



Collaborative Recommendation System Using Dynamic Content based Filtering, Association Rule Mining and Opinion Mining

Anand Shanker Tewari* **Asim Gopal Barman**

National Institute of Technology Patna, India

* Corresponding author's Email: anand@nitp.ac.in

Abstract: This paper proposes a recommendation system (RS) that generates items recommendations to users with the help of dynamic content based filtering, collaborative filtering, association rules and opinion mining. This RS uses dynamic content based filtering for creating and continuously monitoring the changing shopping behaviour of users. The proposed approach finds other like-minded people with the target user that may cooperate with each other, in the form of items ratings using collaborative filtering. The approach uses association rule mining for the analysis of current market trend. It generates association rules only from those items that are liked by the users. Most of the people prefer to read reviews about the product, before purchasing. Almost all well-known e-commerce websites have hundreds of product reviews available, so it becomes very difficult for the user to read each and every review before buying any item. The proposed approach uses its own unique weighted opinion miner that summarizes the reviews and generates the weights for each item based on customers' reviews. These weights help in estimating the popularity of an item among customers. This RS generates the final recommendations to users by combining the outputs from the collaborative-classifier, association rules and weighted opinion miner. The proposed RS is evaluated over live dataset using precision evaluation metric. The result shows that recommendations generated by proposed method out performed existing benchmark recommendations methods.

Keywords: Recommendation system, Collaborative filtering, Content based filtering, Opinion mining, E-commerce.

1. Introduction

With the explosive growth of Internet and other networking technologies, e-commerce is now reaching at every household. Now people minds are also switching from traditional physical shopping experience to online shopping due to scarcity of time. This mind shift has given boost to many e-commerce websites like Amazon.com. In order to attract new customers and retain existing customers, every e-commerce website has its own recommendation system. The RS not only helps in retaining existing customers but also increases the company profit many folds. The main objective of any RS is to recommend items to users according to their interest. The two most widely used recommendation techniques are content based filtering and collaborative filtering. In content based filtering items are recommended to the target user

by matching the contents of items that the user has liked in the past with the content of other unknown items. It creates the profiles of each item with limited set of attributes and matches it with the user profile to make items recommendations [1]. Most common examples of such systems are NewsWeeder and InfoFinder [2, 3]. The major drawback of content based filtering is non-quality assessment and limited scope.

To overcome these shortcomings collaborative filtering has evolved. Collaborative filtering recommends items to the target user based on item ratings of other like-minded people. Few popular example of collaborative filtering are book RS by Amazon.com and the Jester for joke recommendations [4]. In the process of recommendation it is very necessary to realize the market trend and buying habits of the user. To solve this purpose association rule mining is used to

retrieve items that are frequently purchased together by users.

With the advent of Web 2.0, people are expressing their opinion about the products, movies and many other issues in the form of reviews. Despite the availability of ratings, now buyers prefer to read reviews about the products before buying it. Almost all major websites are having large number of reviews. So it becomes very difficult for buyer to read each and every review before making any purchase. Opinion mining is branch of the data mining, it can be used to find semantic polarity *i.e.* *positive or negative* of reviews.

The proposed approach creates the profile of each user, using its unique dynamic content based filter. This filter finds the changes in the interest of the target user dynamically. It also helps in finding the neighbours of the target user having similar mind-set based on contents. The proposed recommender system uses collaborative filtering in the form of classifier, it helps in predicting whether the recommended item will be *liked or disliked* by the target user. It analyses transactions of users to predict current market trend using association rule mining. The proposed system uses its own weighted opinion miner which defines the weightage of every item in recommendation process based on user reviews summarization and item popularity. The complete structure of the paper is organized as follows. Section 2 describes major techniques that are used in the process of recommendation. Section 3 explains the proposed approach. Section 4 elaborates practical implementation and evaluation of the proposed approach and finally Section 5 concludes the paper.

2. Related work

This section describes all the related techniques that have contributed to the development of the proposed approach.

2.1 Content based filtering

Every human being is naturally doing content based filtering in their mind. For example: Whenever people buy newspaper, they in general read only certain sections of the newspaper in which they are interested rather than reading complete newspaper. Earlier information filtering was done by Selective Dissemination of Information (SDI), which was designed to automatically inform scientists about documents publish in their research areas [5]. In content based filtering each user profile is created by information in which user is interested

then it analyses items attributes to identify items that are of interest to the target user [6-8]. In this each item is represented by fixed set of attributes. Presence or absence of an attribute in *item-attribute* matrix is often represented by boolean values *i.e.* 0 or 1. This matrix rarely suffers from sparseness problem. So it results in fairly accurate similarity computation between items based on attribute values [9]. Content based filtering is originated from information retrieval and information filtering [10, 11].

2.1.1. Vector space model

Most content based recommender systems are using Vector Space Model (VSM). In VSM the profile of each user or item can be represented in an n -dimensional space. This model is frequently used in fetching relevant documents from the corpus of documents.

Suppose $D = (doc_1, doc_2, \dots, doc_N)$ is a collection of documents and $T = (t_1, t_2, \dots, t_n)$ be the set of keywords in the collection. Each document is represented by a vector in n -dimensional vector space as $doc_j = (w_{j1}, w_{j2}, \dots, w_{jn})$, where w_{ji} is the weight of word t_i in document doc_j . The weight of any keyword in the document vector shows its relevance. The most common weighting scheme is *Term Frequency–Inverse Document Frequency* (TF-IDF) [10]. The Term frequency (TF_{ij}) of keyword t_i in document doc_j is defined in Eq. (1).

$$TF_{ij} = \frac{f_i}{maxfsum} \quad (1)$$

Where f_i is the frequency of the term t_i and $maxfsum$ is sum of frequencies of all the keywords in the document doc_j . The Inverse Document Frequency (IDF) of keyword t_i is defined in Eq. (2), where N is the total number of documents in the collection and n_t is the count of documents in the collection in which term t_i appears at least once.

$$IDF_i = \log \frac{N}{n_t} \quad (2)$$

The *TF-IDF* weight w_{ji} of keyword t_i in document d_j is defined as shown in Eq. (3).

$$w_{ji} = TF_{ij} \times IDF_i \quad (3)$$

Now the cosine based similarity between the two vector documents doc_1 and doc_2 can be calculated using Eq. (4).

$$\text{Similarity}(\overrightarrow{doc_1}, \overrightarrow{doc_2}) = \text{cosine}(\overrightarrow{doc_1}, \overrightarrow{doc_2})$$

$$\text{cosine}(\overrightarrow{doc_1}, \overrightarrow{doc_2}) = \frac{\overrightarrow{doc_1} \cdot \overrightarrow{doc_2}}{|\overrightarrow{doc_1}| \times |\overrightarrow{doc_2}|} \quad (4)$$

2.2 Collaborative filtering

Content based recommendation cannot filter items based on quality, it cannot differentiate between good item and bad item if both the items are having same attributes. Collaborative filtering is one of the most widely used recommendation technique. It is an automation of the “Word of Mouth” recommendations [12]. It filters items based on quality. Goldberg et al. had introduced the term collaborative filtering by building a system of email filtering called as Tapestry [13]. User based collaborative filtering systems are based on human analysis. It works by collecting the ratings of the target user in a given domain and matching it with other people ratings in the same domain. Then the ratings of the similar people are used to generate the recommendations for the target user [14, 15]. Similarity between the two users ($Sim_{m,n}$) can be calculated using Pearson’s correlation formula as shown in Eq. (5).

$$Sim_{m,n} = \frac{\sum_{a \in I_{mn}} (R_{m,a} - A_m)(R_{n,a} - A_n)}{\sqrt{\sum_{a \in I_{mn}} (R_{m,a} - A_m)^2 \sum_{a \in I_{mn}} (R_{n,a} - A_n)^2}} \quad (5)$$

Where $R_{m,a}$ is the rating of item a by user m and $R_{n,a}$ is rating of same item by another user n . A_n and A_m are average ratings of users m and n respectively. I_{mn} is the set of items rated by both users m and n . These similar users or neighbours, whose similarity are above certain threshold takes part in recommending items to the target user. The rating (P_{xk}) of an item k for target user x is calculated using Eq. (6).

$$P_{xk} = A_x + \frac{\sum_{m=1}^c (R_{m,k} - A_m) \times Sim(x,m)}{\sum_{m=1}^c Sim(x,m)} \quad (6)$$

Where $Sim(x,m)$ is the similarity between target user x and the neighbouring user m and c is the total number of the neighbours of the target user x . A_x and A_m are average ratings of user x and m . $R_{m,k}$ is the rating of item k by user m .

2.3 Association rule mining

Association Rule Mining (ARM) captures the relationship among items on the basis of patterns of co-occurrence across transactions. Market basket

analysis is an example of ARM, it analyses shopping cart to find the relationship among items customer put in their shopping cart [16, 17]. ARM creates a win-win situation not only for the buyers by increasing the convenience but also for the sellers by increasing the profit [18]. Let $I = \{i_1, i_2, i_3, \dots, i_m\}$ is a frequent itemset. An association is an expression of the form $A \rightarrow B$, where $A \subseteq I$, $B \subseteq I$ and $A \cap B = \emptyset$ [19]. In the expression $A \rightarrow B$, A can be called as the Left Hand Side (L.H.S.) and B as the Right Hand Side (R.H.S.) of the association rule. The association rules evaluation metrics are as follows.

Support: It shows number of transactions in the database that contains both A and B.

$$\text{Support}(A \cup B) = P(A \cup B) \quad (7)$$

Confidence: It tells how frequently item B occurs in a transaction, if A has already bought.

$$\text{Confidence}(A \rightarrow B) = P(B|A) \quad (8)$$

2.4 Opinion mining

Items recommendations have relied for a long time on user’s ratings and item descriptions. With the arrival of Web 2.0 now people are actively writing their experiences about various products, services and other things in the form of reviews. Reviews are now other sources of data for generating recommendations [20]. Many e-commerce websites are having large number reviews about various products written by their customers. These reviews help new customers to gain insight about the products and the manufacturers to know the performance of their products in the market [21]. Reviews written by one user can help another user to form an opinion about a product [22]. For a customer it is very difficult to read each and every review about the product before making purchase. Opinion mining is alleviating this pain of customers by analyzing each and every reviews and finding their polarity or semantic orientation i.e. positive, negative or neutral using various machine learning techniques [23, 24]. Opinion mining helps users by extracting information from the opinions or reviews of other people, about some particular item or topic. In

general adjective words are used to describe nouns and pronouns. So for finding a review polarity adjective words are extracted from the reviews using natural language processing method. Next, semantic orientation of these words are found using WordNet. The polarity of these adjective words helps in finding the review polarity of complete review [25, 26]. All the adjective words along with their semantic polarity are collected in *bag of seed words* using WordNet. The pseudo code for calculating review polarity is shown in Fig. 1. The procedure *Review Polarity* calculates the polarity of the complete review, whereas the procedure *WordSense* finds the semantic orientation of individual sentences in the review.

```

1. Procedure Review Polarity( )
2. begin
3.   for each review sentence sent do
4.     polarity = 0;
5.     for each review word rw in sent do
6.       polarity + = WordSense(rw; sent);
7.       /* Positive =1 , Negative = -1*/
8.     end for
9.   if polarity > 0 then
10.    SET sent polarity = Positive ;
11.  else
12.    if polarity < 0 then
13.      SET sent polarity = Negative ;
14.    end if
15.  end for
17. end
    
```

```

1. Procedure WordSense (word, sentence)
2. begin
3.   polarity = polarity of word in bag of
      seed words ;
4.   if there is NEGATIVE WORD appears
      closely around ADJECTIVE word in
      sentence then
5.     polarity = opposite(polarity);
6.   end if
7. end
    
```

Figure.1 Pseudo code for calculating review polarity

3. Proposed approach

The proposed approach uses specialized dynamic content based profiler which works in association with social collaborator. Proposed recommender system analyses the current market trend and opinions of the customers using

association rule and opinion mining. It has six key building blocks: Dynamic Profiler, Similarity Calculator, Collaborative Classifier, Weighted Opinion miner, Association Rule Generator and Final Recommender block. All the blocks works in co-operation with each other and generates the final recommendation for the target user. The working of each block is as follows:

3.1 Dynamic Profiler (DP)

The proposed approach uses specialized dynamic profiler which creates the profile of each user taking into consideration the changing moods of users. It has been observed that user interest is not always constant. It varies from time to time. For example: At certain stage of life user may be interested in competitive books afterward his interest may be switched towards romantic novels, may be later on meditation books and so on. DP analyses user’s browsing history and shopping pattern to find users present and past area of interest.

Table 1. User-Keywords table

User_Id	Keyword	Count	Timestamp

Dynamic profiler creates the profile of each user by Dynamic Keywords Vector (DKV). Each user is represented by their DKV. These keywords stores information regarding user’s past buying history and current browsing information. Keywords of DKV are stored in the *User-Keywords* table, having four columns *User_Id*, *Keywords*, *Count* and *Timestamp* as shown in Table 1. Generally items are arranged in the hierarchical order in e-commerce website as shown in Fig. 2, where root represent *all* items and *leafs* are individual items in the website [27].

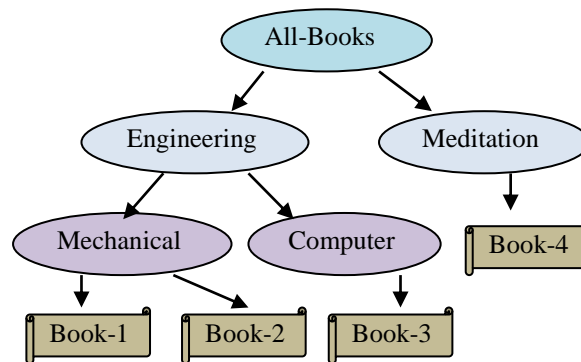


Figure.2 Hierarchical arrangement of items (books) in e-commerce website

Whenever any user buys or browses any new item, DP stores the all the keywords related to that item in User-Keywords (U-Kw) table, excluding the root and the leaf nodes. Example: Whenever any user buys *Book-3*, as shown in Fig. 2, the DP includes keywords *Engineering* and *Computer* in the DKV of that user in U-Kw table. Every time whenever new keyword comes in U-Kw table, *Count* field of the table is incremented by one and current date is set into the *Timestamp* field of the U-Kw table. When the same keywords comes again for insertion into the DKV of the user, then DP simply increases the *Count* value of keywords by one and updates the *Timestamp* field of the U-Kw table. A DKV of a user can have maximum of n keywords in it. Whenever a new keyword comes and a DKV already have n keywords in it, then the keyword having oldest timestamp value is deleted and new keyword is inserted with current timestamp. This approach helps in finding user’s current interest and also preserving his past likings. The pseudo code for storing keywords in user’s DKV is shown in Fig. 3.

3.2 Similarity Calculator (SC)

Once the DKV of every user has found by the DP block, now SC block finds the weights (w_{ji}) of every keyword in user’s DKV using Eq. (3). Afterwards SC finds the similarity between the two users U_1 and U_2 using Eq. (4), this equation can be rewritten as shown in Eq. (9), where U_1 and U_2 are user’s DKVs.

$$\text{cosine}(\vec{U}_1, \vec{U}_2) = \frac{\vec{u}_1 \cdot \vec{u}_2}{|\vec{u}_1| \times |\vec{u}_2|} \quad (9)$$

3.3 Collaborative Classifier (CC)

This block generates recommendations to the target user based on collaborative filtering. CC takes the input from the SC block and finds k most similar users to the target user. This block does not predict the ratings of the items that may be liked by the target user, instead it gives labels (*like* or *dislike*) to the items [28]. It does so using the following steps:

- It divides the existing ratings of every user into four quartiles.
- Then it generates the collaborative ratings of items that may be liked by the target user using Eq. (6).
- Afterward it gives label *like*, to only those items whose predicted ratings are greater than or equal to the target user’s ratings in fourth quartile.

```

1. SearchPath (Item Name/ Category/ Subcategory)
{
    /* This function will find all the intermediate
       nodes in between root node and the argument
       of function. */
}

2. Count( Book Name/ Category/ Subcategory )
{
    /* This function will count the number of
       intermediate nodes in between the root
       node and the argument of function. */
}

3. StoreKeywords (ItemName / Category /
   Subcategory, Date )
{
    a. SearchPath(Item Name/ Category/
       Subcategory)

    if Intermediate nodes found in step a. already
       exist in the DKV of user profile

    then

        Update Timestamp field of the U-Kw
           Table with the current date.
        Update Count field in the U-Kw Table.
        SET Count = Count + 1;

    else

        if Number of keywords in DKV of user >=  $n$ 

        then
            /* Remove older keywords from the U-Kw
               Table based on the Timestamp field of the
               U-Kw Table */
            Number of keywords removed from the
            U-Kw Table = COUNT (Book Name/
                Category/ Subcategory)

        end if

        /* Insert intermediate nodes along with the
           Date of Purchase in the U-Kw Table. */

        SET Keyword = IntermediateNodes;
        SET Timestamp = DateofPurchase;
        Count = 1;

    end if
}

```

Figure.3 Pseudo code for storing keywords in DKV

In the end CC block produces list of items that have label *like*, as recommendations to the target user.

3.4 Association Rule Generator (ARG)

ARG block generates the association rule based on the buying history of the user. It considers each user profile as a transaction to generate association rules [29]. A transaction may have both good and bad items. It is not a good idea to generate association rules from bad items. So ARG block divides the ratings of each user into quartiles and considers only those items for the generation of association rules that are in fourth rating quartile of the user. It means it generates association rules for only those items that are liked by the user. Next it collects all the association rules generated by all the users. Then ARG matches the L.H.S. of the rules with the shopping history of the target user and if match is found then it generates the R.H.S. of the corresponding rule as recommendation to the target user.

3.5 Weighted Opinion Miner (WOM)

This block deals with the opinion of the users about an item. It mines the reviews of every item and finds out the review polarity of every review. This block also keeps track about the number of persons who have given their reviews about an item. Afterward WOM converts every review into its corresponding rating form. WOM considers every positive review equivalent to the highest rating and every negative review equivalent to the lowest rating in the given rating scale [30]. WOM calculates the *opinion weight (OW)* of every item by using the *review ratings*, derived from the reviews of the item.

For every item, WOM calculates the *average review rating (ARR)* of that item and counts (*C*) the number of users who have given their reviews about that item. The weight of an item is calculated using Eq. (10).

$$OW_i = \frac{ARR}{R_{max}} \times \frac{C_i}{C_{max}} \quad (10)$$

In this equation R_{max} , is the highest rating of the given rating scale and C_{max} is the largest value of the count given to any item, where as C_i is the count of item i . Example: Table 2 shows the ARR of different items. For the data given in Table 2, *opinion weight* of an item I_7 is calculated by putting

the values in Eq. (10). The value of ARR is 4, C_7 is 16, R_{max} is 5 and C_{max} is 20.

Table 2. Average Review-Rating conversion table

Item Id (i)	Average Review Rating (ARR)	Total number of reviews of that item (C)
I_1	3	20
I_3	5	17
I_4	2	10
I_7	4	16

Table 3. Recommendation table

User Id	Item Id (i)	Recommendation Value
User 5	I_1	2.8
User 5	I_2	1.5
User 5	I_3	1.6
User 5	I_7	2.7

3.6 Final Recommender (FR)

This block finally generates the recommendation for the target user. It takes the input from the Collaborative Classifier, Association Rule Generator and Weighted Opinion Miner and produces list of recommended items after certain processing. It stores the recommendations in the *Recommendation table* having two fields *Item_Id* and *Recommendation Value (RV)* as shown in Table 3. *Item_Id* uniquely identifies an item and RV of any item is calculated using Eq. (11). It is summation of outputs of the CC, ARG and WOM blocks. Items are recommended to the target user in the decreasing order of their RV values.

$$RV_i = CC + ARG + OW_i \quad (11)$$

Suppose CC block generates the recommendation of three items i.e. I_1 , I_3 and I_7 for user 5. So FR block inserts all these items into Recommendation table and set their recommendation value to be 1. Similarly ARG block has recommended I_1 , I_2 and I_7 to user 5. Now FR block inserts only I_2 in the Recommendation table with recommendation value 1 and increment the recommendation values of already present items I_1 and I_7 by 1. Afterward FR block adds the Opinion Weight of each item to their recommendation values. Suppose the Opinion weights of I_1 , I_2 , I_3 and I_7 are 0.8, 0.5, 0.6 and 0.7. At last FR block recommends items to user 5 in decreasing order of their recommendation values i.e. I_1 , I_7 , I_3 and I_2 . All the recommendations are generated when user is offline, and when user comes

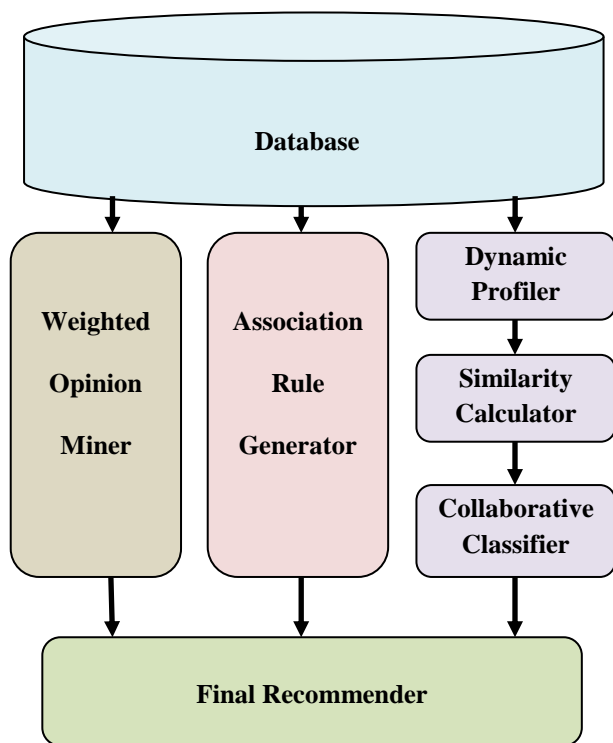


Figure.4 Block diagram of proposed recommendation system

online next time items are recommended based on their RV values. Block diagram of the proposed approach is shown in Fig. 4.

4. Practical implementation and evaluation

The proposed approach has been implemented using Java 7 with NetBeans 8.0 framework. The complete approach runs on Intel Core i5 machine with 4 GB RAM. Reviews, ratings and other information is stored in the MySQL database. The proposed approach has used many established libraries of Java to generate separate recommendations.

4.1 Association rule generation using SPMF

Sequential Pattern Mining Framework (SPMF) is a Java open source library that provides implementation of more than fifty five data mining algorithm and has specialization in frequent pattern mining. SPMF uses Java 7 or higher for execution. The proposed approach has used SPMF for generating association rules.

4.2 Opinion Mining using Stanford Part Of Speech (POS) tagger

Generally features are nouns or noun phrases in review sentences and opinion words are adjectives. Proposed approach used Stanford POS tagger to extract adjectives from the reviews.

4.3 Evaluation

The performance of the system is evaluated over live data of the website www.myopinions.in, having 500 genuine users, 900 reviews of movies and books, 3500 ratings of 1325 books and movies. It is compared with existing benchmark methods of recommendations like content based filtering, collaborative filtering or association rule based recommendation systems. For the evaluation of the proposed approach, movies in the website are also considered as items. The sole purpose of any e-commerce website is to sell items. Keeping this view of seller in mind, the proposed work focuses on predicting whether the items recommended by this approach, are bought by the target user or not. To achieve this purpose the proposed approach is tested using precision evaluation metric. Precision in this context is the ratio of number of relevant recommendation to the total number of recommendations.

The experimental evaluation of the system is initiated by the CC block, it divides the ratings of each user into four quartiles and then generates the list of recommended items that have ratings greater than equal to the target user fourth quartile ratings. The precision graph of collaborative filtering performed by only CC block is shown in Fig. 5.

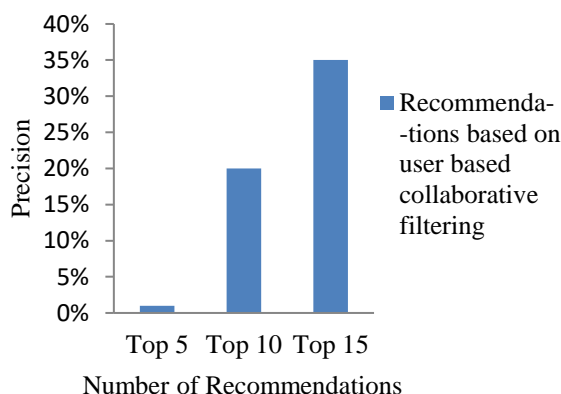


Figure.5 Precision graph of user based collaborative recommendations

In the next step, the profile of each user is constructed in the form dynamic keywords vector, using DP block. Then SC block calculates the similarity among users based on DKVs. Next we selected the 10 neighbours of target user based

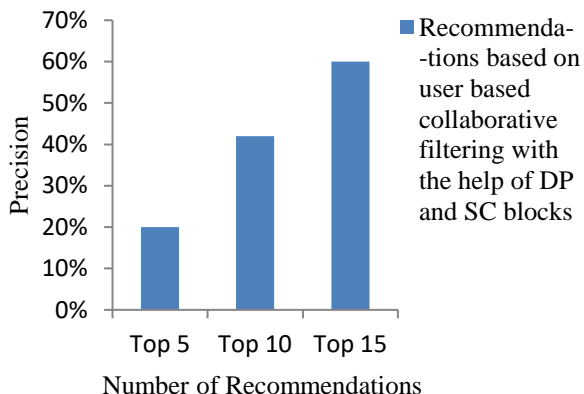


Figure.6 Precision graph of user based collaborative recommendations with the help of DP and SC blocks

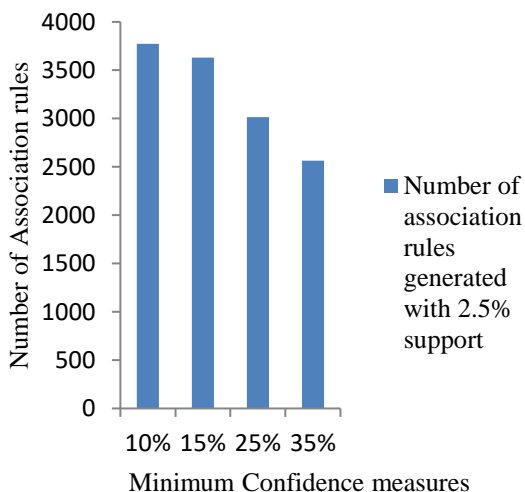


Figure.7 Association rules generated with respect to different confidence measures

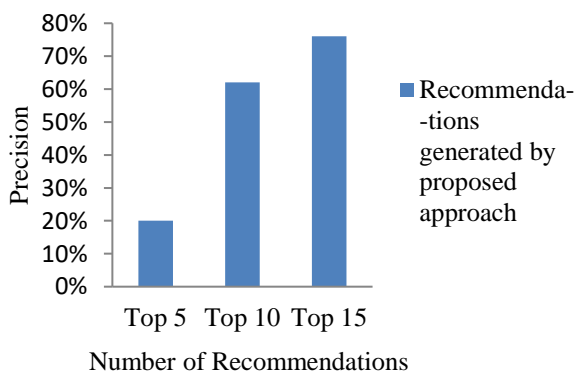


Figure.8 Precision graph of recommendations generated by proposed approach

on the descending order of the similarity values found by the SC block and generated the recommendations using CC block. The same

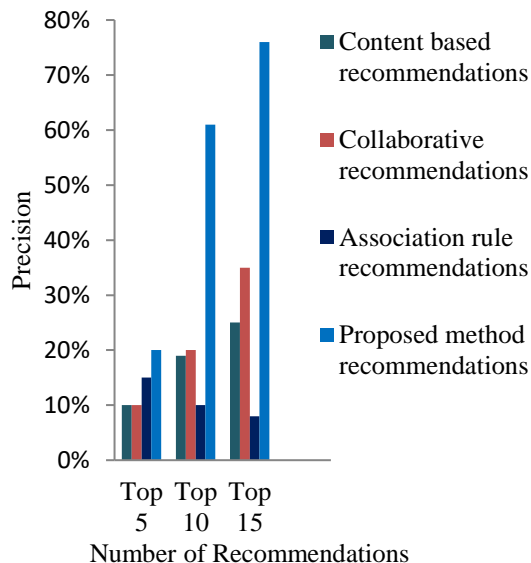


Figure.9 Precision comparison graph of proposed method with other recommendations approaches

process of recommendation is repeated with 20, 30, 40 and 50 neighbours. The recommendations generated using 30 neighbours are more accurate. The precision graph of collaborative recommendations with the help of DP and SC blocks has shown significant improvement as visible in Fig. 6.

Afterward we have generated the association rules using ARG block. It considers all the items bought by the user as a single transaction and association rules are generated using SPMF. Many combinations of support and confidence measures are used for rule generation as shown in Fig. 7. The rules generated by 2.5 % support and 25% confidence are sufficient in numbers and more relevant. Once the rules are generated, ARG block compares L.H.S. of the rule with user transaction and generates the rule R.H.S. as recommendations to the target user.

In the next step, the role of WOM comes it converts every review about items into its rating form using Stanford POS tagger and pseudo code shown in Fig. 1. Now it calculates the opinion weight of every item using Eq. (10).

At last final recommender block outputs the recommendations for the target user after consolidating the recommendations from the CC, ARG blocks and the weights of items generated by the WOM block. The precision graph of final recommendations to the target user is shown in Fig. 8. The comparison of the proposed approach with other popular benchmark recommendation methods is shown in Fig. 9.

5. Conclusion

The proposed RS uses content based filtering, collaborative filtering, association rules and opinion mining collectively. Its unique dynamic profiler block creates the profile of each user using content based filtering. Dynamic profiler with the help of similarity calculator block provides neighbours of target user as input to the collaborative classifier block. The proposed work is evaluated using precision evaluation metric. It has been found in experiments that collaborative filtering independently gives 35% precision for top 15 recommendations. The experiment shows that when DP and SC blocks works in cooperation with CC block the recommendations precision value rises up to 60%. To the best of our knowledge, all researches on opinion based RS has considered only review polarity as important factor for items recommendations but none have considered the number of users who have expressed their opinion about an item for recommendations generation. The WOM block of this RS generates opinion weight for each item based on opinion mining and the number of users who have expresses their opinion about an item. The ARG block generates association rules based recommendations using items that are liked by the user. The final recommender block of this RS creates the list of items to be finally recommended to the target user by combining outputs of ARG, CC and WOM blocks. The experiment result clearly describes that precision values of top 15 recommendations of this approach is around 76%, which is better than traditional benchmark recommendation methods. One more major advantage of this research is that it uses readily available users and items data for recommendations generation, this makes it suitable for almost all e-commerce websites. In future it can be extended by considering the trust factor of the target user on other users while generating recommendations.

References

- [1] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining Content-Based and Collaborative Filters in an Online Newspaper", In: *Proc. of ACM Workshop Recommender Systems: Algorithms and Evaluation*, 1999.
- [2] K. Lang, "Newsweeder: Learning to Filter Netnews", In: *Proc. of the Twelfth International Conf. on Machine Learning*, pp.331-339, 1995.
- [3] B. Krulwich, and C. Burkey, "Learning user information interests through extraction of semantically significant phrases", In: *Proc. of the AAAI Spring Symposium on Machine Learning in Information Access*, pp.110-112, 1996.
- [4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm", *Information Retrieval Journal*, Vol.4, No.2, pp.133-151, 2001.
- [5] E. M. Houseman, and D. E. Kaskela, "State of the art of Selective Dissemination of Information", *IEEE Transactions on Engineering Writing and Speech*, Vol.13, No.2, pp.78-83, 1970.
- [6] P. W. Foltz, and S. T. Dumais "Personalized information delivery: an analysis of information filtering methods", In: *Communications of the ACM*, Vol.35, No.12, pp.51-60, 1992.
- [7] M. Pazzani, "A Framework for Collaborative Content-Based and Demographic Filtering", *Artificial Intelligence Review*, pp.393-408, 1999.
- [8] M. Pazzani, and D. Billsus, "Content-Based Recommendation Systems", In: *The Adaptive Web, LNCS*, Vol.4321, pp.325-341, 2007.
- [9] J. Fan, W. Pan, and L. Jiang, "An improved collaborative filtering algorithm combining content- based algorithm and user activity", In: *Proc. of International Conf. on Big Data and Smart Computing*, pp.88-91, 2014.
- [10] G. Salton, *Automatic Text Processing*, Addison-Wesley, 1989.
- [11] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.
- [12] U. Shardanand, and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'", In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 210-217, 1995.
- [13] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry", In: *Communications of the ACM*, Vol.35, No.12, pp.61-70, 1992.
- [14] J. L. Herlocker, J. A. Konstan , A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering", In: *Proc. Of 22nd International ACM SIGIR Conf. on Research and development in information retrieval*, pp.230-237, 1999.
- [15] J.L. Herlocker, J.A. Konstan, and J. Riedl, "Explaining Collaborative Filtering Recommendations", In: *Proc. of ACM Conf. Computer Supported Cooperative Work*, pp.241-250, 2000.

- [16] J. Han, M. Kamber, and J. Pei, “*Data Mining: Concepts and Techniques*”, The Morgan Kaufmann Series, 2001.
- [17] B. Mobasher, H. Dai, T.Luo, and M. Nakagawa, “Effective personalization based on association rule discovery from web usage data”, In: *Proc. of 3rd ACM International workshop on Web Information and Data Management*, pp. 9-15, 2001.
- [18] W. Yonggang, “Sequential Association Rules Based on Apriori Algorithm Applied in Personal Recommendation”, *International Journal of Database Theory and Application*, Vol.9, No.6, pp.257-264, 2016.
- [19] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, In: *Proc. of ACM SIGMOD International Conf. on Management of Data*, Vol.22, No.2, pp. 207-216, 1993.
- [20] R. Dong, Michael P. O’Mahony, M. Schaal, K. McCarthy, and B. Smyth, “Combining similarity and sentiment in opinion mining for product recommendation”, *Journal of Intelligent Information Systems*, Vol.46, No.2, pp. 285-312, 2016.
- [21] P.V. Rajeev, and V.S. Rekha, “Recommending Products to Customers using Opinion Mining of Online Product Reviews and Features”, In: *Proc. of IEEE International Conf. on Circuit, Power and Computing Technologies*, 2015.
- [22] S. Homoceanu, M. Loster, C. Lofi, and W. Balke, “Will I Like It? Providing Product Overview Based on Opinion Excerpts”, In: *Proc. of 13th IEEE Conf. on Commerce and Enterprise Computing*, pp.26-33, 2011.
- [23] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification Using Machine Learning Techniques”, In: *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, Vol.10, pp.79–86, 2002.
- [24] F. Salvetti, S. Lewis, and C. Reichenbach, “Automatic opinion polarity classification of movie reviews”, *Colorado Research in Linguistics*, Vol.17, No.1, pp.1-15, 2004.
- [25] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to WordNet: An On-line Lexical Database”, *International Journal of Lexicography*, Vol.3, No.4, pp. 235-244, 1990.
- [26] M. Hu, and B. Liu, “Mining and summarizing customer reviews”, In: *Proc. of the 10th International Conf. on Knowledge Discovery and Data Mining*, pp.168-177, 2004.
- [27] Q. Shambour, M. Hourani, and S. Fraihat, “An Item-based Multi-Criteria Collaborative Filtering Algorithm for Personalized Recommender Systems”, *International Journal of Advanced Computer Science and Applications*, Vol.7, No.8, pp.274-279, 2016.
- [28] C. Basu, H. Hirsh, and W. Cohen, “Recommendation as Classification: Using Social and Content-Based Information in Recommendation”, In: *Proc. Of the 15th National Conf. on Artificial Intelligence*, pp.714-720, 1998.
- [29] J. J. Sandvig, B. Mobasher, and R. Burke, “Robustness of collaborative recommendation based on association rule mining”, In: *Proc. of ACM Conf. on Recommender systems*, pp.105-112, 2007.
- [30] Wei Li, and Bo Sun, “An Improved Collaborative Filtering Recommendation Algorithm Incorporating Opinions Analysis”, In: *Proc. of IEEE International Conf. on Intelligent Human-Machine Systems and Cybernetics*, pp.171-173, 2015.