Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 131 (2018) 93–101

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology (ICICT- 2018)

An Improved Recommender System based on Multi-criteria Clustering Approach

Mohammed Wasid*, Rashid Ali

Department of Computer Engineering, Aligarh Muslim University, Aligarh -202001, India

Abstract

Traditional collaborative filtering based recommender systems deal with the two-dimensional user-item rating matrix where users have rated a set of items into the system. Although traditional recommender systems are widely adopted but they are unable to generate effective recommendations in case of multi-dimensionality i.e. multi-criteria ratings, contextual information, side information etc. The curse of dimensionality is the major issue in the recommendation systems. Therefore, in this paper, we proposed a clustering approach to incorporate multi-criteria ratings into traditional recommender systems effectively. Furthermore, we compute the intra-cluster user similarities using a Mahalanobis distance method in order to make more accurate recommendations and compared the proposed approach with the traditional collaborative filtering based method using Yahoo! Movies dataset.

Keywords: Recommender systems; Mahalanobis distance; K-Means clustering; Multi-criteria recommender system

1. Introduction

Recommender system is an information filtering software tool which generates suggestions to internet users for the products that are most likely to be preferred by them[1]. Here, 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. RS has been established themselves in various domains including e-commerce and e-business applications where both user and service provider are get benefited from. The service provider can achieve much more benefit from the fast processing of the data (higher the percentage of sales) whereas a user can get benefited through extracting relevant products in the

* Corresponding author. Tel.: +91 8955574555.
E-mail address: erwasid@gmail.com

10.1016/j.procs.2018.04.190

lesser amount of time. Collaborative filtering (CF) is the extensively implemented and popular technology used in both industry and academia due to its simplicity and accurate enough recommendations[2]. Finding similar users (items) for target user (item) is the crucial step in CF technique[3]. Currently, most of the CF similarity measures are based on commonly rated items. Although these CF recommendation methods are widely used but still they are suffering from a number of inadequacies, including data sparsity and prediction accuracy[4].

Although researchers are working towards the improvement in the accuracy of recommender systems using the overall user-item single-criterion ratings[3,6], multi-criteria recommender system (MCRS) allows to represent the preferences of users on several aspects of the items[7]. A sample of the multi-criteria rating data can be shown in Table 1. Where each user-item overall rating have four different criteria's in the subscript. The motive of MCRS is to provide more efficient suggestions to users by using these multiple components of a product. Multi-criteria ratings represent more complex preferences of each user which improves the performance of recommender systems[7,8]. A restaurant may have cleanliness, service, cuisines, and vicinity as four different criteria's whereas story, visual, acting, and direction can be four different criteria's of a movie. There are two main questions one should consider while developing a multi-criteria collaborative recommender, first, how to incorporate multi-criteria ratings into traditional collaborative filtering technique and improve its accuracy? Second, how to compute the similarity between user profiles with multi-criteria ratings? There is a need for selecting an efficient similarity measure to deal with multi-criteria ratings.

Table 1. An example of multi-criteria rating matrix.

	Titanic	Star wars	Inception	Avatar	Matrix
John	$5_{3,4,3,2}$	$3_{2,3,1,2}$	-	$4_{5,4,3,4}$	$2_{2,3,2,1}$
Alice	$1_{2,1,2,1}$	-	$3_{2,3,2,2}$	$4_{5,3,2,4}$	$4_{5,4,2,4}$
Maria	$2_{2,1,2,2}$	$3_{3,2,2,1}$	$4_{5,4,3,4}$	-	$3_{2,3,2,1}$
Bob	-	$4_{5,4,3,3}$	$3_{2,3,1,2}$	$3_{2,3,2,2}$	$3_{1,2,3,2}$

Therefore, we incorporate the multi-criteria ratings into the collaborative recommender system through K-means clustering in order to deal with the multi-dimensionality issue. Where users are clustered using their multi-criteria ratings in order to reduce the search space and computational time while generating neighborhood set of a user. Users with similar criteria preferences fall into the same user group. In this way, the multi-criteria rating is selected to select those users which are most similar to each other. In the second phase, we use Mahalanobis distance to compute the similarity between users in the same cluster. Hence, only those users are considered for the distance computation which are close to the target user and reduces the computational time.

The remainder of the paper is organized as follows: Section 2 introduces the background. The proposed work is discussed in Section 3. In Section 4, we evaluate the proposed method using the Yahoo! Movies dataset. Finally the conclusion is shown in Section 5.

2. Background

2.1. Recommendation techniques

Generally, recommender systems are categorized into three types namely collaborative filtering, content-based, and hybrid technique[4].

2.1.1. Content-based technique (CB)

The key mechanism of CB technique is that users who have shown preferences on some items in the past will have the similar taste on the other similar items in the future[1]. The similarity between items is analyzed based on the 'description' of the items in the CB technique.

2.1.2. Collaborative filtering technique (CF)

Traditional CF recommendation technique works using the analysis of the historical user-item rating data, find the other like-minded users, predict the ratings of the item for the target user and generate the recommendations[17].

2.1.3. Hybrid filtering

The shortcomings of both collaborative filtering and content-based techniques can be overcome by combining them into a single hybrid technique. Hybrid filtering is able to generate recommendations using the strengths of both techniques.

2.2. Similarity measures

In our work, we use following similarity measures to compute the similarity/distances between users in the system.

2.2.1. Correlation-based similarity

Pearson Correlation coefficient (PC) is widely used and popular for measuring similarity of users or items for classical CF technique[3,6,17]. The similarity between two user u and user v is calculated as follows:

$$PC(\mathbf{u},\mathbf{v}) = \frac{\sum_{s \in S_{uv}} (r_{\mathbf{u},s} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},s} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i \in S_{uv}} (r_{\mathbf{u},s} - \bar{r}_{\mathbf{u}})^2} \sqrt{\sum_{i \in S_{uv}} (r_{\mathbf{v},s} - \bar{r}_{\mathbf{v}})^2}}, \tag{1}$$

where $r_{u,s}$ is the rating of item s given by user u and $\bar{r}_u$ is the mean of the total rating given by the user u. S_{uv} is the set of items commonly rated by both user's u and v.

2.2.2. The mahalanobis distance (MD)

This method compute the inverse of the variance-covariance matrix of the dataset and use correlation of the data[14,15]. The variance–covariance matrix vc_m is constructed using following equation:

$$vc_m = \frac{1}{n-1}(M_c)^T (M_c), \tag{2}$$

where M is the user-item matrix having n items in the rows computed for x variables. Whereas, M_c is the column-centered user-item matrix $(M - \bar{M})$. For user u and user v, the variance-covariance matrix is computed as follows:

$$vc_m = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \tag{3}$$

where σ_1^2 and σ_2^2 represents the variances of the values of, respectively, the first and second users. $\rho_{12}\sigma_1\sigma_2$ is the covariance between the two users. The MD for a single user is then computed similar to the concept of Euclidean distance method.

$$MD = \sqrt{(u - \bar{u}) vc_m^{-1} (u - \bar{u})^T}, \tag{4}$$

with,

$$vc_m^{-1} = \begin{bmatrix} \sigma_2^2/det(vc_m) & -\rho_{12}\sigma_1\sigma_2/det(vc_m) \\ -\rho_{12}\sigma_1\sigma_2/det(vc_m) & \sigma_1^2/det(vc_m) \end{bmatrix}, \tag{5}$$

where $det(vc_m)=\sigma_1{}^2\sigma_2{}^2(1-\rho_{12}{}^2)$, is the determinant of the variance–covariance matrix. Since, equation (4) considers only a single user u, we will discuss MD for multiple users in detail in section 3.2.

2.3. Clustering-based collaborative filtering

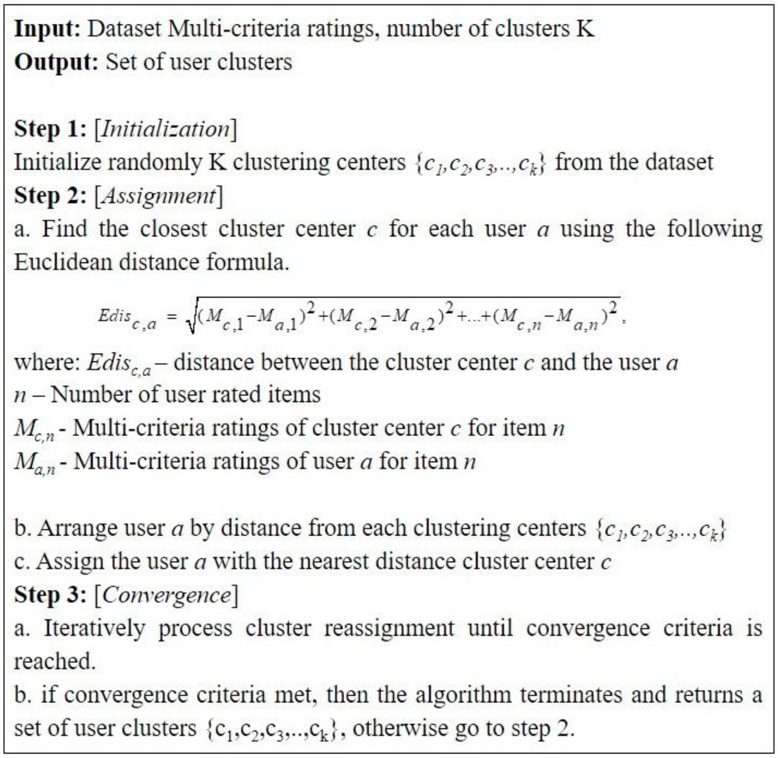
There has been a lot of research work being done in the area of recommender systems using clustering. A user based clustering has been proposed using user-user similarity and resulting clusters are used for neighborhood set generation[10]. Whereas, items are partitioned using clustering algorithm based on rating data[12]. Ungar and Dean[13] clusters both user-item separately using K-means and Gibbs sampling. Here, item clusters are used to re-cluster users based on the number of items in each item cluster they have rated and vice-versa. A novel clustering-based collaborative filtering approach was developed where user groups are formed using a proposed approach to lessen the impact of the data sparsity[11]. After cluster formation, nearest neighbors are found from each user group to produce the accurate recommendation to the user. Similarly, Xiaojun[9] proposed an improved clustering-based collaborative filtering recommender method. Where author has used K-means clustering to cluster the users and then an improved similarity method is developed to generate most similar neighbors in the cluster to the target user. K-means clustering in collaborative filtering recommender has also been used to cope up with scalability and accuracy problems[5].

3. Proposed Approach

In the proposed technique, the actual neighbor of a user is found based on Mahalanobis distance within the user cluster. Unlike traditional similarity measures, the Mahalanobis distance not only considers the commonly rated items between two users but also the variance and co-variance between them which make it more efficient to generate the more similar neighborhood set to the target user, thus, improve the effectiveness of the recommendations[15].

3.1. User cluster creation

In order to generate most relevant user clusters according to their multi-criteria rating preferences, the first step is to extract the user preferences from the existing dataset using the multi-criteria rating they have given to the items in the system. After that, the number of user cluster centers has to be defined. Determining the optimal number of clusters is a tough task in K-means clustering. The next step is to cluster each user with any one of these pre-specified cluster centers based on their distance (similarity). The main problem here is to find out a way to calculate the efficient similarity between the multi-criteria ratings of different users. Usually, determining user groups is performed using some similarity measures such as Pearson correlation, Euclidean distance, or cosine similarity. Users will fall under that cluster which has minimum distance (maximum similarity) from them, so that, users with most common preferences are grouped into the same cluster. This process iterates until convergence criteria is reached. The detailed algorithm for creating multi-criteria user clusters is presented in Figure 1.



Input: Dataset Multi-criteria ratings, 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 the following Euclidean distance formula.

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

where: $Edis_{c,a}$ – distance between the cluster center c and the user a
n – Number of user rated items
$M_{c,n}$ - Multi-criteria ratings of cluster center c for item n
$M_{a,n}$ - Multi-criteria ratings of user a for item n

b. Arrange user a by 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.

Fig. 1. The Algorithm for user cluster formation.

3.2. Neighborhood set generation

In this paper, we use Mahalanobis distance method to compute the distance between user's co-rated item ratings. After creating groups, in order to obtain most similar top-N neighbors to the target user, the next step is to compute similarity among different users in the same cluster. The idea of similarity computation is to extract those users who have provided similar ratings to the similar items. The Mahalanobis distance between users can be computed with the help of following equations. Let's assume, there are two users *u* and *v* and our job is to compute the distance between them, in such case, equation (4) for two different users *u* and *v* can be rewritten as[14],

$$[(u-\bar{u})(v-\bar{v})]vc_m^{-1} = \begin{bmatrix} \frac{\sigma_2^2(u-\bar{u})-(v-\bar{v})\rho_{12}\sigma_1\sigma_2}{det(vc_m)} & \frac{\sigma_1^2(v-\bar{v})-(u-\bar{u})\rho_{12}\sigma_1\sigma_2}{det(vc_m)} \end{bmatrix}, \tag{6}$$

Multiply $\begin{bmatrix} (u-\bar{u}) \\ (v-\bar{v}) \end{bmatrix}$ in both side of the above equation (6), we get:

$$[(u-\bar{u})(v-\bar{v})]vc_m^{-1}\begin{bmatrix} (u-\bar{u}) \\ (v-\bar{v}) \end{bmatrix} = \frac{\sigma_2^2(u-\bar{u})^2-(v-\bar{v})(u-\bar{u})\rho_{12}\sigma_1\sigma_2}{det(vc_m)} + \frac{\sigma_1^2(v-\bar{v})^2-(u-\bar{u})(v-\bar{v})\rho_{12}\sigma_1\sigma_2}{det(vc_m)}, \tag{7}$$

$$= \frac{\sigma_2^2 (u-\bar{u})^2 (1-\rho_{12}^2) + \sigma_1^2 (v-\bar{v})^2 - 2(u-\bar{u})(v-\bar{v})\rho_{12}\sigma_1\sigma_2 + \sigma_2^2 (u-\bar{u})^2 \rho_{12}^2}{\sigma_1^2 \sigma_2^2 (1-\rho_{12}^2)}, \tag{8}$$

$$= \frac{(u-\bar{u})^2}{\sigma_1^2} + \frac{(v-\bar{v})^2}{\sigma_2^2 (1-\rho_{12}^2)} - 2\frac{(u-\bar{u})(v-\bar{v})\rho_{12}}{\sigma_1\sigma_2(1-\rho_{12}^2)} + \frac{\rho_{12}^2 (u-\bar{u})^2}{\sigma_1^2 (1-\rho_{12}^2)}, \tag{9}$$

$$= \frac{(u-\bar{u})^2}{\sigma_1^2} + \left[\frac{(v-\bar{v})^2}{\sigma_2\sqrt{1-\rho_{12}^2}} - \frac{\rho_{12}(u-\bar{u})}{\sigma_1\sqrt{1-\rho_{12}^2}} \right], \tag{10}$$

Compare equation (10) with equation (4), the MD is,

$$MD = \sqrt{\left(\frac{u-\bar{u}}{\sigma_1}\right)^2 + \left[\left\{\left(\frac{v-\bar{v}}{\sigma_2}\right) - \rho_{12}\left(\frac{u-\bar{u}}{\sigma_1}\right)\right\}\frac{1}{\sqrt{1-\rho_{12}^2}}\right]^2}, \tag{11}$$

In the above equation (11), the subtraction part (second part) of the formula is used to correct the correlation between the data. In case two variables are uncorrelated (when ρ_{12}=0) then this equation becomes a method similar to the Euclidean distance measure. Furthermore, equation (11) allows to computes the distance between single common items of two users but in CF recommender a user may share multiple co-rated items with other users. Therefore, we modify the above formula such that it can compute the distance between two users *u* and *v* who share a set of items (|*items*| ≥1) is computed using the following equation (12).

$$MD(u,v) = \frac{\Sigma_{s \varepsilon S_{uv}} \sqrt{\left(\frac{r_{u,s}-\bar{u}}{\sigma_1}\right)^2 + \left[\left\{\left(\frac{r_{v,s}-\bar{v}}{\sigma_2}\right) - \rho_{12}\left(\frac{r_{u,s}-\bar{u}}{\sigma_1}\right)\right\}\frac{1}{\sqrt{1-\rho_{12}^2}}\right]^2}}{|S_{uv}|}, \tag{12}$$

where $r_{u,s}$ is the rating of item *s* given by user *u* and $\bar{u}$ is the mean of the ratings given by the user *u* to all items. S_{uv} is the set of co-rated items of both the users' *u* and *v*. After computing the distance between each user in the cluster, top-N most close users (less distanced) are added in the neighborhood set of the target user.

3.3. Prediction and recommendations

After similarity computation step, in order to predict the rating of an item for the target user, the system estimates the collective ratings given by the members of the target user's neighborhood set of the cluster. The predicted rating, $p_{u,i}$, of item *i* of a user *u* is computed by following equation[6,16].

$$p_{u,i} = \bar{r}_u + N \sum_{u' \in H} dis(u,u') \times (r_{u',i} - \bar{r}_{u'}), \tag{13}$$

where H denotes the neighbors who have rated item i. The multiplier N (normalizing factor) is typically measured as $N = 1/\sum_{u'\in H} | dis(u,u') |$ and $dis(u,u')$ is the similarity/distance between target user u and neighborhood user u'.

4. Experimental Evaluation

4.1. Experimental Settings

Yahoo! Movies dataset consists 62156 ratings provided by 6078 users on 976 movies. For simplicity, we have extracted only those users who have given ratings to at least 20 movies. Where 484 users and 945 movies satisfied this condition and contributed 19050 ratings out of 62156. Furthermore, we divided each user's ratings randomly into training set and testing set in the percentage ratio of 70% and 30% respectively. After calculating the similarity or distances between users successfully we select top-30 most similar users for the neighborhood set formation.

4.2. Evaluation Metric

To evaluate the proposed technique, we used Mean Absolute Error (MAE) performance metric due to its simplicity and accuracy that matches the goal of our experiment. The MAE is widely used metric to evaluate the performance of collaborative recommenders that have been employed by various researchers[3,6,16]. The MAE measures the deviation of predicted and actual user ratings. The MAE is calculated as follows:

$$\mathrm{MAE} = \frac{\sum_{j=1}^{n} | p_j - r_j |}{n}, \tag{14}$$

where n represents the number of ratings, r_j is the actual rating of the item, and p_j is the predicted rating for target user on item j.

4.3. Result and Analysis

Since our dataset contains both single-criteria and multi-criteria user provided ratings, therefore, in this section we will discuss the results obtained from both clustering and non-clustering environments.

Table 2 shows the comparison of our proposed approach with the existing method on Yahoo! Movies dataset for both clustering and non-clustering environments. Table 2 shows the relative performance of existing Pearson collaborative recommender (PCRS) with our proposed Mahalabosis distance recommendation scheme (MDRS). It is clear from the results that MDRS considerable performed better than Pearson based collaborative filtering based method in terms of Mean absolute error of the system for both non-clustering and clustering environments.

Table 2. MAE of non-clustering and clustering collaborative recommenders.

Approach	MAE (Non-Clustering)	MAE (Clustering)
PCRS	2.4577	2.2734
MDRS	2.3094	2.1751

Figure 2 shows the graphical representation of the results obtained from both non-clustering and clustering approaches in Table 2. The Mahalabosis distance-based method MDRS shows the improved result compared to the PCRS method. Where lower the MAE value shows the better the result. Furthermore, from the results, we can clearly observe that the clustering based technique always have better performance than non-clustering approaches.

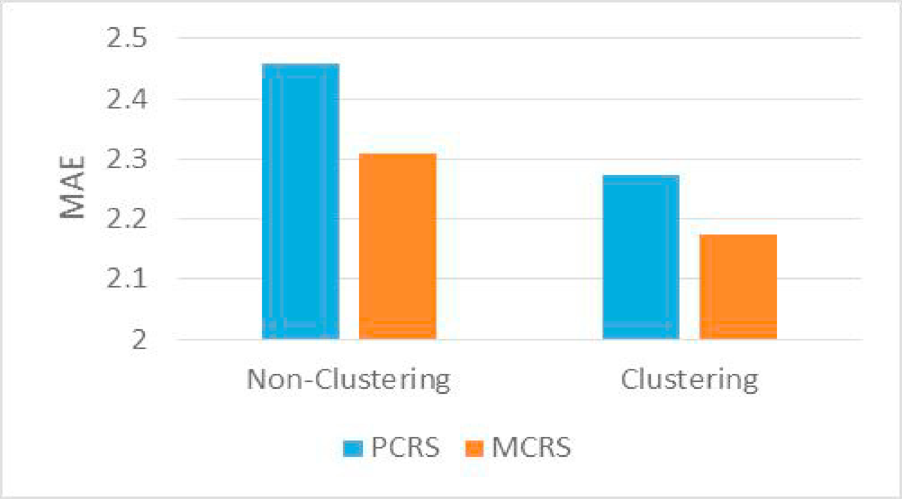


Fig. 2. Comparison of non-clustering with clustering collaborative recommenders.

5. Conclusion

In this paper, we have incorporated the multi-criteria ratings into the traditional collaborative filtering based recommender system using K-means algorithm. Our approach handled the dimensionality problem by treating the third dimension (multi-criteria) as the clustering parameter of the users. Our method is based on an assumption that every user has different opinion on different criteria. Therefore, to distinguish different users, the prime concern of this work is to identify user segments with similar tastes. These user segments are formed to choose more correct and reliable neighborhood set for the target user. We have used the Mahalanobis distance method to generate more accurate neighbors for each user within the cluster. Results demonstrate that our proposed approach is more accurate and effective than the traditional Pearson based collaborative filtering based approach.

References

1. Lu, Jie, et al., “Recommender system application developments: a survey”, *Decision Support Systems*, Vol. 74, 2015, pp. 12-32.
2. Shi, Y., Larson, M., & Hanjalic, A., “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges”, *ACM Computing Surveys (CSUR)*, Vol. 47, no. 1, 2014, pp. 3.
3. Wasid, Mohammed, Vibhor Kant, and Rashid Ali, “Frequency-based similarity measure for context-aware recommender systems”, In *Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2016, pp. 627-632.
4. Adomavicius, Gediminas, and Alexander Tuzhilin, “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, no. 6, 2005, pp. 734-749.
5. Bilge, Alper, and Huseyin Polat, “A comparison of clustering-based privacy-preserving collaborative filtering schemes”, *Applied Soft Computing*, Vol. 13, no. 5, 2013, pp. 2478-2489.
6. Wasid, Mohammed, and Vibhor Kant, “A particle swarm approach to collaborative filtering based recommender systems through fuzzy features”, *Procedia Computer Science*, Vol. 54, 2015, pp. 440-448.
7. Nachiketa Sahoo, Ramayya Krishnan, George Duncan, and Jamie Callan, “Research note-the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! Movies”, *Information Systems Research*, Vol.23, no. 1, 2012, pp. 231–246.
8. Qiudan Li, Chunheng Wang, and Guanggang Geng, “Improving personalized services in mobile commerce by a novel multicriteria rating approach”, In *Proceedings of the 17th ACM conference on World Wide Web*, 2008, pp. 1235–1236.
9. Xiaojun, Liu, “An improved clustering-based collaborative filtering recommendation algorithm”, *Cluster Computing*, 2017, pp. 1-8.
10. Sarwar, Badrul M., George Karypis, Joseph Konstan, and John Riedl, “Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering”, In *Proceedings of the fifth international conference on computer and information technology*, Vol. 1, 2002.
11. Zhang, Jia, Yaojin Lin, Menglei Lin, and Jinghua Liu, “An effective collaborative filtering algorithm based on user preference clustering”, *Applied Intelligence*, Vol**.** 45, no. 2, 2016, pp. 230-240.
12. O’Connor, Mark, and Jon Herlocker, “Clustering items for collaborative filtering”, In *Proceedings of the ACM SIGIR workshop on recommender systems*, UC Berkeley, Vol. 128, 1999.

13. Ungar, Lyle H., and Dean P. Foster, “Clustering methods for collaborative filtering”, In *AAAI workshop on recommendation systems*, Vol. 1, 1998, pp. 114-129.
14. De Maesschalck, Roy, Delphine Jouan-Rimbaud, and Désiré L. Massart, “The mahalanobis distance”, *Chemometrics and intelligent laboratory systems*, Vol. 50, no. 1, 2000, pp. 1-18.
15. Komkhao, Maytiyanin, Jie Lu, Zhong Li, and Wolfgang A. Halang. “Incremental collaborative filtering based on Mahalanobis distance and fuzzy membership for recommender systems”, *International Journal of General Systems*, Vol. 42, no. 1, 2013, pp. 41-66.
16. Wasid, Mohammed, Rashid Ali, and Vibhor Kant, “Particle swarm optimisation-based contextual recommender systems”, *International Journal of Swarm Intelligence,* Vol. 3, no. 2-3, 2017, pp. 170-191.
17. Wasid, Mohammed, and Rashid Ali, “Context Similarity Measurement Based on Genetic Algorithm for Improved Recommendations”, *Applications of Soft Computing for the Web*. Springer, Singapore, 2017, pp. 11-29.