Computing and Informatics, Vol. 38, 2019, 597–620, doi: 10.31577/cai_2019_3_597

View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Computing and Informatics (E-Journal - Institute of Informatics,...

FUZZY SIDE INFORMATION CLUSTERING-BASED FRAMEWORK FOR EFFECTIVE RECOMMENDATIONS

Mohammed Wasid

Department of Computer Science & Engineering
Government Engineering College, Bharatpur-321303, India
&
Department of Computer Engineering
Aligarh Muslim University, Aligarh-202001, India
e-mail: erwasid@gmail.com

Rashid Ali

Department of Computer Engineering
Aligarh Muslim University, Aligarh-202001, India
e-mail: rashidaliamu@rediffmail.com

Abstract. Collaborative filtering (CF) is the most successful and widely implemented algorithm in the area of recommender systems (RSs). It generates recommendations using a set of user-product ratings by matching similarity between the profiles of different users. Computing similarity among user profiles efficiently in case of sparse data is the most crucial component of the CF technique. Data sparsity and accuracy are the two major issues associated with the classical CF approach. In this paper, we try to solve these issues using a novel approach based on the side information (user-product background content) and the Mahalanobis distance measure. The side information has been incorporated into RSs to further improve their performance, especially in the case of data sparsity. However, incorporation of side information into traditional two-dimensional recommender systems would increase the dimensionality and complexity of the system. Therefore, to alleviate the problem of dimensionality, we cluster users based on their side information using k-means clustering algorithm and each user's similarity is computed using the Mahalanobis distance method. Additionally, we use fuzzy sets to represent the side information more efficiently. Results of the experimentation with two benchmark datasets show that our framework improves the recommendations quality and predictive accuracy of both traditional and clustering-based collaborative recommendations.

Keywords: Recommender systems, collaborative filtering, Mahalanobis distance, k-means clustering, multi-criteria, demographic recommender

1 INTRODUCTION

Recommender System (RSs) is the information filtering software tool which gives suggestions to internet users for the products which are more likely to be preferred by them or relevant to their choice [1]. Here, a suggestion can be from any domain, such as which movie to watch, which song to listen, what products to buy, or which online news to read. Collaborative filtering (CF) is the most used and popular technology implemented in both industry and academia due to its simplicity and accurate enough recommendations ability [2]. The CF technique is further classified into product-based and user-based methods. The core idea of the product-based method is to provide product suggestions to users based on the other similar products, while the user-based method generates recommendation to a user (target user) by finding a set of users who have high correlations with this user. In both ways, finding similar users (products) to target user (product) is the crucial step for CF technique [3]. Currently, most of the CF similarity measures are based on commonly rated products. Although these CF recommendation methods are popular and widely used, they still suffer a number of inadequacies, including [4]:

- Data sparsity: This is a very usual problem in collaborative recommenders where users give ratings to a small set of products from a broad set of products available in the system. The actual problem occurs in the neighborhood set generation phase, where a very few or no common product ratings are available for similarity computation between users which leads to invalid neighborhood set formation.
- Cold-start: This problem arises in a scenario similar to that of data sparsity. In this problem, it is tough to produce recommendations for users who are newly introduced into the system or have not rated a single product yet.
- Multidimensionality: Traditional RSs works fine in case of two dimensions, i.e., users and products, but tends to fail when a recommendation is needed for a system having more than two dimensions. Therefore, the curse of dimensionality is one of the major issues in classical recommender systems.

All of the above-listed problems are approximately dependent on each other; for instance, while handling data sparsity issue through some user-product features in the system, the multidimensionality issue comes into the picture [5]. Therefore, to tackle these problems, a compact model is required where the fusion of user-product side information into traditional recommender system does not affect its dimensions.

Although researchers are working towards the improvement in the accuracy of recommender systems using the overall user-product single-criterion ratings [3, 6], it has been seen that the user's demographic data, products description, contextual and multi-criteria rating factors significantly influence the utility of recommendations. In this work, we incorporate

1. user demographic information (user age) and
2. multi-criteria ratings as side information for two different recommenders.

A collaborative filtering technique with demographic information is known as demographic recommender systems. We have used user age as the third dimension to alleviate data sparsity issue. Let us suppose a user x who is new in the system and has no past ratings (user cold-start), then, in such case, other users can only be matched with him/her by his/her age. Hence the young age group users will be closer to each other rather than the old age people. Whereas, a multi-criteria recommender system (MCRS) can provide more effective and accurate suggestions to users as compared to the classical RSs. In MCRS the preference of a user is represented based on several aspects of the products [7]. Instead of having a single overall rating for a product, MCRS represents products using multiple components to attain preference of a user in depth [7, 8]. For example, hotels can be evaluated more effectively based on their different components (cleanliness, rooms, cuisines, and price) instead of evaluating on a single overall rating. Now, our job is to find a technique which can combine these side information with traditional recommendation systems without dimension expansion and further improve their accuracy. After connecting the side information into the system, we have to identify a method to compute effective similarity between user models with side information. There is a need for selecting an efficient similarity measure for a top-N neighborhood set generation.

Therefore, in this work, we proposed a framework which has achieved our goals. In our framework, we treated the side information (user age or multi-criteria ratings) as a clustering parameter for k-means clustering. The users are clustered using their age (or multi-criteria ratings) and this reduces the user search space while forming neighborhood set of the user. Each user is assigned to his/her most similar side information cluster. Further, we use Mahalanobis distance measure to compute the distance between users within the group [35]. Our work follows a simple and effective method to generate more expert recommendations to the users. The contribution of the paper can be outlined as follows:

- We propose a recommendation method that improves the accuracy of collaborative filtering and is based on side information, Mahalanobis distance, and k-means clustering algorithm.

- We present a novel framework to deal with the curse of dimensionality in collaborative filtering recommender systems. No method in the literature deals with the multi-criteria and demographic multi-dimensionality problem using a single framework.
- A novel user profile is built on user-product side information, where fuzzy logic is used to handle the sparsity and uncertainty and reduces the complexity of the system.
- Modified Mahalanobis distance measure is proposed for users matching.
- We perform extensive experiments on two real datasets, namely Yahoo! Movies for multi-criteria ratings and MovieLens for user age feature. The results show that our proposed framework can deal with the multi-dimensionality and data sparsity issues effectively and accurately as compared to other traditional techniques.

This paper is organized as follows: Section 2 introduces the background and related work. The proposed recommendation framework is introduced in detail in Section 3. In Section 4, we evaluate the proposed method using the MovieLens 100 K and Yahoo! Movies datasets and compare it with the existing methods. We conclude the paper in Section 5.

2 BACKGROUND

2.1 Recommendation Techniques

Recommender systems employ different information filtering techniques to product recommendations based on the type of application. There are many techniques implemented in literature, but the Content-based (CB), Collaborative filtering (CF), and hybrid techniques are the major recommendation techniques [4]. We will discuss each of them in the following subsections.

2.1.1 Content-Based Technique

This technique suggests products similar to the ones user selected in the past. The core mechanism of this technique depends upon the content or feature of the past preferred products [1]. These contents or features are further used to build a user profile for each user. Thereafter, similarity between user profiles and other products are computed to obtain similar products.

2.1.2 Collaborative Filtering Technique

This is the most widely implemented technique in literature. It produces recommendations based on preferences of other like-minded users in the system. CF works in three steps. First, similarity is computed between users on the basis of their historical ratings. Secondly, the neighborhood set is formed by obtaining most similar users to the target user. Thirdly, predictions and recommendations are made through neighborhood users' collective ratings [2]. We can observe from above mentioned steps that the similarity computation is a critical step for CF technique and the performance of the system considerably depends on the quality of neighborhood set selection. Therefore, there is a need for a good mechanism to find neighbors of target user which can facilitate better recommendations to the users. We will discuss some of the well-known similarity measures in Section 2.2.

2.1.3 Hybrid Filtering

Here, more than one filtering technique is combined to improve the effectiveness of the recommender system. Hybrid filtering is used to remove drawbacks of each technique separately [19]. There are multiple ways to implement a hybrid recommender system; it can be implemented by combining separate recommender techniques or by adding content-based characteristics to collaborative model and vice-versa.

2.2 Similarity Measures

Similarity computation is an intermediate and primary step in collaborative filtering which is used for neighborhood set formation [10, 19]. Here user-product rating matrix is used for similarity computation. Most of the similarity measures fail in case of sparse data. In our work, we will perform experiments on the following similarity measures to compute the similarity/distances between users.

2.2.1 Pearson Correlation Coefficient (PC)

This is the most popular similarity computation method usually applied to memory-based CF [33]. In this method, the similarity between two users is based only on the ratings both users have given to products in the past. The PC is calculated as follows:

$$PC(x,y) = \frac{\sum_{n\in N_{xy}}(r_{x,n} - \bar{r_x})(r_{y,n} - \bar{r_y})}{\sqrt{\sum_{n\in N_{xy}}(r_{x,n} - \bar{r_x})^2}\sqrt{\sum_{n\in N_{xy}}(r_{y,n} - \bar{r_y})^2}} \tag{1}$$

where $r_{x,n}$ is the rating of user x on product n and $\bar{r_x}$ is the mean of the total ratings given by the user x. N_{xy} is the set of products commonly rated by both user x and y.

2.2.2 Cosine-Based Similarity (CS)

This method uses the concept of angle to compute the similarity among different users. This similarity is based on the cosine of the angle between two users x and y.

It is calculated using:

$$CS(x,y) = \frac{\sum_{n \in N_{x,y}} r_{x,n} \cdot r_{y,n}}{\sqrt{\sum_{n \in N_{x,y}} r_{x,n}^2} \sqrt{\sum_{n \in N_{x,y}} r_{y,n}^2}}. \tag{2}$$

2.2.3 Extended Jaccard Coefficient (JC)

This method can be used for continuous or discrete non-negative features and gets reduced to the Jaccard coefficient in case of the binary attributes as input. This coefficient, which is represented as *JC*, is defined by the following equation:

$$JC(x,y) = \frac{\sum_{n \in N_{x,y}} r_{x,n} r_{y,n}}{\sum_{n \in N_{x,y}} r_{x,n}^2 + \sum_{n \in N_{x,y}} r_{y,n}^2 - \sum_{n \in N_{x,y}} r_{x,n} r_{y,n}}. \tag{3}$$

2.2.4 The Mahalanobis Distance (MD)

This is a well-known distance measuring formula which is calculated using the inverse of the variance-covariance matrix of the dataset of interest [14, 15]. The *MD* for a single user x is computed similar to the concept of Euclidean distance method.

$$MD = \sqrt{(x - \bar{x}) vc_z^{-1} (x - \bar{x})^T} \tag{4}$$

where

$$vc_z^{-1} = \begin{bmatrix} \sigma_2^2 / det(vc_z) & -\rho_{12}\sigma_1\sigma_2 / det(vc_z) \\ -\rho_{12}\sigma_1\sigma_2 / det(vc_z) & \sigma_1^2 / det(vc_z) \end{bmatrix}$$

where σ_1^2 and σ_2^2 are the variances of the values of the first and second users respectively. $\rho_{12}\sigma_1\sigma_2$ is the covariance between the two users and $\det(vc_z)$ is the determinant of the variance-covariance matrix (vc_z), which is computed as follows:

$$vc_z = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}. \tag{5}$$

Since Equation (4) is used only for a single user x, we will discuss *MD* for multiple users in detail in Section 3.2.

2.3 Related Work

In our work, a user-based clustering has been applied to identify users with similar preferences based on user age or multi-criteria ratings. For example, in a restaurant, one may like the food, but not the service, or vice-versa. So, in the real-life scenario, the side information dramatically affects the overall preference of a user. There has been a lot of research done in the area of recommender systems using clustering, like a user-based clustering has been proposed using user-user similarity computations and resulting clusters are used for neighborhood set generation [10].

Ungar and Dean [21] clustered both users and products separately using k-means and Gibbs sampling. Here, users can be un-clustered using the number of products in each product cluster they have rated and vice-versa. Tsai and Chihli [22] proposed cluster ensembles method for collaborative filtering recommendation where they have used the self-organizing map and k-means clustering and three different ensemble methods. A novel clustering-based CF approach was developed where user groups are formed using a proposed method to reduce the impact of the data sparsity [20, 32]. After cluster formation, nearest neighbors are found from each user group to produce the useful recommendation to the users. Similarly, Liu [9] proposed an improved clustering-based collaborative filtering recommender method. In this approach the authors applied k-means clustering to cluster the users and then an enhanced similarity method was developed to generate nearest neighbors in the cluster for the target user. The problem of computational time has also been addressed using k-means clustering by clustering user-product ratings and generate nearest neighbors [23].

Liu et al. [11] presented a multi-criteria recommendation approach by clustering users based on their criteria preferences in a preference lattice. Authors showed that some set of criteria dominate the overall ratings and different users have their own different dominant set of criteria. A clustering and regression-based technique was proposed in [12, 13] to improve the predictive accuracy of multi-criteria CF: different clustering methods were used to detect similar customer segments and regression was used for learning important weights for the various quality factors. Nilashi et al. in [26] proposed a technique to solve scalability and sparsity problems of multi-criteria recommender systems using dimensionality reduction and Neuro-Fuzzy techniques. In their approach the Neuro-Fuzzy technique was used to solve sparsity problem, and scalability was handled using higher order singular value decomposition along with supervised learning (classification) methods. Similarly, in [24], a hybrid recommendation model was proposed to overcome the same issues by using ontology and dimensionality reduction techniques. The authors have used EM clustering to cluster user-product and Singular Value Decomposition (SVD) techniques for dimensionality reduction. Whereas, [25] shows a method which combines dimension reduction and user clustering in collaborative filtering in which the authors have used principal component analysis and SVD techniques for dimensionality reduction plus k-means and agglomerative hierarchical clustering techniques for user clustering. Authors in [30, 31] deal with the curse of dimensionality by handling data sparsity problem of CF technique. Xu et al. [27] used the clustering algorithm to cluster user profile and then combined it with product-based collaborative filtering to improve its performance. Furthermore, authors have incorporated fuzzy set theory to deal with different rating schemas and tackle the scalability issue of recommender systems.

Furthermore, many authors have chosen certain clustering parameters from the user-product profile features in literature. Frémal and Lecron in [16] presented a clustering based recommender system based on product's metadata: they have used movie genre as a clustering parameter. Since a single movie can have multiple genres, therefore, authors assigned a single product to multiple clusters and results from every cluster were combined using different weighting strategies. In the same way, [29] proposed a technique to cluster products based on the contents using k-means algorithms. Product-grain clustering [18] was introduced by choosing contexts as a clustering parameter using k-means algorithm. Further, these context clusters were incorporated into matrix factorization technique to overcome data sparsity, scalability and prediction quality issues. Wang et al. [28] proposed a new algorithm which clusters user attributes using k-means algorithm. Here, longitude and latitude of the user are considered as the clustering parameter and after cluster formation, the similarity of each user is calculated within the respective cluster.

The major problem with these works is that they all have used some additional techniques to alleviate the issue of dimensionality reduction which makes the system more complicated. A complex system will take more time to produce recommendations which may irritate the online user. Unlike the approaches mentioned above, our proposed framework is straightforward and requires no mathematical modelling for dimensionality reduction because we have treated the other dimensions as the feature of user cluster. Our work in this paper can be summarized twofold. First, we identify the clustering parameter for cluster segments and incorporate fuzzy sets to handle the uncertainty issue associated with them. After cluster formation, in the second fold, we applied different similarity measures and compared with our proposed Mahalanobis distance-based method. However, our goal is to study the impact of different similarity measures for clustering and non-clustering approaches.

3 THE PROPOSED FRAMEWORK

In this section, we give a description of our proposed framework which improves recommendation system's accuracy by identifying different users' belongingness into different clusters based on their side information. In the proposed technique, the actual neighbors of users are found based on their Mahalanobis distance within the user cluster. Unlike traditional similarity measures, the Mahalanobis distance not only considers the commonly rated products between the two users but also finds the variance and covariance between them which makes it more efficient to generate the more similar neighborhood set to the target user, thus, improving the effectiveness of the CF recommendation [15]. The framework is constructed according to the idea and description mentioned above. Figure 1 presents the architecture of the system. The proposed algorithm has three main components which are explained in the below subsections.

3.1 User Cluster Formation

In order to generate most relevant user clusters according to their side information, the first step is to identify the clustering parameter from the existing dataset. After

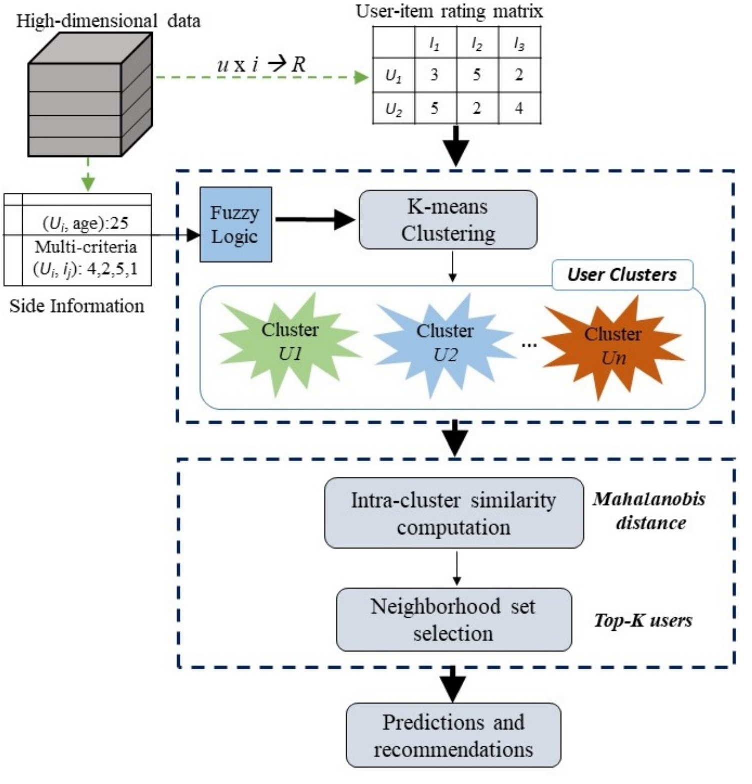


Figure 1. Clustering with fuzzy side information framework

which, the number of user cluster centers has to be defined. The next step is to cluster each user with any one of these pre-specified cluster centers based on their in-between distance (similarity). Those users are assigned to the cluster which has minimum distance (maximum similarity) from them, so that users with most common preferences are grouped in the same cluster. This process iterates until convergence criteria are met. The algorithm for creating user clusters is shown below.

3.1.1 Fuzzy Approach

Fuzzy set has been used to deal with the vague concepts, like 'old', 'short', 'poor', and so on. To incorporate the fuzzy sets into recommender system, proper fuzzification (designing of membership functions) will be required and an appropriate distance function will be needed to match the local and global similarities between different users.

Algorithm 1 Algorithm for user cluster formation

Input: Dataset with Side Information, number of clusters K
Output: Set of user clusters

Step 1: [*Initialization*]
Initialize randomly K clustering centers $\{c_1, c_2, c_3, .., c_k\}$ from the dataset.
Step 2: [*Assignment*]
a. Find the closest cluster center c for each user a using:
(i) Equation (8) in case of fuzzified side information OR
(ii) following Euclidean distance for non-fuzzified side information:

$$Edis_{c,a} = \sqrt{(M_{c,1} - M_{a,1})^2 + (M_{c,2} - M_{a,2})^2 + ... + (M_{c,n} - M_{a,n})^2}$$

where: $Edis_{c,a}$ - distance between the cluster center c and the user a;
n – number of user rated products;
$M_{c,n}$ – age or multi-criteria ratings of cluster center c for product n
(n=4 for multi-criteria ratings; n=1 for age side information);
$M_{a,n}$ – age or multi-criteria ratings of user a for product n.
b. Arrange user a by its distance from each clustering centers $\{c_1, c_2, c_3, .., c_k\}$.
c. Assign the user a with the nearest distance cluster center c.
Step 3: [*Convergence*]
a. Iteratively process cluster reassignment until convergence criteria is reached.
b. If convergence criteria met, then the algorithm terminates and returns a set of user clusters $\{c_1, c_2, c_3, .., c_k\}$, otherwise go to step 2.

Side information fuzzification: We have used the following fuzzy sets to deal with the uncertainty associated with the user-item side information (multi-criteria ratings and user age).

In our approach, multi-criteria ratings from Yahoo! Movies dataset are classified into six fuzzy sets, namely very bad (VB), bad, average, good, very good, and excellent (Exl) [6], as shown in Figure 2, following are the membership functions for these fuzzy sets:

$$P_{VB}(m) = \begin{cases} 1 - m, & m \leq 1, \\ 0, & m \geq 1, \end{cases} \tag{6a}$$

$$P_{t(v)}(m) = \begin{cases} 0, & m \leq v - 2, m > v, \\ m - v + 2, & v - 2 < m \leq v - 1, \\ v - m, & v - 1 < m \leq v \end{cases} \tag{6b}$$

where *t(v)* represents the bad, average, good, and very good for each value of $v = 2$, 3, 4, and 5, respectively.

$$P_{Exl}(m) = \begin{cases} 0, & m \leq 4, \\ m - 4, & 4 < m \leq 5. \end{cases} \tag{6c}$$

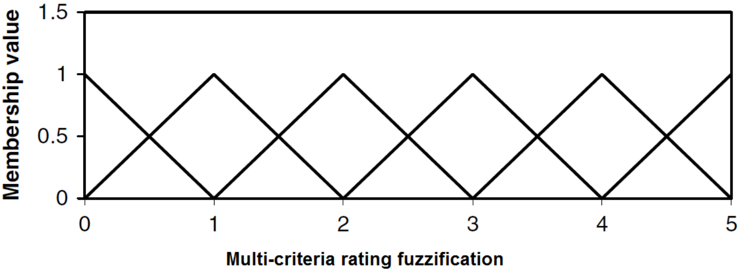


Figure 2. Fuzzy sets for multi-criteria rating side information

The 'user age' feature from MovieLens dataset is fuzzified into three fuzzy sets namely, young, middle-aged, and old [6], as shown in Figure 3, the membership functions of these fuzzy sets are shown below.

$$Q_{young}(m) = \begin{cases} 1, & m \leq 20, \\ (35 - m)/15, & 20 < m \leq 35, \\ 0, & m > 35, \end{cases} \tag{7a}$$

$$Q_{middle}(m) = \begin{cases} 0, & m \leq 20, m > 60, \\ (m - 20)/15, & 20 < m \leq 35, \\ 1, & 35 < m \leq 45, \\ (60 - m)/15, & 45 < m \leq 60, \end{cases} \tag{7b}$$

$$Q_{Old}(m) = \begin{cases} 0, & m \leq 45, \\ (m - 45)/15, & 45 < m \leq 60, \\ 1, & m > 60. \end{cases} \tag{7c}$$

Fuzzy distance function: After side information fuzzification process, to compute the distances between fuzzified features, we replace the Euclidean distance method in Algorithm 1 mentioned above in step *2(i)* with the following modified Euclidean fuzzy distance formula [17].

$$Gfd(X, Y) = \sqrt{\sum_{i=1}^{z} (Lfd(x_i, y_i))^2} \tag{8}$$

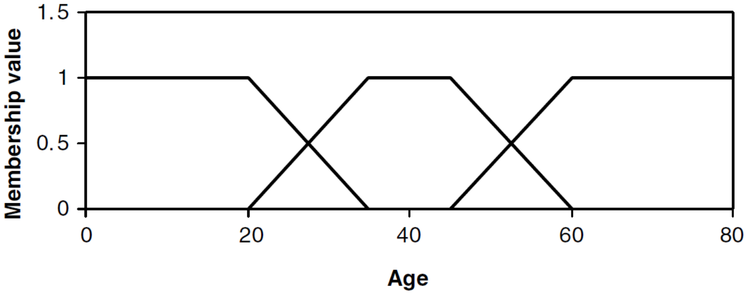


Figure 3. Fuzzy sets for age side information

where *Gfd* is the global fuzzy distance, z is the length of fuzzified vector and *Lfd* is the local fuzzy distance of users:

$$Lfd = dis(x_i, y_i) \times d(x_i, y_i). \tag{9}$$

Here, $d(x_i, y_i)$ computes the difference between vectors x and y of size m, and $dis(x_i, y_i)$ is an Euclidean distance function given shown below.

$$dis(x_j, y_j) = \sqrt{\sum_{i=1}^{m} (x_{i,j} - y_{i,j})^2} \tag{10}$$

where $x_{i,j}$ denotes the membership value of the i^{th} feature in the j^{th} fuzzy set.

3.2 Similarity Computation and Neighbors Selection

After creating groups, to generate most similar top-N neighbors of the target user, the next step is to compute similarities among different users within their respective clusters. The logic behind similarity computation is to extract those users who have provided similar ratings to the same products. As Mahalanobis distance method persists multiple benefits compared to the classical similarity methods, we choose this method to compute the distance between two users who have co-rated a product with a rating. Such as, the individual item's rating does not affect the distance as it only depends upon the variance and covariance of the total ratings given by user. Also, in multi-dimensional space, the MD works well by removing the scaling as well as the collinearity impact of the variables and then calculates the simple Euclidean distance between users. Moreover, there are very few research works published in the field of recommender systems which incorporates Mahalanobis distance for recommendations [15, 34]. The Mahalanobis distance between users can be calculated with the help of following equations.

Let us assume there are two users x and y and our task is to compute the distance between them, in such case, Equation (4) for two different users can be rewritten as [14]:

$$[(x-\bar{x})(y-\bar{y})]vc_z^{-1} = \begin{bmatrix} \frac{\sigma_2^2(x-\bar{x})-(y-\bar{y})\rho_{12}\sigma_1\sigma_2}{det(vc_z)} & \frac{\sigma_1^2(y-\bar{y})-(x-\bar{x})\rho_{12}\sigma_1\sigma_2}{det(vc_z)} \end{bmatrix}. \tag{11}$$

By multiplying $\begin{bmatrix}(x-\bar{x})\\(y-\bar{y})\end{bmatrix}$ in both side of the above Equation (11), we will get

$$\begin{aligned}[(x-\bar{x})(y-\bar{y})]vc_z^{-1}\begin{bmatrix}(x-\bar{x})\\(y-\bar{y})\end{bmatrix} &= \frac{\sigma_2^2(x-\bar{x})^2-(y-\bar{y})(x-\bar{x})\rho_{12}\sigma_1\sigma_2}{det(vc_z)}\\ &\quad+\frac{\sigma_1^2(y-\bar{y})^2-(x-\bar{x})(y-\bar{y})\rho_{12}\sigma_1\sigma_2}{det(vc_z)}\end{aligned} \tag{12}$$

$$= \frac{\sigma_2^2(x-\bar{x})^2(1-\rho_{12}^2)+\sigma_1^2(y-\bar{y}^2)-2(x-\bar{x})(y-\bar{y})\rho_{12}\sigma_1\sigma_2+\sigma_2^2(x-\bar{x})\rho_{12}^2}{\sigma_1^2\sigma_2^2(1-\rho_{12}^2)} \tag{13}$$

$$= \frac{(x-\bar{x})^2}{\sigma_1^2}+\frac{(y-\bar{y})^2}{\sigma_2^2(1-\rho_{12}^2)}-2\frac{(x-\bar{x})(y-\bar{y})\rho_{12}}{\sigma_1\sigma_2(1-\rho_{12}^2)}+\frac{\rho_{12}^2(x-\bar{x})^2}{\sigma_1^2(1-\rho_{12}^2)} \tag{14}$$

$$= \frac{(x-\bar{x})^2}{\sigma_1^2}+\left[\frac{(y-\bar{y})}{\sigma_2\sqrt{1-\rho_{12}^2}}-\frac{\rho_{12}(x-\bar{x})}{\sigma_1\sqrt{1-\rho_{12}^2}}\right]^2. \tag{15}$$

After comparing Equation (15) with Equation (4), the Mahalanobis distance for users x and y will be

$$MD = \sqrt{\left(\frac{x-\bar{x}}{\sigma_1}\right)^2+\left[\left\{\left(\frac{y-\bar{y}}{\sigma_2}\right)-\rho_{12}\left(\frac{x-\bar{x}}{\sigma_1}\right)\right\}\frac{1}{\sqrt{1-\rho_{12}^2}}\right]^2}. \tag{16}$$

The subtraction portion of the above formula is used to correct the correlation between the data. This equation will become a simple Euclidean distance method in case of uncorrelated variables. The above equation is limited to compute similarities between two users who have co-rated a single movie only. Therefore, we updated the formula so that it can calculate distances between multiple users and multiple products ($|items| \geq 1$).

$$MD(x,y) = \frac{\sum_{n\in N_{xy}}\sqrt{\left(\frac{r_{x,n}-\bar{x}}{\sigma_1}\right)^2+\left[\left\{\left(\frac{r_{y,n}-\bar{y}}{\sigma_2}\right)-\rho_{12}\left(\frac{r_{x,n}-\bar{x}}{\sigma_1}\right)\right\}\frac{1}{\sqrt{1-\rho_{12}^2}}\right]^2}}{|N_{xy}|} \tag{17}$$

where N_{xy} is the set of co-rated products by both the users x and y. Rating of user x on product n is represented by $r_{x,n}$ whereas $\bar{x}$ represents the mean of the ratings given by the user x to all products. Finally, after similarity computation between users within the cluster, the top-N most similar users are selected for neighborhood formation.

3.3 Prediction and Recommendations

The collective ratings given by the users of the neighborhood set is used to predict the ratings of all unseen products for the target user [3, 6]. This method is used for all clusters to predict each user's unseen ratings. Finally, top predicted products can be recommended to the target user.

$$pre_{x,i} = \bar{r}_x + nf \sum_{x' \in B} dis(x, x') \times (r_{x',i} - \bar{r}_{x'}). \tag{18}$$

Here, $dis(x, x')$ is the distance between target user x and a neighborhood set member x', B is the neighborhood set of those users who have experienced product i earlier. The multiplier nf, a normalizing factor, is computed as:

$$nf = \frac{1}{\sum_{x' \in B} |dis(x, x')|}.$$

4 EXPERIMENTAL EVALUATION

4.1 Experiment Datasets

MovieLens 100 K dataset contains 943 users' ratings for 1 682 movies gathered by the GroupLens research laboratory at the University of Minnesota. The dataset consists of 100 000 ratings where each user has rated at least 20 movies. The ratings follow the 1-bad, 2-average, 3-good, 4-very good, and 5-excellent numerical scale. This dataset also contains the user-product background information like age, occupation, genre, etc. The sparsity level of the dataset is 93.69 %. Another dataset that we use is the Yahoo! Movies dataset, which contains 62156 ratings rated by 6 078 users on 780 movies. The dataset includes user ratings, movie criteria ratings, total number of movies rated by a user, and corresponding index of the movie which is rated. Additionally, each movie is associated with four different criteria namely story, acting, direction, and visuals, for which users have provided their ratings individually. This dataset follows rating scale from 1-bad to 13-excellent. Since MovieLens supports rating scale from 1-min to 5-max, therefore, we opt to normalize Yahoo! ratings in the same range to get similar range results. For normalization, we formed five rating groups $[\mathrm{Poor}(1,2,3), \mathrm{Fair}(4,5), \mathrm{Average}(6,7,8), \mathrm{Good}(9,10), \mathrm{Excellent}(11,12,13)]$ of these 1–13 ratings, such that, the average of each group $\{2, 4.5, 7, 9.5, 12\}$ have equal step size (i.e., 2.5 in our case). The sparsity level for this dataset is 98.69 %.

4.2 Experimental Settings

From the MovieLens dataset, we selected only those users who have rated at least 60 movies and discarded movies with zero ratings: 497 users and 1 682 movies satisfied this condition and contributed 84,596 ratings out of 100 000. Similarly, from Yahoo! Movies dataset, we extracted those users who have rated at least 20 movies. Where 484 users and 945 movies satisfied this condition and contributed 19 050 ratings out of 62 156. Furthermore, we divided each user's ratings randomly into training set and testing set in the percentage ratio of 70 % and 30 % respectively. The ratings in the training set are used for building the model and neighborhood set generation whereas the testing rating set is marked as unseen products of the target user. After calculating the similarity or distances among users' effectively, we selected top-30 users for the neighborhood set formation. The size of the neighborhood set is chosen experimentally. For k-means algorithm, we choose four cluster centers randomly as an initial point and repeated the cluster assignment process for 30 repetitions or until similar cluster appeared for three consecutive times. We have conducted multiple experiments on the following recommendation methods to demonstrate the effectiveness of the proposed scheme.

- Each User method: In this approach, each user's similarity is computed with every other user in the system. Therefore, we called this approach as the Each User method also known as the non-clustering method.
- Clustering with Side Information approach (Clust-SI): To build a cluster, the first thing to do is to identify the clustering parameters through which we can form similar user clusters. After multiple experiments, we choose 'User Age' feature as a clustering parameter from MovieLens and multi-criteria ratings from Yahoo! Movies datasets. We termed these selected parameters as the side information.
- Clustering with Fuzzy Side Information approach (Clust-FSI): This is the extended version of the Clust-SI approach where we apply fuzzy sets on the side information to deal with the uncertainty issue associated with them and obtain as close as possible neighborhood set for the target user.

4.3 Evaluation Matrices

In this paper, mean absolute error (MAE), root mean square error (RMSE), and coverage of the system evaluation matrices are used for evaluating the performance of the experimental methods. The motivation behind choosing these performance measures is their simplicity and vast use for measuring the effectiveness of RSs [3, 6, 17, 19]. The MAE calculates the average of the absolute differences between actual ($r_{k,j}$) and predicted user ratings ($pre_{k,j}$). The following formula gives the $MAE(k)$ for target user x_k:

$$MAE(k) = \frac{1}{p_k} \sum_{j=1}^{p_k} |pre_{k,j} - r_{k,j}| \tag{19}$$

where p_k is the cardinality of the test ratings set of user x_k. Whereas, RMSE method squares the error, therefore, its value grows faster than MAE when there is a big gap between actual and predicted values. Lower the MAE and RMSE infers more accurate predictions given by the system.

$$RMSE(k) = \sqrt{\frac{\sum_{j=1}^{p_k} (pre_{k,j} - r_{k,j})^2}{p_k}}. \tag{20}$$

The third evaluation metric which we used gives a total coverage of the system. Coverage evaluates a system by measuring the percentage of products for which a recommender system can provide predictions. Usually, a RSs may not be efficient to make predictions for every product in the system. Higher the coverage corresponds to better the prediction ability of the system.

$$Cov = \frac{\sum_{t=1}^{P_k} f_t}{\sum_{t=1}^{P_k} p_t} \tag{21}$$

where f_t is the total number of predicted products for target user x_t.

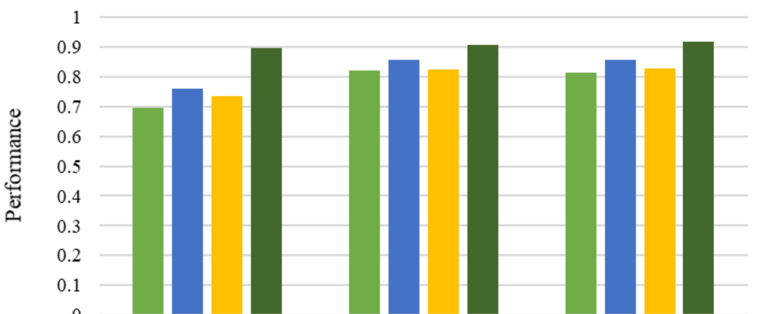
4.4 Experimental Results and Analysis

This section presents the results of the experiments conducted on two real datasets to the improvement of collaborative filtering recommender's performance. The goal of this experiment is to compare the results of the proposed method with other state-of-the-art CF methods under multiple evaluation metrics.

4.4.1 Comparing the Each-User and Clust-SI Methods

In this experiment, we run the Clust-SI approach and compare its results with the traditional Each-User approach. For Each-User, experiments are run for entire training users' database for each step. This type of computation is too time-consuming and sometimes it leads to overfitting problem. In our experiments, we follow the user partitioning technique to shorten the user space. We applied the k-means clustering algorithm to form user clusters by choosing side information as a clustering parameter. Further, the neighborhood set for users is obtained from their respective user clusters only. This agrees with our thinking of reducing the user search space and may reduce the chances of overfitting. Another benefit of the use of side information clustering is that it will alleviate the multi-dimensionality issue because we are not building a complex model by selecting the side information as a third dimension of the system. We picked the user age feature from the MovieLens dataset as the clustering parameter where users are grouped into multiple clusters using k-means clustering. Whereas, for Yahoo! Movies dataset, we selected the multi-criteria movie ratings as the clustering parameter where users with high (low) similar interests are grouped into same (different) groups.

Results summarized in following figures and tables show that Clust-SI outperforms Each-User for all the performance measures. Results of Clust-SI are much better than Each-User for every corresponding similarity approach. The comparing approaches are labelled with four letters (e.g., MLCS or YMCS), the first two letters represent the name of the dataset (i.e. ML for MovieLens and YM for Yahoo! Movies) and the remaining letters represent the name of the measure used (i.e. CS for Cosine-based similarity). The MAE and RMSE of Clust-SI are always smaller than the corresponding approaches of Each-User, as shown in Tables 1 and 2 for both MovieLens and Yahoo! Movies datasets, respectively. Whereas, the coverage is higher for all the comparing approaches.



	Each User	Clust-SI	Clust-FSI
MLCS	0.6954	0.8199	0.8125
MLPC	0.7607	0.8574	0.8548
MLJC	0.7354	0.8257	0.8275
MLMD	0.8946	0.906	0.9173

Figure 4. Comparison of coverage for different collaborative recommenders on ML dataset

4.4.2 Comparing the Clust-SI and Clust-FSI Methods

In this experiment, fuzzy logic is used to handle the uncertainty issue associated with the side information. The multi-criteria ratings and user age are fuzzified with the help of fuzzy function shown in Figures 4 and 5, respectively. The purpose of using fuzzy logic for side information is to get as close as possible to the set of users for the target user. Where a young user will be matched with other young users instead of the old users. It will generate more effective neighborhood set for the target user in comparison to the non-fuzzified methods. Since we clustered users based on their fuzzified side information, we call this approach as Clust-FSI

MAE	Non-Clustering	Clustering Based Approach	
	Each User	*Clust-SI*	*Clust-FSI*
MLCS	0.8498	0.8163	**0.8151**
MLPC	0.8544	**0.8196**	0.8199
MLJC	0.8304	**0.8143**	0.8146
MLMD	0.8142	0.8139	**0.8133**

a) Comparison of MAE for different collaborative recommenders

RMSE	Non-Clustering	Clustering Based Approach	
	Each User	*Clust-SI*	*Clust-FSI*
MLCS	1.0632	**1.0241**	1.0243
MLPC	1.07	**1.0286**	1.0325
MLJC	1.0413	**1.0215**	1.0237
MLMD	1.0194	1.0186	**1.0161**

b) Comparison of RMSE for different collaborative recommenders

Table 1. Performance on MovieLens Dataset

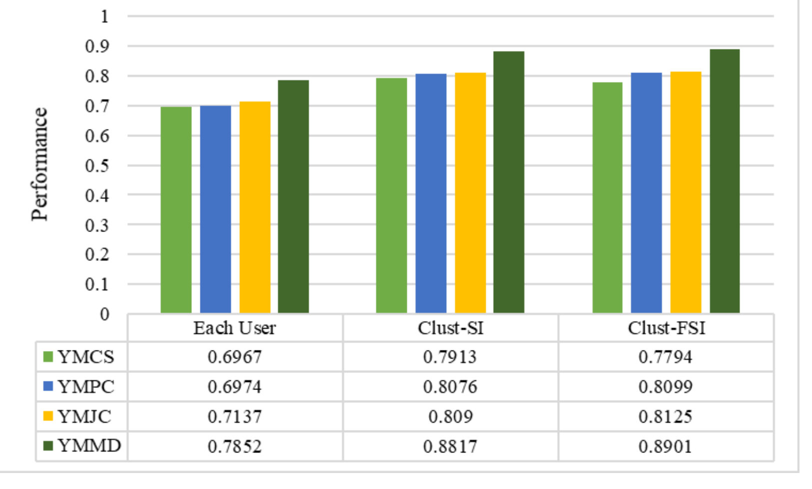


Figure 5. Comparison of coverage for different collaborative recommenders on YM dataset

approach. The experimental results of this approach are compared with Each-User and Clust-SI approaches, as shown in above figures and tables. Results summarized in Tables 1 and 2 show that Clust-SI and Clust-FSI outperform Each-User method for both datasets. The MAE and RMSE for Each-User method are always higher than the clustering based approaches, therefore, we can infer that the use of the side information clustering has improved the performance of the system. Similarly, Figures 4 and 5 prove that the coverage of the clustering based approaches are higher than the non-clustering method where higher coverage shows the higher accuracy

MAE	Non-Clustering	Clustering based approach	
	Each user	*Clust-SI*	*Clust-FSI*
YMCS	0.9694	**0.89908**	0.9006
YMPC	0.9586	0.8938	**0.8833**
YMJC	0.9459	0.8957	**0.8803**
YMMD	0.8871	0.8822	**0.8675**

a) Comparison of MAE for different collaborative recommenders

RMSE	Non-Clustering	Clustering based approach	
	Each user	*Clust-SI*	*Clust-FSI*
YMCS	1.1923	1.109	**1.0207**
YMPC	1.1772	1.1094	**1.1006**
YMJC	1.1664	1.0957	**1.0866**
YMMD	1.0904	1.0839	**1.0735**

b) Comparison of RMSE for different collaborative recommenders

Table 2. Performance on Yahoo! Movies Dataset

of the system. Since non-clustering approach fails, as compared to the clustering based approaches, now we will analyze the performance of the clustering based approaches.

We compared the Clust-SI and Clust-FSI approaches based on four different similarity methods (CS, PC, JS, and MD) concerning MAE, RMSE, and Coverage of the system using two different datasets. Table 1 shows that the MAE and RMSE of MLPC and MLJC methods of Clust-SI are slightly better than the Clust-FSI on MovieLens dataset whereas the methods for Clust-FSI outperform the Clust-SI methods except the MAE of YMCS method for Yahoo! Movies dataset, as shown in Table 2. From Figures 4 and 5 we can see that the coverage of MLCS, MLPC and YMCS for Clust-SI approach is more accurate than the Clust-FSI methods. It means that these methods have greater ability to predict ratings as compared to the fuzzy-based clustering methods.

From the above-shown results we can infer that the clustering based technique has always improved the performance in comparison to the non-clustering methods. Furthermore, the Clust-SI approach shows more accurate results than the Clust-FSI approach for MovieLens dataset. In most of the cases, the Clust-FSI techniques outperform the non-fuzzified clustering approach for the Yahoo! Movies dataset. From these observations we can say that our proposed method can work more efficiently in case of sparse data (Yahoo! Movies dataset is sparser than MovieLens dataset). Moreover, the results prove that the Mahalanobis distance-based measure remains the best-performing method throughout the experiments for all performance measures on both datasets.

5 CONCLUSION

In this paper, we introduced a recommender system framework based on the side information clustering. We focus on the data sparsity and dimensionality issues using Mahalanobis distance and k-means clustering through

1. user's age and
2. multi-criteria movie ratings.

Our approach is based on the assumption that each user has a different opinion on different features. Therefore, to distinguish users, the prime concern of this work is to identify user segments with similar tastes. Fuzzy sets for side information have been applied to choose more accurate and reliable neighbors for the target user in each cluster. Moreover, we know that the traditional Cosine similarity and Pearson Correlation Coefficient methods have a shortage that they work only for commonly rated products and are not suitable for capturing user preferences at ground level. To overcome this problem, we used the well-known Mahalanobis distance method which considers user preferences from local to global level through the variance-covariance matrix. In experiments, we evaluated the effectiveness of our proposed algorithm on accuracy and recommendation performance improvement. The experimental results on two benchmark datasets with different side information demonstrated that our proposed fuzzy-based algorithm for Mahalanobis distance method (Clust-FSI − MD) has better performance compared with the classical recommendation algorithms. Additionally, our proposed framework can be seen as a cross-domain recommendation model because this compact model can be applied to different domains with different clustering parameters. For instance, the model which is used for multi-criteria movie recommendation can also be applied to context-aware music recommendation system. Therefore, in future work, we will try to apply our proposed framework to other domains where multi-dimensionality is the major issue.

REFERENCES

[1] Bobadilla, J.—Ortega, F.—Hernando, A.—Gutiérrez, A.: Recommender Systems Survey. Knowledge-Based Systems, Vol. 46, 2013, pp. 109–132, doi: 10.1016/j.knosys.2013.03.012.

[2] Su, X.—Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, Vol. 2009, 2009, Art. No. 421425, 19 pp., doi: 10.1155/2009/421425.

[3] Wasid, M.—Kant, V.—Ali, R.: Frequency-Based Similarity Measure for Context-Aware Recommender Systems. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016, pp. 627–632, doi: 10.1109/icacci.2016.7732116.

[4] Adomavicius, G.—Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, 2005, No. 6, pp. 734–749, doi: 10.1109/tkde.2005.99.

[5] Domingues, M. A.—Jorge, A. M.—Soares, C.: Dimensions as Virtual Products: Improving the Predictive Ability of Top-N Recommender Systems. Information Processing and Management, Vol. 49, 2013, No. 3, pp. 698–720, doi: 10.1016/j.ipm.2012.07.009.

[6] Wasid, M.—Kant, V.: A Particle Swarm Approach to Collaborative Filtering Based Recommender Systems Through Fuzzy Features. Procedia Computer Science, Vol. 54, 2015, pp. 440–448, doi: 10.1016/j.procs.2015.06.051.

[7] Sahoo, N.—Krishnan, R.—Duncan, G.—Callan, J.: Research Note – The Halo Effect in Multicomponent Ratings and Its Implications for Recommender Systems: The Case of Yahoo! Movies. Information Systems Research, Vol. 23, 2012, No. 1, pp. 231–246, doi: 10.1287/isre.1100.0336.

[8] Li, Q.—Wang, C.—Geng, G.: Improving Personalized Services in Mobile Commerce by a Novel Multicriteria Rating Approach. Proceedings of the 17th International Conference on World Wide Web (WWW '08), 2008, pp. 1235–1236, doi: 10.1145/1367497.1367743.

[9] Liu, X.: An Improved Clustering-Based Collaborative Filtering Recommendation Algorithm. Cluster Computing, Vol. 20, 2017, No. 2, pp. 1281–1288, doi: 10.1007/s10586-017-0807-6.

[10] Sarwar, B. M.—Karypis, G.—Konstan, J.—Riedl, J.: Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT), Vol. 1, 2002.

[11] Liu, L.—Mehandjiev, N.—Xu, D. L.: Multi-Criteria Service Recommendation Based on User Criteria Preferences. Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, 2011, pp. 77–84, doi: 10.1145/2043932.2043950.

[12] Nilashi, M.—Jannach, D.—bin Ibrahim, O.—Ithnin, N.: Clustering- and Regression-Based Multi-Criteria Collaborative Filtering with Incremental Updates. Information Sciences, Vol. 293, 2015, pp. 235–250, doi: 10.1016/j.ins.2014.09.012.

[13] Nilashi, M.—bin Ibrahim, O.—Ithnin, N.—Sarmin, N. H.: A Multi-Criteria Collaborative Filtering Recommender System for the Tourism Domain Using Expectation Maximization (EM) and PCA–ANFIS. Electronic Commerce Research and Applications, Vol. 14, 2015, No. 6, pp. 542–562, doi: 10.1016/j.elerap.2015.08.004.

[14] De Maesschalck, R.—Jouan-Rimbaud, D.—Massart, D. L.: The Mahalanobis Distance. Chemometrics and Intelligent Laboratory Systems, Vol. 50, 2000, No. 1, pp. 1–18, doi: 10.1016/s0169-7439(99)00047-7.

[15] Komkhao, M.—Lu, J.—Li, Z.—Halang, W. A.: Incremental Collaborative Filtering Based on Mahalanobis Distance and Fuzzy Membership for Recommender Systems. International Journal of General Systems, Vol. 42, 2013, No. 1, pp. 41–66, doi: 10.1080/03081079.2012.710437.

[16] Frémal, S.—Lecron, F.: Weighting Strategies for a Recommender System Using Item Clustering Based on Genres. Expert Systems with Applications, Vol. 77, 2017, pp. 105–113, doi: 10.1016/j.eswa.2017.01.031.

[17] Al-Shamri, M. Y. H.—Bharadwaj, K. K.: Fuzzy-Genetic Approach to Recommender Systems Based on a Novel Hybrid User Model. Expert Systems with Applications, Vol. 35, 2008, No. 3, pp. 1386–1399, doi: 10.1016/j.eswa.2007.08.016.

[18] Shi, Y.—Lin, H.—Li, Y.: Context-Aware Recommender Systems Based on Item-Grain Context Clustering. In: Peng, W., Alahakoon, D., Li, X. (Eds.): AI 2017: Advances in Artificial Intelligence. Springer, Cham, Lecture Notes in Computer Science, Vol. 10400, 2017, pp. 3–13, doi: 10.1007/978-3-319-63004-5_1.

[19] Wasid, M.—Ali, R.—Kant, V.: Particle Swarm Optimisation-Based Contextual Recommender Systems. International Journal of Swarm Intelligence, Vol. 3, 2017, No. 2-3, pp. 170–191, doi: 10.1504/IJSI.2017.087874.

[20] Zhang, J.—Lin, Y.—Lin, M.—Liu, J.: An Effective Collaborative Filtering Algorithm Based on User Preference Clustering. Applied Intelligence, Vol. 45, 2016, No. 2, pp. 230–240, doi: 10.1007/s10489-015-0756-9.

[21] Ungar, L. H.—Foster, D. P.: Clustering Methods for Collaborative Filtering. In AAAI Workshop on Recommendation Systems, Vol. 1, 1998, pp. 114–129.

[22] Tsai, C.-F.—Hung, C.: Cluster Ensembles in Collaborative Filtering Recommendation. Applied Soft Computing, Vol. 12, 2012, No. 4, pp. 1417–1425, doi: 10.1016/j.asoc.2011.11.016.

[23] Wei, S.—Ye, N.—Zhang, S.—Huang, X.—Zhu, J.: Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity. 2012 Fifth International Conference on Business Intelligence and Financial Engineering (BIFE), 2012, pp. 69–72, doi: 10.1109/bife.2012.23.

[24] Nilashi, M.—Ibrahim, O.—Bagherifard, K.: A Recommender System Based on Collaborative Filtering Using Ontology and Dimensionality Reduction Techniques. Expert Systems with Applications, Vol. 92, 2018, pp. 507–520, doi: 10.1016/j.eswa.2017.09.058.

[25] Son, N. T.—Dat, D. H.—Trung, N. Q.—Anh, B. N.: Combination of Dimensionality Reduction and User Clustering for Collaborative-Filtering. Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence (CSAI 2017), ACM, 2017, pp. 125–130, doi: 10.1145/3168390.3168405.

[26] Nilashi, M.—Bin Ibrahim, O.—Ithnin, N.—Zakaria, R.: A Multi-Criteria Recommendation System Using Dimensionality Reduction and Neuro-Fuzzy Techniques. Soft Computing, Vol. 19, 2015, No. 11, pp. 3173–3207, doi: 10.1007/s00500-014-1475-6.

[27] Xu, S.—Watada, J.: A Method for Hybrid Personalized Recommender Based on Clustering of Fuzzy User Profiles. 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2014, pp. 2171–2177, doi: 10.1109/fuzz-ieee.2014.6891690.

[28] Wang, Z.—Yu, N.—Wang, J.: User Attributes Clustering-Based Collaborative Filtering Recommendation Algorithm and Its Parallelization on Spark. In: Zhang, L., Song, X., Wu, Y. (Eds.): Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems (AsiaSim 2016, SCS AutumnSim 2016). Springer,

Singapore, Communications in Computer and Information Science, Vol. 643, 2016, doi: 10.1007/978-981-10-2663-8_46.

[29] Kużelewska, U.—Guziejko, E.: A Recommender System Based on Content Clustering Used to Propose Forum Articles. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (Eds.): Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 286, 2014, pp. 285–292, doi: 10.1007/978-3-319-07013-1_27.

[30] Boratto, L.—Carta, S.: Using Collaborative Filtering to Overcome the Curse of Dimensionality When Clustering Users in a Group Recommender System. Proceedings of the 16th International Conference on Enterprise Information Systems (ICEIS 2014), Vol. 2, 2014, pp. 564–572, doi: 10.5220/0004865005640572.

[31] Koohi, H.—Kiani, K.: A New Method to Find Neighbor Users That Improves the Performance of Collaborative Filtering. Expert Systems with Applications, Vol. 83, 2017, pp. 30–39, doi: 10.1016/j.eswa.2017.04.027.

[32] Najafabadi, M. K.—Mahrin, M. N.—Chuprat, S.—Sarkan, H. M.: Improving the Accuracy of Collaborative Filtering Recommendations Using Clustering and Association Rules Mining on Implicit Data. Computers in Human Behavior, Vol. 67, 2017, pp. 113–128, doi: 10.1016/j.chb.2016.11.010.

[33] Resnick, P.—Iacovou, N.—Suchak, M.—Bergstrom, P.—Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM, 1994, pp. 175–186, doi: 10.1145/192844.192905.

[34] Wasid, M.—Ali, R.: An Improved Recommender System Based on Multi-Criteria Clustering Approach. Procedia Computer Science, Vol. 131, 2018, pp. 93–101, doi: 10.1016/j.procs.2018.04.190.

[35] Mahalanobis, P. C.: On the Generalized Distance in Statistics. Proceedings of the National Institute of Sciences of India, Vol. 2, 1936, No. 1, pp. 49–55.

Mohammed Wasid received his B.Tech. from Rajasthan Technical University and M.Tech. from The LNM Institute of Information Technology, Jaipur in 2012 and 2015, respectively. He is working toward the Ph.D. degree from Aligarh Muslim University, Aligarh. Currently he serves as Assistant Professor at the Department of Computer Science and Engineering, Government Engineering College, Bharatpur. His main research interests include recommender systems, evolutionary algorithms, and machine learning.

Rashid Ali received his B.Tech. and M.Tech. degrees from the AMU Aligarh, India in 1999 and 2001, respectively. He received his Ph.D. in computer engineering in 2010 from the AMU Aligarh. He has authored more than 100 papers in various international journals and international conference proceedings. He has presented papers in many international conferences and has also chaired sessions in few international conferences. He has reviewed articles for some of the reputed international journals and international conference proceedings. His research interests include web-searching, web-mining, recommender systems, online social network analysis, and soft computing techniques.