system as well as new Item in the system.

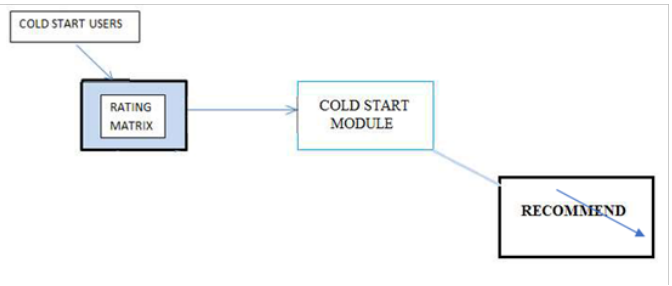

Fig. 2 Architecture of the Proposed System

We start by taking initial user-item matrix and perform K-means upon it based on Item features which is Movie genre for this dataset after that we take new user/item preference into account and merge the clusters to obtain similar users. The list of similar users is used for recommending

New User: If algorithm is applied for a new user the resulting list of user Id corresponds to all user who have similar preference to new user. Thus, these users rated items can be recommended to the new user.

New Item: If algorithm is applied for a new item the resulting list of user id based on item features corresponds to all user to whom this item can be recommended.

5.2 Proposed algorithm

The pseudo code of the proposed system is described below

Input: user-item matrix

Output: user list

Steps:

1.) Find all Different features for an item

2.) Apply K-means clustering on every feature

3.) Take Preference of New Item or New User

4.) Merge Clusters according to New Item/User preference

5.) Obtained User list signify similar users in case of new user.

6.) Obtained User list signify users to whom new item can be recommended

7.) Stop

When we apply K-means on single feature it groups all user who like that feature together and user who have rating less than average in another group. Merging of cluster is done via computing Intersection function.

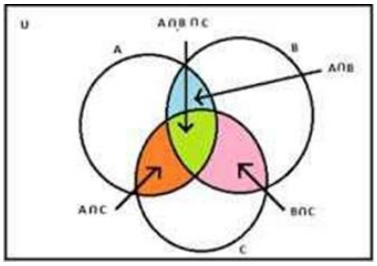

Fig 3: Venn diagram

6. Implementation and results

6.1 Implementation details

1) Python 3.7 is used for implementation of above framework with SCIKIT learn library to implement K-means algorithm.

2) Dataset used is Movie-lens 1million containing 650 user and 1900 movies which contains 1 Million edges in from of user ratings where each is movie rated from rating 1-5. Following is representation of data in dataset.

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Fig. 4 First 5 rows and columns of the dataset

6.2 Results

While clustering over an item feature below graph gives us a visual representation of how users are separated based on their ratings on a feature. For movie genre set to romance following is the resultant from clustering.

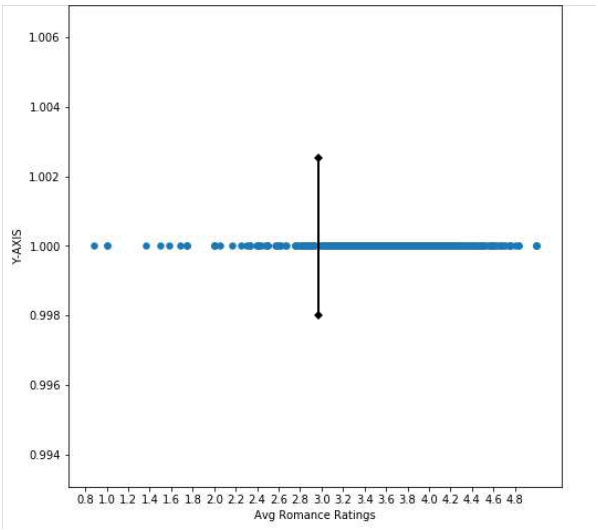

Fig 5 Plot for average romance user ratings

For average user ratings over romance genre it can be seen that after average rating of 3 is a crucial point where user below average rating 3 can be considered as user who do not like this genre and users above 3 are the user who prefer this genre. Similarly, for other genres

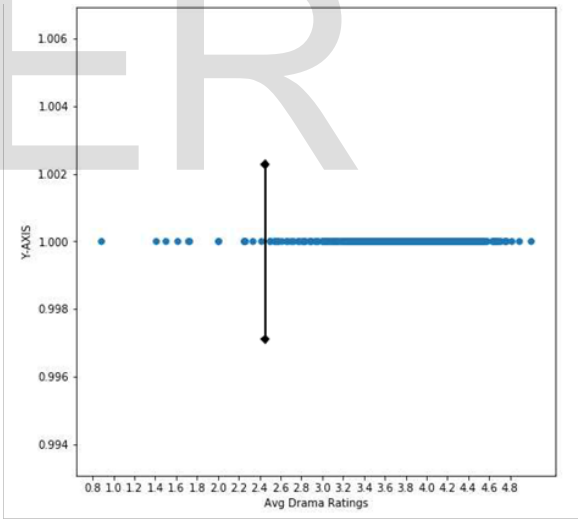

Fig 6. Plot for average Drama user ratings

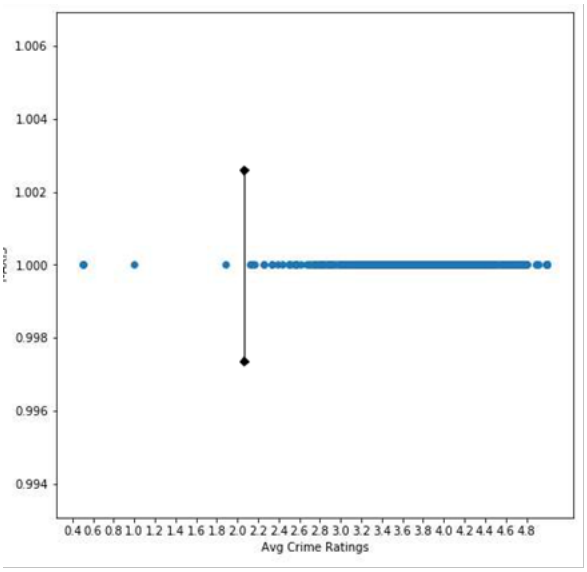

Fig 7. Plot for average crime user ratings

Both Drama and Crime genre similarly show separation points at 2.4 and 2.0 average ratings. As a result, clustering on item features leads to solution for both new user and new item problem.

7. Conclusion

Cold start problem is one of the most common challenges in CF system, most of the traditional approaches relied upon conducting initial survey from new user so that similarity can be drawn. Such approach relies majorly upon the quality of the survey conducted, if item selected for survey are not optimal it will further lead to incorrect recommendation from the system. Several approaches formed in item selection for such survey but led solution to new user problem. Another part of cold start problem is new item in the system. In recent study and research conducted mentioned in Table 1 directs toward the use of item features to resolve cold start problem along with clustering or decision tree. In this thesis, approach of item feature clustering led to solution for cold start users and items in a single algorithm. The merging of multiple clusters into one led to recommendation even over more than one item features. In conclusion K-means algorithm over item feature clustering results in cold start problem solution in collaborative recommendation system

8. References

[1] Du, Y., Du, X. and Huang, L., 2016. Improve the collaborative filtering recommender system performance by trust network construction. Chinese Journal of Electronics, 25(3), pp.418-423.

[2] Leskovec, J., Rajaraman, A. and Ullman, J.D., 2014. Mining of massive datasets. Cambridge university press.

[3] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (Vol. 3, pp. 147-150). IEEE.

[4] Mining of Massive Datasets, Ananad Rajarman, Jeffrey David Ullman, Cambridge University Press New York, NY, USA 2011.

[5] J. Ben Schafer, Joseph Konstan, John Riedl 2005, Recommender Systems in E-Commerce

[6] A Novel Approach for Collaborative Filtering to Alleviate the New Item Cold-Start Problem, Dongting Sun and Zhigang Luo

[7] Soryoung Kim, Sang-Min Choi, Yo-Sub Han 2014, Analyzing Item Features for Cold-Start Problems in Recommendation Systems, 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

[8] Cong Li and Li Ma 2009, Collaborative Filtering Cold-Start Recommendation Based on Dynamic Browsing Tree Model in E-commerce, 2009 International Conference on Web Information Systems and Mining

[9] Reshma R, Ambikesh G and P Santhi Thilagam 2016, Alleviating Data Sparsity and Cold Start in Recommender Systems using Social Behaviour, 2016 fifth international conference on recent trends in information technology

[10] Alhijawi, B. and Kilani, Y., 2016, June. Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. In Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on (pp. 1-6). IEEE.

[11] Adibi, P. and Ladani, B.T., 2013, May. A collaborative filtering recommender system based on user's time pattern activity. In Information and Knowledge Technology (IKT), 2013 5th Conference on (pp. 252-257). IEEE.

[12] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on(Vol. 3, pp. 147-150). IEEE

[13] Dakhel, G.M. and Mahdavi, M., 2011, December. A new collaborative filtering

Algorithm using K-means clustering and neighbors' voting. In Hybrid Intelligent Systems (HIS), 2011 11th International Conference on (pp. 179-184). IEEE

[14] Katarya, R. and Verma, O.P., 2016. A collaborative recommender system enhanced with particle swarm optimization technique. Multimedia Tools and Applications, 75(15), pp.9225- 9239.

[15] Tewari, A.S. and Barman, A.G., 2016, December. Collaborative book recommendation system using trust based social network and association rule mining. In Contemporary Computing and

Informatics (IC3I), 2016 2nd International Conference on (pp. 85-88). IEEE.

[16] Alqadah, F., Reddy, C.K., Hu, J. and Alqadah, H.F., 2015. Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems, 44(2), pp.475-491.

[17] Yang, Z., Wu, B., Zheng, K., Wang, X. and Lei, L., 2016. A survey of collaborative filtering-based recommender systems for mobile Internet applications. IEEE Access, 4, pp.3273- 3287.

[18] Pujahari, A. and Padmanabhan, V., 2015, December. Group Recommender Systems: Combining user-user and item-item Collaborative filtering techniques. In Information Technology (ICIT), 2015 International Conference on (pp. 148-152). IEEE

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

[20] Mehta, B., 2007, July. Unsupervised shilling detection for collaborative filtering. In AAAI (pp. 1402-1407).

[21] Nadimi-Shahraki, M.H. and Bahadorpour, M., 2014. Cold-start problem in collaborative recommender systems: efficient methods based on ask-to-rate technique. Journal of computing and information technology, 22(2), pp.105-113.

[22] Nasiri, M. and Minaei, B., 2016. Increasing prediction accuracy in collaborative filtering with initialized factor matrices. The Journal of Supercomputing, 72(6), pp.2157-2169.

[23] Sharma, R., Gopalani, D. and Meena, Y., 2017, February. Collaborative filtering-based recommender system: Approaches and research challenges. In Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference on (pp. 1-6). IEEE.

[24] Te Braak, P., Abdullah, N. and Xu, Y., 2009, September. Improving the performance of collaborative filtering recommender systems through user profile clustering. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (Vol. 3, pp. 147-150). IEEE.

[25] Tewari, A.S. and Barman, A.G., 2016, December. Collaborative book recommendation system using trust based social network and association rule mining. In Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on (pp. 85-88). IEEE.

[26] Wang, P. and Ye, H., 2009, April. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. In Industrial and Information Systems, 2009. IIS'09. International Conference on (pp. 152-154). IEEE.

[27] Xie, F., Chen, Z., Shang, J., Huang, W. and Li, J., 2015, March. Item similarity learning methods for collaborative filtering recommender systems. In Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on (pp. 896-903). IEEE