Context based Recommendation Methods: A Brief Review

Arati R. Deshpande
PhD Research Scholar, Computer Engineering
PAHER, University
Udaipur, India

Emmanuel M., PhD
Professor, Dept. of IT
Pune Institute of Computer Technology
Pune, India

ABSTRACT
Recommendation systems consist of methods for recommending products or any items that are of interest to users in web applications for personalized experience. The recommendation helps the users to reduce the time and complexity of searching for the required information. The recommendation methods use the information of users and items as well as users' past history of interaction to suggest preferred items. The context based methods use the situation about the user, item or interaction to give recommendations to users. Currently with the growth of techniques in acquiring the information of interaction of users with the system, the context based methods for recommendation improve the quality of recommendation. A brief review of the approaches and methods for context based recommendation is presented here with the challenges and future directions.

General Terms
Data Mining, Information Retrieval.

Keywords
Recommendation systems, Context aware, Pre and Post filtering, Contextual Model

1. INTRODUCTION
The increase in digital information is providing users with more options to choose, which leads to information overload. The personalized recommendations take in to account the preferences of user to give the recommendations according to his/her interests and are likely to be preferred by the user. The recommended items can be products like books, music, videos, movies, electronic goods or resources like learning resources, papers, news, or people like friends, peers or activities like download, watch and connect. The examples of the web sites giving recommendations are products in Amazon, people in LinkedIn, music in Lastfm, movies in Netflix and friends in Facebook [1].

The recommendation algorithm predicts the ratings or ranking of unseen items for a user and recommends the top N items which he/she can explore more. The quality of recommendation depends on how much relevant these items are to the user. Though the relevance of recommendations can be increased with the use of user and item profiles with data mining and machine learning techniques, it is difficult to make accurate recommendations as rating predictions are done with the available user and item data. Currently the context based (aware) recommendation is gaining importance as the user may have rated the items according to context and also he/she needs the recommendation according to the current context. The context can be current situation, time or location. The context along with the other data such as user and/item profile can be applied to increase the relevance and the accuracy of recommendation [2]. With the use of digital technologies like mobile, social networking and e-commerce in day to day life, the context of user, item or the actions can be captured and stored. The user's interest differs in different situations which can be incorporated in context based methods.

2. TYPES OF RECOMMENDATION METHODS
The recommender systems recommend the items using mainly three basic methods which use the algorithms from data mining, information retrieval and machine learning. These are content based filtering, collaborative filtering and hybrid filtering. In addition to these approaches, new methods have been proposed like context based methods, social network based methods and soft computing methods [3].

2.1 Content Based Filtering
Content based recommender systems are the earlier recommender systems that have been developed. The items similar to the ones which are positively rated or liked by the user in the past are recommended. The user and item profile consists of attributes or features of user and item respectively. For example, a movie attributes can be movie id, title, genre, actor and director. The user attributes can be user id, user address, age, user purchases, user rating and user reviews. In this above example of user and item profile, if a user likes the horror movies, the horror movies are recommended to the user, which are not yet liked by him [4]. The main steps of content based filtering are,

1. Extract the item attributes to generate item profile for all items.
2. Generate the user profile for each active user.
3. Compare the item profile with user profile.
4. Recommend the items which match the user profile more and which are not seen by the user.

The content based filtering is applied to web pages, movies and books using textual description of items [5], [6]. The item profile is created using keywords in the description of the text. The number of keywords and importance of keywords in a document are determined using term frequency/inverse document frequency (TF-IDF) [7]. The term frequency of keyword k_i in document d_j is

$$TF_{i,j} = \frac{f_{i,j}}{max f_{z,j}} \quad (1)$$

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

where $f_{i,j}$ is the frequency of keyword k_i in document j and $f_{z,j}$ is the maximum of frequencies of all keywords in document j. N is the total number of documents and n_i is the number of documents containing keyword i. The weight of keyword k_i in document j is $w_{i,j}$ defined in terms of TF and IDF as,

$$w_{i,j} = TF_{i,j} \times IDF_i \qquad (3)$$

Each document is represented as a vector of weights of each keyword,

$$d_j = (w_{1,j}, w_{2,j} \dots w_{k,j}) \qquad (4)$$

User profile is created with the items which the user has liked in the past. Based on a set of web pages that were rated as relevant or irrelevant by the user, the Naïve Bayesian classification is used for classification of unrated web pages in [8].

The classification is used to build the user profile and its updation in content based filtering before similarity calculation in [9]. In this method the decision tree C4.5 algorithm is used to classify only old sellers as trustworthy and untrustworthy using seller attributes and customer transactions. Top K sellers who are most similar to user are recommended.

With the development of web 3.0, social network information of users plays an important role in generating user profile. The content generated by users which is user generated content (UGC) on social network like Facebook, Twitter or LinkedIn can be used for extracting the information about users' preferences and has become an important content for recommendation. The social network data like LinkedIn is used in [10] to extract the users connected to active user and the specialties, interest, groups and associations are also extracted and stored in documents. The user profile is matched with the document profile with cosine similarity to find the top K documents. To include the information of content on the web, friend of a friend (FOAF) to analyze social network services and really simple syndication (RSS) to analyze contents is used in [11].

With the development of mobile technology, recommendation systems are developed for mobile systems also. M-learning content recommendation in [12] calculates the M–learning similarity, social interaction like common friends and popularity. The content is recommended to a learner using Bayes classification. The content based filtering can recommend too many similar items which can be overcome by combining with the other methods.

2.2 Collaborative Filtering

In collaborative filtering the recommendations are given to a user who is currently using an application and is called as an active user. The collaborative filtering works on the assumption that the active user will prefer the items liked by the users who have the tastes same as him/her. The similar users of an active user can be found by considering the ratings given by the users for the same items. This is known as user based collaborative recommendation. The main steps of collaborative filtering are

1. For all users U and items I and ratings R of users on items, form U X I matrix containing ratings of user on item as elements.
2. Find the similarity of the active user *u*, with all other users of the system.
3. Find the *k* most similar users from above which form *k* nearest neighbours of active user u.
4. Predict the ratings of user *u* on item *i*, which is not seen by the user *u*.
5. Repeat the step 4 for all items which are not seen by user *u*.
6. Select the top N items from the predicted ratings for recommendations for user *u*.

The user based or memory based collaborative filtering uses the whole user item matrix to generate the prediction of ratings by the active user [13]. For a set of users $U=\{u_1,u_2,u_3 \dots u_n\}$ and set of items $I=\{i_1,i_2,i_3 \dots i_n\}$, rating of a user u on item i is $r_{u,i}$. If user u is an active user, each user is represented by a vector of his ratings with k items where usually $k<n$. For finding the similarity between users u_x and u_y, the user vectors $u_x=\{r_{x,1},r_{x,2},r_{x,3} \dots r_{x,k}\}$ and $u_y=\{r_{y,1},r_{y,2},r_{y,3} \dots r_{y,k}\}$ are used where $(1,2,3...k) \in$ I, are the items rated by both u_x and u_y. The similarity measures used are like Pearson coefficient, cosine measure, adjusted cosine measure, Jaccard coefficient. For cosine similarity,

$$sim(u_x, u_y) = \frac{\sum_{i=1}^{k} r_{u_x,i}\ r_{u_y,i}}{\sqrt{\sum_{i=1}^{k} r_{u_x,i}^2 \sum_{i=1}^{k} r_{u_y,i}^2}} \qquad (5)$$

For Pearson correlation,

$$sim(u_x, u_y) = \frac{\sum_{i=1}^{k}(r_{u_x,i} - r_{u_x})(r_{u_y,i} - r_{u_y})}{\sqrt{\sum_{i=1}^{k}(r_{u_x,i} - r_{u_x})^2 \sum_{i=1}^{k}\left(r_{u_y,i} - r_{u_y}\right)^2}} \qquad (6)$$

where r_{ux} and r_{uy} are average ratings for user u_x and u_y.

For Tanimoto or Jaccard coefficient,

$$sim(u_x, u_y) = \frac{\left|I_{u_x} \cap I_{u_y}\right|}{\left|I_{u_x}\right| + \left|I_{u_y}\right| - \left|I_{u_x} \cap I_{u_y}\right|} \qquad (7)$$

where $|I_{ux}|$ and $|I_{uy}|$ are number of items rated by user u_x and u_y respectively. The k nearest neighbours are taken as users with maximum similarity values to active user u. The rating prediction is done on items i_l not seen by user u as

$$r_{u,i_l} = r_u + \frac{\sum_{v=1}^{p} sim(u,v)(r_{v,i} - r_v)}{\sum_{v=1}^{p} sim(u,v)} \qquad (8)$$

where users 1...p have rated item i and who are similar to user u, r_u and r_v are average rating of user u and v respectively.

The item based collaborative filtering computes the item similarity instead of user similarity in [14]. The item similarity can be computed with correlation or cosine or adjusted cosine measure. The active user has a set of items rated by him $I=\{i_1,i_2,...i_n\}$. The items similar to target item i (for which the rating is to be predicted) are taken from I and from that the k most similar items are selected. The target item rating is predicted by computing the weighted sum of ratings given by the active user for k similar items. With $S_{i,j}$ as the similarity of item i with item j, the rating of user u on item i is,

$$r_{ui} = \frac{\sum_{j=1}^{k} S_{i,j} \times r_{u,j}}{\sum_{j=1}^{k} S_{i,j}} \qquad (9)$$

For item based collaborative filtering, the accuracy in terms of mean absolute error (MAE) is better than user based algorithm and the online computation of similarity is reduced.

Presently collaborative filtering algorithms incorporate other features of user and item to improve the quality of recommendation. A typicality-based collaborative filtering approach named TyCo, in which the neighbours of users are found based on user typicality in user groups instead of co-rated items of users is proposed in [15].

A multi criteria item based collaborative filtering framework based on item based and multi criteria recommendation is given in [16]. The rating is not one single rating for an item. One item has many ratings depending on the criteria. The user item matrix will have many ratings expressed as $R=U\ X\ IR_1, R_2, R_3 \dots R_c$ where c is the number of criteria. The item similarity is computed using the average of similarity of each rating criteria or using different distance measures. The rating prediction can be done separately for each criteria and overall rating is predicted using regression.

2.3 Hybrid Filtering

The hybrid approach combines collaborative and content based methods to overcome limitations of both methods. Different ways in which the hybrid system can be combined are 1) By combining the predictions of content and collaborative filtering after separately implementing both. 2) By using content based properties in collaborative approach or reverse. 3) By modeling content and collaborative approach together.

A hybrid recommendation algorithm for e-learning is proposed in [17]. It uses item based collaborative filtering for rating prediction and sequential pattern mining for user access patterns of items. The sequential pattern mining is used to assign weights to each item. The hybrid algorithm performs better than collaborative filtering.

A hybrid approach for tourism system is proposed in [18] with associative classification. The users are grouped according to user attributes and item attributes in to clusters. The rules are generated with associative classification. The active user's last transaction is used to assign the groups of users for active user. The items related with those groups are recommended to the user.

A hybrid recommendation system on cloud is given in [19]. The hybrid framework consists of different recommendations for different pages like login, browse catalogue, search and basket. User login page uses the collaborative filtering with user interesting model. User interesting model and rating model are combined for similarity computations with map reduce.

3. EVALUATION OF RECOMMENDATION METHODS

There are number of measures which are used to evaluate the performance of various recommendation algorithms. The quality of recommendation depends on prediction accuracy, relevance and efficiency of the system. Statistical accuracy metrics measure the difference between the predicted rating and actual rating [20]. Mean absolute error (MAE) measures the deviation of predictions generated by recommender system to actual values. The MAE for each user *i* is calculated for *n* items and average of all MAE for *m* users is taken. Lower MAE corresponds to accurate recommendation. Given $ar_{i,j}$ as predicted rating and $r_{i,j}$ as actual rating,

$$MAE = \frac{\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\left|ar_{i,j}-r_{i,j}\right|/_{n}\right)}{m} \qquad (10)$$

Root mean square error(RMSE) is the square root of the average of square of loss of absolute error over the whole test set. Coverage measures the percentage of items for which the filtering algorithm can provide the predictions.

$$Coverage = \frac{\sum_{i=1}^{m} n_{p,i}}{\sum_{i=1}^{m} n_i} \qquad (11)$$

where n_i *is* number of items for which user *i* has given ratings and $n_{p,i}$ is number of items for which the predictions are given for user *i* and *m* is number of users [21].

For top N items, the need is to know whether the user will purchase some or all items in the list, to evaluate the value of the list. This can be measured with precision, Recall, and F1. The dataset is divided in to two training and test disjoint sets. The recommendation algorithm is applied on the training set to generate the top N set. The items in the test set and items in the top N which are same form the hit set or relevant items. Precision is the ratio of number of items relevant in the top N set to number of top N recommendation. Recall is the number of relevant items in top N set to total number of test set items. F1 is the harmonic mean of precision and recall.

$$F1 = \frac{2*precision*recall}{precion+recall} \qquad (12)$$

4. CONTEXT BASED RECOMMENDATION METHODS

The traditional recommendation methods incorporate the data about users, items and implicit or explicit ratings to predict the ratings for items to recommend top N items which have higher predictions and are not seen by the user. But the items preferred by any user also depend on the context at the time of user interaction with the system. The context is the information that can characterize an event or situation [22]. This context can be about user, item or the activity of interaction. For example, old people prefer philosophy books where as young people prefer action thriller books. In this case the age information about the user can be the context. Tourists prefer beaches in summer season. Here the time or season of interaction with the system is the context. The context influences the decision of any user in many e-commerce or web based applications. Usually the user and item attributes like age of person, price of an item are taken as user and item profile respectively and the information like time of purchase, company of people when watching TV or movie, location of user are taken as the context in many of the context based recommendation systems.

4.1 Approaches in Context Based Recommendation

The recommendation using context has three main approaches which are pre filtering, post filtering and contextual modeling [23]. Pre filtering uses the context to reduce or filter the information of user item matrix before applying any recommendation method. The Post filtering uses context to reduce or filter the recommendation list that is generated after applying recommendation method. The contextual modeling incorporates the context in to the recommendation method. The pre and post filtering methods can use the existing recommendation algorithms for recommendation whereas contextual modeling modifies the existing recommendation method.

4.1.1 Pre Filtering Methods

A multidimensional view of recommendation was proposed in [24] by incorporating context as third dimension in addition to user and item as first two dimensions in the user item matrix. In this the reduction based method was proposed and implemented which reduces the user item matrix to contain the ratings of items given in the specific context. For user *U*, item *I* and context *T*(time) dimensions, $\forall (u,i,t) \in (UxIxT)$ the rating prediction function *R* is,

$$R(u,i,t)_{UxIxT}^{D} = R(u,i)_{UxI}^{D(Time=t)(u,i,r)} \qquad (13)$$

where *D*(*Time=t*)(*u,i,r*) denotes the rating set obtained by selecting only those records where time dimension has value *t* which is taken as the contextual segment with time as context and with value *t*. It is shown that the reduction based method outperforms the user based collaborative filtering for some contextual segments. The contextual segment on which the reduction based method outperforms depends on the application.

Item splitting method proposed for the pre filtering approach in [25] uses context to split items and applies the traditional CF on the *U X I* matrix for recommendation. If the ratings for item *i*, are different under context $c=c_j$ and $c\neq c_j$, then the item is split in to two with one having ratings of item *i* when $c=c_j$ and other having ratings of item *i*, when $c\neq c_j$. The rating predictions for all items not rated by user are computed with modified *U X I* matrix and top K items with highest predicted ratings are recommended. The context based approach with reduction and item splitting have better accuracy than context free approach.

The context as three 'x' months duration from current date (time) is used to define the contextual segments for pre filtering in [26]. The Fuzzy inference system (FIS) was used to obtain the recommendation for popularity. The context is given by the two input variables, item popularity and user participation and recommendation is output variable. The output weight value given by recommendation for each context is used to calculate the average prediction of rating for an item.

$$r_{u,i} = \frac{w_{c_1,i}r_{c_1,u,i}+w_{2,i}r_{c_2,u,i}+w_{c_3,i}r_{c_3,u,i}}{w_{c_1,i}+w_{c_2,i}+w_{c_3,i}} \qquad (14)$$

Here $r_{u,i}$ is average prediction of rating of user u for item i, $r_{cj,u,i}$ is the prediction of rating for item i for user u in context c_j and $w_{cj,i}$ is the weight for recommendation obtained by FIS for an item i in context c_j. The pre filtering method improves the recommendation process, as only relevant data is taken for recommendation.

A scalable context aware recommendation system proposes pre filtering with clustering [27]. The clustering of users is done before collaborative filtering to reduce the size of user item matrix. The users are clustered with hierarchies according to their demographic values as context. The users belonging to the active user's cluster are taken for recommendation matrix for collaborative filtering. The run time performance of collaborative filtering increases by a factor of k if k equal partitions of users are created.

4.1.2 Contextual Modeling Methods

Contextual Modeling with matrix factorization technique gives better accuracy than pre filtering [28]. The context is modeled as part of matrix factorization to predict the user rating in a given item. The predicted rating of user *u* on item *i* in contexts c_1 to c_k is

$$r_{u,i,c1\ldots ck} = \overrightarrow{v_u}.\overrightarrow{q_i} + \bar{i} + b_u + \sum_{j=1}^{k} B_{i,j,cj} \quad (15)$$

where v_u and q_i are d dimensional real valued vectors representing the user *u* and the item *i*. $\bar{i}$ is the average of the item *i* ratings in the data set R, b_u is the baseline parameter for user *u*. $B_{i,j,cj}$ are the parameters modeling the interaction of the contextual conditions and the items. In order to generate rating predictions, the model parameters should be learned using the training data.

Contextual modeling with user similarity is given in [29]. Context is defined with hierarchical structure. The context is incorporated in prediction of rating as a filter to selection of neighbours for active user. A user u has contextual profile for each context k as *prof*(u,k) and the active user's neighbours are selected from users who have profile in that context. The contextual modeling approach performs better with non contextual approach and comparable to pre filtering approach.

The contextual modeling with concept hierarchy for user similarity approach is given in [30]. The user and items are profiled as concept vectors and cosine similarity is measured between user and items not seen by a user and top k items are recommended. The user satisfaction is evaluated on a learning portal and found to be satisfactory.

4.1.3 Post Filtering Methods

The comparison of pre and post filtering is done in [31]. The exact pre filtering approach is compared with weight and filter approach of post filtering. The e commerce dataset with time of year and Amazon transactions with intent of purchase were used for evaluation. The post filtering with filter method has better performance than pre filtering. Post filtering with good filter can give better results.

The recommendation with context graph (CGR) proposed in [32] uses the random walk approach for finding the relevance of an item *i* for user *u*. The users, items, attributes of users and items and contexts are represented as vertices of graph and relevance of item an *i* for user *u* is computed as P(i | u) with weights for edges connecting the vertices. The post filtering is applied to this relevance score to find the rank of an item. Given an instance of the context factors $C = \{1, \ldots \ldots c_n\}$, the likelihood of an unseen item $i \in I$ to be accessed by $u \in U$ can be estimated as,

$$P(i|u) = P(i|C)P(i|u) \qquad (16)$$

$$P(i|C) \propto P\sum_{c_k\in C} P(i\,|c_k)P(c_k|\,C) = \frac{1}{|C|}\sum_{c_k\in C}\frac{freq\,(i,c_k)}{\sum_j freq(j,c_k)} \qquad (17)$$

where freq(i, c_k) is the occurrence frequency of item i, given the context condition c, and $P(c_k\,|\,C)$ takes $1/\|C\|$ for simplicity. For the estimation of $P(i \mid u)$, CGR based method or collaborative filtering approaches, or their combinations can be used.

5. CHALLENGES AND FUTURE TRENDS

The context based recommendation systems are performing better than the non contextual systems in case of contextual factors affecting the recommendation. The comparison of context based methods is given in Table 1. The context based methods use either pre filtering, post filtering or contextual modeling. The context is modeled as a hierarchy of contexts. The evaluation measures used are mostly accuracy and relevance and which can be extended for user satisfaction, diversity and trust. The challenges like data sparsity, cold start, scalability, diversity and privacy are still to be improved in current context of information availability. Considering only the context leads to sparsity problem which has only fewer ratings in user item matrix to predict rating for an item. The cold start problem in which the rating prediction cannot be done when a new user or item is added is to be handled in all recommendation systems.

Table 1. Comparison of Context Based Methods

Sr. No.	Title / Author/ Publisher	Context Information	Approach/ Algorithm combined with context	Evaluation Parameter	Data set used	Advantage	Limitation
1.	"Incorporating contextual information in recommender systems using a multidimensional approach" / Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin / ACM, 2005 [24]	Hierarchy of contexts	Pre filtering / User based CF	MAE, Precision, Recall	Real world Generated dataset	Context information is used only when it out performs	Computation of outperforming segments is to be done
2.	"Experimental evaluation of context-dependent collaborative filtering using item splitting." Baltrunas, Linas, and Francesco Ricci" / Springer, 2014 [25]	Tree concept Hierarchy	Pre filtering / User based CF	MAE, Precision, Recall	Real world Generated dataset, Yahoo dataset	Split is applied only when it is influencing rating	Time complexity of split for more contexts and Sparsity increases if too many items are split
3.	"A Pre-filtering Based Context-Aware Recommender System using Fuzzy Rules"/ Ramirez-Garcia, Xochilt, and Mario Garcia-Valdez / Springer 2015 [26]	Time duration	Pre filtering / User based CF	MAE	Movielens	Reduces the sparsity problem by Fuzzy inference system	Single context and not well distributed Test Data set is used
4.	"SCARS: A scalable context-aware recommendation system" / Datta, Suman, Joydeep Das, Prosenjit Gupta, and Subhashis Majumder / IEEE, 2015 [27]	Hierarchy of User attributes	Pre filtering / User based CF	MAE, RMSE, Precision, Recall	Movielens	Decreases the runtime for calculation of user similarity	Only specified user profile is used for clustering which may result in sparsity
5.	"Matrix Factorization Techniques for Context Aware Recommendation" / Linas Baltrunas, Bernd Ludwig, Francesco Ricci /ACM 2011 [28]	Parameters of Matrix Factorization	Modeling / Matrix factori-zation	MAE	Real world generated data set, Yahoo Webscope,	Generalizes contextual factors and conditions	Small data sets need less contextual factors
6.	"A contextual modeling approach to context-aware recommender systems" / Panniello, Umberto, and Michele Gorgoglione / CARS 2011 [29]	Tree concept Hierarchy	Modeling (user selection)/ User based CF	F measure, RMSE, Precision, Recall	Ecommerce website	No searching of best performing pre or post filtering is needed	Modification in existing recommender method
7.	"Contextual model of recommending resources on an academic networking portal" / Pandey, Anoop Kumar, Amit Kumar, and Balaji Rajendran./ CS & IT 2013 [30]	Tree Concept hierarchy	Modeling/ Content based	User satisfaction	Learning portal	User profile and item profile are created with concepts	Concept hierarchy to be defined and specific to learning portal
8.	"Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems." Panniello, Umberto, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone/ACM,2009 [31]	Hierarchy of contexts	Post filtering/ User based CF	F measure, MAE	Ecommerce , Amazon transactions	Only rearranging of recommend list with context is required	Post filtering method depends on filter method
9.	"Context Aware Recommendation via graph based Contextual modeling and Post filtering" / Hao Wu, Kun Yue, Xiaoxin Liu, Yijian Pei, and Bo Li / Hindwai, 2015 [32]	Context graph	Post filtering/ Random walk	Precision, Recall	LDOS CoMoDa, Trip Advisor	Any context can be modelled as graph	Post filtering gives less accuracy with CGR method

As the number of users and items are very large in current web applications, the scalability of the recommendation is to be considered. The diversity is recommending diverse items which user can prefer instead of typical items always. This has to be considered in new e commerce applications as the numbers of items are very large and different. The privacy of user information is to be considered when giving personalized recommendations.

As the recommendations are applied in domains like e-commerce, learning management systems, health care, mobile and cloud based applications, methods have to be developed to utilize the specific context for each of these.

The future trends can be

1. Combining the data mining algorithms with new information like social links, location and user actions as context with existing methods.
2. Context recommendation methods with implicit context inference and modeling.
3. Recommendation methods incorporating the context in recommendation process.
4. Developing context based recommendation systems for different domains like mobile, cloud and user applications.

6. CONCLUSION

The context based recommendation methods are currently being incorporated in many applications. The context can be extracted from user interaction, social network and sensors. The k-nearest neighbor (KNN), matrix factorization, clustering and classification methods are used to model the recommendation process. The contextual modeling uses the context in recommendation process itself which requires changing the existing method. The pre filtering methods increase the scalability as well as use existing collaborative and hybrid methods. Pre filtering methods also help to analyze the effect of context on recommendation. The performance of pre or post filtering depends on the application. Our future work will include the pre filtering method for recommendation incorporating the rule mining techniques to reduce the user, item and rating data according to context and to provide accurate recommendations. Thus the context based recommendation methods try to improve the accuracy, relevance as well as user satisfaction in recommendation systems.

7. REFERENCES

[1] Ricci, Francesco, Lior Rokach, and Bracha Shapira "Introduction to recommender systems" in Recommender Systems handbook, pp. 1-35, Springer US, 2011.

[2] Baltrunas, Linas, Kaminskas Marius, Ludwig Bernd, Moling Omar, Francesco Ricci, Aydin Akan, Luke Karl Heinz and Schwaiger Ronald, "Incarmusic: Context-aware music recommendations in a car," In Proceedings of the EC-Web Conference, pp. 89–100, Springer 2011.

[3] J.Bobadilla, F.Ortega, A. Hernando, A. Gutierrez, "Recommender systems survey," Knowledge-Based Systems, vol. 46, pp. 109-132, 2013.

[4] Lops, Pasquale, Marco De Gemmis, and Giovanni Semersro. "Content based recommender systems," in Recommender Systems handbook, pp. 73-105, Springer US, 2011.

[5] Nessel, Jochen, and Barbara Cimpa. "The movieoracle-content based movie recommendations," in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol.3, pp.361-364, 2011.

[6] Rana, Chhavi, and Jain Sanjay Kumar. "Building a Book Recommender system using time based content filtering," in WSEAS Transactions on Computers, vol. 11, no. 2, pp. 27-33, Feb 2012.

[7] Dr. Emmanuel M.,Saurabh M Khatri,Dr. Ramesh Babu D. R.A, "Novel scheme for Term weighting in Text Categorization : Positive Impact factor", in International Conference on Systems, Man, and Cybernetics, pp. 2292-2297, IEEE, 2013.

[8] Pazzani, Michael J., and Daniel Billsus. "Content-based recommendation systems," in The adaptive web, pp.325-341, Springer Berlin Heidelberg, 2007.

[9] Lee, Seungsup, Kenuho Choi, and Yongmoo Suh. "A personalized trustworthy seller recommendation in an open market," Expert Systems with Applications, vol.40, no.4, pp. 352-1357, 2013.

[10] Lops, Pasquale, Marco de Gemmis, Geovanni Seneraro, Fedelucio Narducci and Cataldo Musto "Leveraging the linkedin social network data for extracting content based user profiles," in Proceedings of the 5th ACM conference on Recommender Systems, pp.293-296, 2011.

[11] Sohn, Jong,Soo, Un-Bong Bae and In-Jeong Chung "Content Recommendation Method Using Social Network Analysis," Wireless personal communications, vol.73, no.4, pp. 1529-1546, 2013.

[12] Chao, Han-Chieh, Lai Cin-Feng, Chen Shih-Yeh, and Huang Yueh-Min, "A M learning Content Recommendation service by exploiting Mobile social interactions," IEEE Transactions on Learning Technologies, vol.7, no.3, pp. 221-230, July-Sept. 2011.

[13] Adomavicius, Gediminas, and Alexander Tuzhilin, "Towards next generation of recommender systems: A survey of state of the art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol.17, no.6, pp. 34-739, June 2005.

[14] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, "Item based collaborative filtering recommendation algorithms," in WWW '01: Proceedings of the 10th International conference on the World Wide Web, pp. 285-295, ACM, 2001.

[15] Cai,Yi,Ho-fung Leung, Qing Li, Huaqing Min, Jie Tang, and Juanzi Li. "Typicality-based collaborative filtering recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no.3,pp. 766-779, 2014.

[16] Bilge Alper, and Cihan Kaleli, "A multi-criteria item-based collaborative filtering framework," in 11th International Joint Conference on Computer Science and Software Engineering, pp. 18-22, IEEE, 2014.

[17] Chen, Wei, Zhendong Niu, Xiangyu Zhao, and Yi Li. "A hybrid recommendation algorithm adapted in e-learning environments," World Wide Web, vol. 17, no. 2, pp. 271-284, Springer, 2014.

[18] Lucas, Joel P., Nuno Luz, María N. Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, and Constantino

Martins. "A hybrid recommendation approach for a tourism system," vol. 40, no. 9 pp. 3532-3550, Elsevier, 2013.

[19] Dong, Fang, Junzhou Luo, Xia Zhu, Yuxiang Wang, and Jun Shen. "A Personalized Hybrid Recommendation System oriented to e-commerce mass data in the cloud," IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1020-1025, 2013.

[20] Vozalis, Emmanouil, and Konstantinos G. Margaritis. "Analysis of recommender systems algorithms," In The 6th Hellenic European Conference on Computer Mathematics & its Applications, pp. 730-745, June 2003.

[21] Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems (TOIS) vol.22, no.1, pp. 5-53, ACM, 2004.

[22] Gregory D Abowd, Anind K. Dey, Peter J Brown, Nigel Davies, Mark Smith and Pete Steggles, "Towards a better understanding of context and context awareness," In Handheld and ubiquitous computing, pp. 304-307, Springer 1999.

[23] Adomavicius, Gediminas, and Alexander Tuzhilin, "Context-aware recommender systems," in Recommender Systems handbook, pp. 217-253, Springer US, 2011.

[24] Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," ACM Transactions on Information Systems (TOIS), vol. 23, no.1, pp. 103-145, ACM, 2005.

[25] Baltrunas, Linas, and Francesco Ricci. "Experimental evaluation of context-dependent collaborative filtering using item splitting," User Modeling and User-Adapted Interaction, vol.24, no.1-2 pp. 7-34, 2014.

[26] Ramirez-Garcia, Xochilt, and Mario Garcia-Valdez. "A Pre-filtering Based Context-Aware Recommender System using Fuzzy Rules," Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization, pp. 497-505, Springer International Publishing, 2015.

[27] Datta, Suman, Joydeep Das, Prosenjit Gupta, and Subhashis Majumder. "SCARS: A scalable context-aware recommendation system," in 3rd International Conference on Computer, Communication, Control and Information Technology *(C3IT)*, pp. 1-6. IEEE, 2015.

[28] Linas Baltrunas, Bernd Ludwig, Francesco Ricci, "Matrix Factorization Techniques for Context Aware Recommendation," In Proceedings of 5th ACM conference on Recommender Systems, pp. 301-304, 2011.

[29] Panniello, Umberto, and Michele Gorgoglione. "A contextual modeling approach to contexts-aware recommender systems," In Proceedings of the 3rd Workshop on Context-Aware Recommender Systems, 2011.

[30] Pandey, Anoop Kumar, Amit Kumar, and Balaji Rajendran. "Contextual model of recommending resources on an academic networking portal," *CCSIT*, 2013.

[31] Panniello, Umberto, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. "Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems," In Proceedings of the 3rd ACM conference on Recommender systems, pp. 265-268, 2009.

[32] Hao Wu, Kun Yue, Xiaoxin Liu, Yijian Pei, and Bo Li "Context Aware Recommendation via graph based Contextual modeling and Post filtering," International Journal of distributed Sensor Networks, Hindwai, 2015.