An Evolutionary Scheme for Improving Recommender System Using Clustering

ChemsEddine Berbague[1](✉), Nour El Islem Karabadji[1,2], and Hassina Seridi[1]

[1] Electronic Document Management Laboratory (LabGED),
Badji Mokhtar-Annaba University, BP 12, Annaba, Algeria
{berbague,Karabadji,Seridi}@labged.net
[2] High School of Industrial Technologies, P.O. Box 218, 23000 Annaba, Algeria

Abstract. In user memory based collaborative filtering algorithm, recommendation quality depends strongly on the neighbors selection which is a high computation complexity task in large scale datasets. A common approach to overpass this limitation consists of clustering users into groups of similar profiles and restrict neighbors computation to the cluster that includes the target user. K-means is a popular clustering algorithms used widely for recommendation but initial seeds selection is still a hard complex step. In this paper a new genetic algorithm encoding is proposed as an alternative of k-means clustering. The initialization issue in the classical k-means is targeted by proposing a new formulation of the problem, to reduce the search space complexity affect as well as improving clustering quality. We have evaluated our results using different quality measures. The employed metrics include rating prediction evaluation computed using mean absolute error. Additionally, we employed both of precision and recall measures using different parameters. The obtained results have been compared against baseline techniques which proved a significant enhancement.

1 Introduction

A recommender system RS is an extension of the conventional information retrieval techniques. It plays a major role in e-commerce websites as well as social networks such as Google, Facebook, Netflix and others whereas RS aims to estimate user ratings or to personalize a bag of recommendation for every user based on his preferences made explicitly by bias of his ratings or implicitly during the interaction with the system (browsing sequences). Besides, that contextual information (time, location), as well as social and demographic information, are used to make high-quality recommendations. There are different recommendation algorithms each of which fits a specific type of information and deals with a well-defined limitation; two well-known algorithms are listed in the items below.

One, Content based filtering algorithms denoted CBF [1] uses available information about items (features, attributes) to make recommendations for a target use & here they look for a set of nearest neighbors for the previously seen items; which will be used to estimate user's preferences. CBF is a very beneficial filtering

A. Amine et al. (Eds.): CIIA 2018, IFIP AICT 522, pp. 290–301, 2018.
https://doi.org/10.1007/978-3-319-89743-1_26

algorithm since it can deal with low users overlap by focusing only on user's profile; however it involves a hard content preprocessing step which raises a complicated task especially for media content (images, videos). Besides, for a given user, CBF algorithm tends to recommend a set of very similar items to those already seen in the past, by consequence limiting the diversity of recommendations, this effect is denoted in the literature by the overspecialization problem.

Second, memory based collaborative filtering algorithms denoted CF [2,3] have given good results in terms of accuracy and used widely in many researches [4]; User based CF bases its recommendation on the concept of the collective users trends toward products and items whereas the algorithm predicts the unrated items for a specific user by considering similarities with the rest of users around him, So close users would influence strongly the predicted rating value while far users would have a limited effect; Similarities distance between users may be computed in many ways such as cosine similarity, Pearson correlation, etc. K nearest neighbors (KNN) [2] is the most used algorithm for collaborative filtering, KNN takes on consideration only a limited number of users; where it determines for a given user k nearest neighbors then calculates predictions for the unrated items by aggregating nearest users ratings. Some contributions [3,5] have restricted the neighbors set size to include only users who (1) satisfy a minimum similarity threshold; (2) simultaneously allow making both accurate and diverse recommendations.

For a given user $u \in U$ and item $i \in I$ the estimated rating is calculated using the Eq. 1:

$$Pr(u,i) = \bar{r_u} + \frac{\sum_{j \in N(u)} sim(u,j) \times R(j,i)}{\sum_{j \in N(u)} sim(u,j)} \tag{1}$$

Where U denote the set of all users in the dataset, $|.|$ the size of the set while i_u is the set of users who have rated the item i.

An ideal collaborative filtering engine should cope with scalability problem, and recommendation quality since rating prediction in whole datasets might be a very complicated task and involves long time and high computation complexity which are proportional to the number of users and items in the rating matrix. A common approach to deal with scalability issues is to make a preprocessing step consists of clustering users into groups of similar users and restrict collaborative computing on the scale of the cluster to which a target user belongs. K-Means is a partitioning algorithm used widely in recommendation system to address the scalability issue; the algorithm is known by its suitability for large scale datasets and fast convergence, while it does not guarantee the best results since it usually gets stuck at local optimality solutions. The initialization step plays a crucial role in clustering quality; random based initialization of K-means in the classical version influences strongly the quality of obtained results. K-means clustering involves specifying initially a fixed number of clusters, the algorithm stops after getting a lower error rate according to a predefined threshold or after reaching a number of iteration.

In more details, the algorithm assigns each data vector to the closest center and iteratively looks for the best center inside each newly formed cluster. Clustering results depend on the initially chosen seeds which raises the worst drawback.

Many papers have addressed the initial seeds selection dilemma in K-means algorithm; it exists two considerable classified solutions one based on statistic and another one based on the evolutionary algorithms. The first type scans data samples to look for their statistical features, density and variance are some measurements used to select the appropriate initial seeds; By revenge mainly evolutionary approaches perform a research in the search space of possible solutions by adopting an adequate formulation of the clustering problem and optimizing a clustering quality measure, density, mean square error have been used to guide the optimization.

In this paper, we present a partitioning based genetic algorithm to enhance user based collaborative filtering. This latter' objective is achieved by pulling up an optimal partitioning of users over k groups. This may allow us to deal with scalability problem in recommender systems. The proposed optimization algorithm guided by a clustering quality measure explores possible solutions over the whole ways allowing for partitioning a set of n users into k non-empty subsets. Mainly, our contribution in this work consists in (1) proposing a new genetic individuals encoding that represents possible solutions where each solution is represented as a set of centers and the number of the most similar users around them. (2) designing a multi-objective optimization fitness function to ensure similarity intergroup and diversity between centers. The experimental results on the benchmark Movielens dataset using different parameters and compared with baseline approaches proved a significant enhancement.

The remainder of this paper includes in Sect. 2 related work explaining clustering in recommendation systems and discussing the initial seed selection problem in K-means algorithm, Then in Sect. 3, we cite minutely our proposed clustering approach based completely on a genetic algorithm. In Sect. 4, we present the experimental results on MovieLens dataset compared to K nearest neighbors algorithm and k-means.

2 Related Works

Collaborative filtering (memory based) is a widely used recommendation algorithm that bases its recommendation on selecting a set of neighbor users by measuring similarity between each pair of users, this step is a time consuming and a high computation complexity task, these problems have been treated by proposing distributed collaborative filtering [6], other approaches consist of clustering users into similar groups on which computation is restricted.

Many statistical and bio-inspired clustering algorithms have been proposed in the literature for recommender system. However, K-means is still the most used algorithm due to its simplicity and fast convergence. In contrary this algorithm suffers the initial seeds selection dilemma. A comparative study [7] has taken the different existing approaches of k-means initialization in terms of complexity.

In [8] Zahra et al. have proposed many variants of K-means algorithm, it employs mainly similarity measures as well as density, variance, average to select best initial seeds, in some other extensions authors have made a filtering on users based on their statistical measures to improve their proposed K-means algorithms, the filtering has extracted users with high similarity distance from the remaining data samples or users who have larger rating size, the proposed algorithms in the paper have been applied to many recommendation system data sets and their results have been compared in terms of mean square error and precision measure, however, the proposed algorithms have not dealt with sparsity problem, measuring similarity between two users profiles might reflect inaccurate similarity.

In [9] Cao et al. have proposed an iterative progressive initial seed selection for k means algorithm, their method consists of a random selection of the first seed, however, the remainder $k-1$ seeds are selected in regardless to the actually inserted seeds in the seed set, two selection criteria are employed to ensure the coherence of the seeds: coupling combining indicates the probability of belonging of two samples to the same cluster, cohesion give an index about the breadth of neighborhood of a given sample, the efficiency of the algorithm is highly influenced by the selection of the first seed, selecting inappropriate first seed would affect the rest of seeds.

In [10] Li has proposed a center clustering initialization based on euclidean distance measurement and neighbors selection, the algorithm constitutes a set of pairs each of which contains a data sample and its nearest neighbors found by scanning the dataset and calculating the euclidean distance, the authors have put some initial seed selection rules based on two main assumptions: A data sample and its NN are in the same cluster or the overlap of two clusters; The more two pair of user and his NN are dissimilar the more they belong to different clusters, these algorithms are still can deal only with a limited number of cluster equal 2.

Besides statistically based clustering methods, there are evolutionary approaches [11] which have been used widely for clustering in recommendation systems since the conventional clustering techniques are subject to failing stuck at local optimality solutions, evolutionary clustering techniques have been classified into two categories: Algorithms that require specifying the number of clusters and algorithms that do not require such information.

In [12] Kuo et al. proposed a k-means algorithm based on ant colony, the execution starts by initializing a number of clusters and its correspondents randomly selected centroids, equal pheromone values are laid on each possible path, ants are assigned to random samples, at every iteration pheromone values are updated so ants move from a source to a center respecting a transmission probability based on a statistical based variance value.

In [13] Kalyani and Swarup have proposed a supervised k-means algorithm whereas a PSO is integrated into the clustering process; a set of randomly generated particles represent initial seeds which are fed to a classical k-means algorithm, the quality measure computed by k-means results is used to supervise

PSO behavior. However, supervision process is complex since the quality computation involves executing the k-means algorithm to obtain the clustering quality value.

Globally most proposed evolutionary (EA) based clustering algorithms are combined with a baseline clustering technique, such as k-means. The objective of EA is enhancing the clustering by exploring the search space to find the best initial seeds. Genetic algorithms are an evolutionary approach proposed early in 1989 and used mainly for optimization problems. The algorithm is inspired from the natural selection and survive for the fittest principals. It has been used for recommendation systems in many aspects such as similarity computation [14], clustering [15] and recommendation algorithms hybridization [16]. Most presented works using genetic algorithms have discussed encoding scheme problem, fitness function and genetic operators as well as the population initialization choice. We present in our paper a completely genetic based clustering algorithm that finds the appropriate number of clusters without requiring k-means post clustering.

3 Partitioning Based Genetic Algorithm

On the contrary of traditional algorithms that fail stuck at local optimality, genetic algorithms (GA) are a powerful optimization techniques since it looks for a global optimum solution. GAs apply genetic operations at every iteration to produce completely new different solutions, these operations are applied initially on a first generation generated randomly. In particular, we fixed the initial number of individuals to 20, using a single point crossover operator with a probability of 0.5 as well as employing bit flip mutation operator with a probability of 0.1.

3.1 Encoding and Decoding Phase

This phase is an essential step of our proposed genetic algorithm. Formally, this phase consists of designing a bijective function that allows us representing each solution of whole ways allowing for partitioning a set of n user into k non-empty groups.

Encoding. The encoding step consists in representing solutions as arrays of 0s and 1s (i.e., binary strings). Therefore, we propose a new binary string encoding, each one must store the full information about its corresponding partition solution. Our designed encoding is presented in Fig. 1, where a chromosome is composed of k genes and each gene stores two informations: (a) an integer C_i representing a center; (b) an integer α_i representing the number of must nearest users over the center C_i. This encoding allows representing the whole possible partition configurations. Where k is the number of partition for which we take as parameters two thresholds minimum Min_C and maximum Max_C number of clusters and look for the best users partition over k groups. Each group (i.e.,

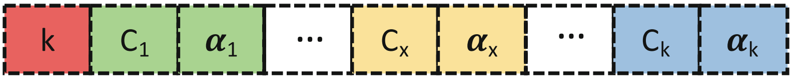

Fig. 1. Chromosome encoding scheme

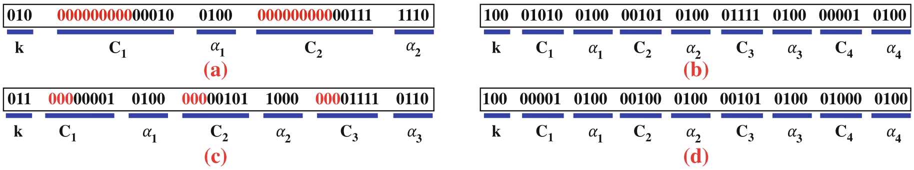

Fig. 2. Chromosome instances

cluster) of the k ones is represented by its center C_i and the number of its members α_i. A C_i storage space is represented by X_C bits which encode a user identifier $[1..|U|]$. Even here we can note that the C_i identifiers must be different. While α_i storage space is represented by X_α bits which encode members number in each group and their sum must be equal to $(|U| - k)$. The chromosome size $|ch|$ is number of bits need to represent in binary: (1) Max_C; (2) $|U|$; (3) $|U|-k$. Thus $|ch| = X_{Max_C} + (Max_C * (X_\alpha + X_\alpha))$. To put it simply we propose to see Example 1.

Example 1. Assume a data with $|U| = 200$. Thus, the minimum Min_C and the maximum Max_C number of clusters are 2 and 10 respectively if the size of the cluster including the center is at least 20. Therefore, to encode all possible configurations, we consider the largest case which requires the greatest storage space. For this instance case we need to encode information over $X_k + Max_C * (X_C + X_\alpha)$ bits= 164 bits. First, to encode the clusters number, 4 bits are required to store a binary representation of integers between 2 and 10. Then, a maximum size value of a group including a center may be 180 for two clusters partition. According to this maximum size value (i.e., 180), the storage space X_α required is 8 bits. While the most value of center identifier may be 200, then the storage space X_C required is 8 bits.

Decoding. The decoding steps allow converting the binary gene codes encoded in a chromosome ch into the appropriate k groups. This conversion consists in determining centers' identifier (i.e., users considered as centers) and the groups' size over these centers. Mainly this decoding phase follows two steps: (1) Binary subsequence represented the group's number is converted to the appropriate integer (i.e., k). (2) According to this groups number, couples of binary subsequences represented centers and groups size are converted to the appropriate user identifier and groups member size (i.e., an integer).

Example 2. Assume a data with $|U| = 20$, a minimum Min_C and a maximum Max_C number of clusters equal to 2 and 4 respectively. The search space is composed of the whole partition ways which is equivalent to the ways of writing $|U|$ as a sum of positive integers by considering all possible permutations. Figure 2 illustrates 4 chromosome instances that encode 4 different solutions which could be decoded into: (a) $k = 2$, $(C_1 = 2,\ \alpha_1 = 4)$, $(C_2 = 5,\ \alpha_2 = 14)$; (b) $k = 4$, $(C_1 = 10,\ \alpha_1 = 4)$, $(C_2 = 5,\ \alpha_2 = 4)$, $(C_3 = 15,\ \alpha_3 = 4)$, $(C_4 = 1,\ \alpha_4 = 4)$; (c) $k = 3$, $(C_1 = 1,\ \alpha_1 = 4)$, $(C_2 = 5,\ \alpha_2 = 8)$, $(C_3 = 15,\ \alpha_3 = 6)$; (d) $k = 4$, $(C_1 = 1,\ \alpha_1 = 4)$, $(C_2 = 4,\ \alpha_2 = 4)$, $(C_3 = 5,\ \alpha_3 = 4)$, $(C_4 = 8,\ \alpha_4 = 4)$.

All chromosomes are 0's and 1's strings of 39 length. Clusters number is encoded on 3 bits. Cluster centers C_i are encoded at least on 5 bits and at most 9 bits (i.e., 5 bits for chromosomes (b) and (d), 6 bits for chromosome (c) and 9 bits for chromosome (a)). Groups size are encoded on 4 bits. According to the decoding phase: first the k clusters number is designed where the three first bits of chromosomes are converted to integer (i.e., 010⇔2, 100⇔4, 011⇔3, 100⇔4 for chromosomes (a), (b), (c) and (d) respectively). Next, using k we can determined the couples of centers identifier and size of the groups over them. The number of bits required to encode centers is $log_2(|U|)$. However, if k is less than Max_C, extra 0's bits are added to X_c. According to this example, nine 0's and three 0's bits are added to centers identifier bits on chromosomes (a) and (b) respectively. Converting these centers identifier bits results users identifier considered as centers. Then, bits representing groups size are converted to the appropriate integer. Finally, each chromosome is converted to its corespondent groups, where each center C_i and its α_i most nearest neighborhoods compose a cluster.

3.2 Fitness Function

A good clustering quality in recommendation context implies maximizing accuracy on the level of each cluster as well as keeping a meaningful distance between centers, our fitness function combines these two measure whereas we have computed chromosome fitness as the next:

$$group\_precision(ch) = (max(r) - min(r)) - (\frac{1}{k} \times (\sum_{i=1}^{k} MAE(G_i)) \tag{2}$$

$$center\_diversity(ch) = \frac{1}{k \times (k-1)} \times (\sum_{i=1}^{k} \sum_{j=i+1}^{k-1} (1 - sim(C_i, C_{j+1}))) \tag{3}$$

$$fitness(ch) = group\_precision(ch) + center\_diversity(ch) \tag{4}$$

Where max(r) and min(r) denote the biggest and lowest rating values equivalent to 5 and 1 respectively, G_i is the ith cluster and C_i is the center of the cluster i. In fact, we have chosen to combine two statistical terms: the first one has the goal of giving a significant indicator about centers positions. Logically, those two centers should not be very close or belonging to the same cluster. However targeting only maximizing distances between pair of centers will lead to searching for the farthest users in search space. For that reason, we added a second term to control the internal prediction error in each cluster.

4 Experiments

In this section, we validate the proposed clustering algorithm experimentally, the obtained results are compared to both KNN algorithm applied directly on the whole dataset, by applying kmeans data reduction and *PCA-GAKM* a well known collaborative filtering.

4.1 Dataset Description

Experiments are performed on movielens data that contains 100.000 ratings assigned by 943 users on 1682 movies, every user gives a rating in a scale of 1 to 5, the main objective of a recommender is to predict the unrated movies which represent about 93% missing values of all possible ratings. We have kept 90% of ratings for training while carrying out randomly 10% of ratings for the test.

4.2 Evaluation Measure

There are several evaluation metrics in the literature. The available evaluators include rating prediction error and recommendation set evaluators. We have evaluated our algorithm from both of these aspects. We cite in next items the employed evaluation metrics which we used in the experimental section.

Rating Prediction Evaluation. Mean Absolute Error (MAE) is a statistical measure used widely over different studies to evaluate the recommendation accuracy. Whereas MAE computes the proportion of the sum absolute difference between every real rating in the test set and its equivalent predicted by the recommender, as shown in the Eq. 5:

$$MAE = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i \in I_u} |p_u^i - r_u^i|}{|I_u|} \tag{5}$$

Where $|U|$ is the number of user, P_u^i is the predicted value, $|I_u|$ is the number of rated items by the user u.

Recommendation Set Evaluation. In contrary of rating prediction evaluation, recommendation set evaluation gain an increasing importance. We used both of precision and recall for validation our results. Whereas for a u's recommendation set I_u, correct u's recommendation set I_u^c and a minimum relevance threshold β we define precision and recall in the next items.

A. Precision: this metric computes the portion of relevant items in the recommendation set among the total list.

$$Precision = \frac{1}{|U|}\sum_{u \in U}\frac{|\{i \in I_u | R_{ui} > \beta\}|}{|I_u|} \quad (6)$$

B. Recall: this metric computes the portion of relevant items in the recommendation set among the total relevant items.

$$recall = \frac{1}{|U|}\sum_{u \in U}\frac{|\{i \in I_u | R_{ui} > \beta\}|}{|\{i \in I_u^c | R_{ui} > \beta\}|} \quad (7)$$

4.3 Experimental Design

Our main objective is to validate the efficiency of the proposed algorithm against two of state of the art algorithms, the proposed GA configurations consist at initializing the number of chromosomes at 20, the maximal iteration number at 200. While for both K-means and *PCA-GAKM* we chose the best clustering configuration that allows minimizing their MAE.

4.4 Analyzing Neighbor Size Parameter

Collaborative filtering algorithm involves two stages: a neighbor selection process followed by a rating prediction step. In particular, neighbors selection deals with two important parameters. Expressly the distance measure choice and the size of the neighborhood. For the first parameter we have employed euclidean distance however for the second one we moved the number of neighbors from 5 to 60 by an increment of 5 each time.

KNN algorithm has been applied one time on the whole dataset another one separately on each cluster gotten using the different clustering algorithms: GA based clustering, K-means, PCA-GAKM. In detail, We have applied different evaluation metrics on KNN considering both rating accuracy and recommendation set evaluators, Whereas we adopted MAE metric for validating rating accuracy. While for recommendation set evaluation, We have used precision and recall by fixing the number of recommendation at 5.

Prediction Rating Accuracy. We have used MAE measure in wide neighbor size range to evaluate our results against baseline recommender. Notably that The results shown in the Figs. 3 and 4 show the superiority of our algorithm against both of classical KNN and the clustering techniques. Whereas we observe that in general MAE trace descends by increasing the size of neighbors. In fact K-means clustering is almost stable in MAE range of 0.81 to 0.82 while our algorithm gives widely better results than KNN as well as it overpasses PCA-GAKM after a neighbors size of 10.

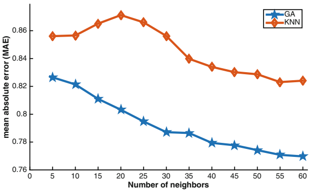

Fig. 3. Proposed algorithm against KNN

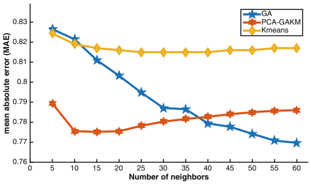

Fig. 4. Proposed algorithm against Kmeans and PCA-GAKM

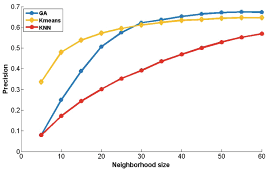

Fig. 5. Precision comparison

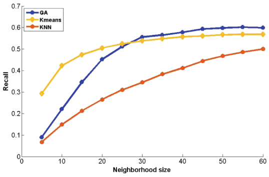

Fig. 6. Recall comparison

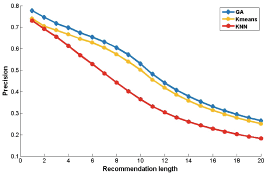

Fig. 7. Comparison of precision with different recommendation length

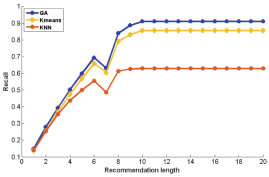

Fig. 8. Comparison of recall with different recommendation length

Recommendation Set Accuracy. We examine our proposed algorithm in terms of precision and recall against KNN and K-means algorithm. We observe in Figs. 5 and 6 that both of clustering algorithms are better than KNN. Notably that K-means starts better than our proposed algorithm however after a neighborhood size of 30 it becomes worse.

4.5 Analyzing Recommendation Length Parameter

We analyze in this section the number of recommendation parameter in terms of accuracy and recall. The length of recommendation has been increased incrementally from 1 to 20. The results in Figs. 7 and 8 show precision and recall variation while increasing recommendation length. We observe that precision corresponds inversely and recall corresponds directly the recommendation length. Additionally, both of clustering algorithms keep giving better results than KNN with a clear superiority of our algorithm against Kmeans.

As a summarize, the results show that the users closeness computed using euclidean distance in the classical KNN for a k ranging between 5 and 60 could not ensure the best precision values. In fact, higher neighbor size increases accuracy in cost of complexity since looking for 60 neighbors in KNN is harder than clustering. Because in the first case the search is open to include the whole set members while in the second one is limited by the cluster's size. Notably that the clustering approach has achieved a better balance between results accuracy and the scalability. Whereas, the proposed genetic algorithm overpassed KNN as well as both of Kmeans and PCA-GAKM clustering.

5 Conclusion

Scalability problem is a major drawback in collaborative memory based filtering algorithm. We have targeted in this paper this issue by proposing a genetic based partitioning algorithm in the aim of better clustering users. The treated problem have been encoded to reduce search space by indicating the number of possible clusters as well as specifying their maximum and minimum size. Besides that, we have maximized the quality of clustering in a way to achieve more accurate results. However, sparsity problem might influence similarity measurement between the set of users. Many statistical methods have emerged to make users profiles more dense, in addition, it starts to appear that accuracy is an insufficient measure to evaluate the satisfaction of users. Pushed by a large amount of redundant recommendations, diversity and novelty of recommendation are becoming more interesting quality measures. Our next intent is orienting clustering to diversity enrichment purpose passing first by addressing sparsity problem faced during our current research.

References

1. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
2. Candillier, L., Meyer, F., Boullé, M.: Comparing state-of-the-art collaborative filtering systems. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 548–562. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73499-4_41
3. Karabadji, N.E.I., Beldjoudi, S., Seridi, H., Aridhi, S., Dhifli, W.: Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. Expert Syst. Appl. **98**, 153–165 (2018)
4. Mustafa, N., Ibrahim, A.O., Ahmed, A., Abdullah, A.: Collaborative filtering: techniques and applications. In: 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), pp. 1–6, January 2017
5. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user similarity model to improve the accuracy of collaborative filtering. Knowl. Based Syst. **56**, 156–166 (2014)

6. Narang, A., Srivastava, A., Katta, N.P.K.: Distributed scalable collaborative filtering algorithm. In: Jeannot, E., Namyst, R., Roman, J. (eds.) Euro-Par 2011, Part I. LNCS, vol. 6852, pp. 353–365. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23400-2_33
7. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
8. Zahra, S., Ghazanfar, M.A., Khalid, A., Azam, M.A., Naeem, U., Prugel-Bennett, A.: Novel centroid selection approaches for KMeans-clustering based recommender systems. Inf. Sci. **320**, 156–189 (2015)
9. Cao, F., Liang, J., Jiang, G.: An initialization method for the K-means algorithm using neighborhood model. Comput. Math. Appl. **58**(3), 474–483 (2009)
10. Li, C.S.: Cluster center initialization method for K-means algorithm over data sets with two clusters. Procedia Eng. **24**(2011), 324–328 (2011). International Conference on Advances in Engineering
11. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., de Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **39**(2), 133–155 (2009)
12. Kuo, R., Wang, H., Hu, T.L., Chou, S.: Application of ant K-means on clustering analysis. Comput. Math. Appl. **50**(10), 1709–1724 (2005)
13. Kalyani, S., Swarup, K.: Particle swarm optimization based K-means clustering approach for security assessment in power systems. Expert Syst. Appl. **38**(9), 10839–10846 (2011)
14. Alhijawi, B., Kilani, Y.: Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6, June 2016
15. Kim, K.J., Ahn, H.: A recommender system using GA K-means clustering in an online shopping market. Expert Syst. Appl. **34**(2), 1200–1209 (2008)
16. Fong, S., Ho, Y., Hang, Y.: Using genetic algorithm for hybrid modes of collaborative filtering in online recommenders. In: 2008 Eighth International Conference on Hybrid Intelligent Systems, pp. 174–179, September 2008