


A collaborative recommender system enhanced with particle swarm optimization technique

Rahul Katarya¹  · Om Prakash Verma¹

Received: 25 September 2015 / Revised: 9 March 2016 / Accepted: 18 March 2016 /

Published online: 29 March 2016

© Springer Science+Business Media New York 2016

Abstract In a web environment, one of the most evolving application is those with recommendation system (RS). It is a subset of information filtering systems wherein, information about certain products or services or a person are categorized and are recommended for the concerned individual. Most of the authors designed collaborative movie recommendation system by using K-NN and K-means but due to a huge increase in movies and users quantity, the neighbour selection is getting more problematic. We propose a hybrid model based on movie recommender system which utilizes type division method and classified the types of the movie according to users which results reduce computation complexity. K-Means provides initial parameters to particle swarm optimization (PSO) so as to improve its performance. PSO provides initial seed and optimizes fuzzy c-means (FCM), for soft clustering of data items (users), instead of strict clustering behaviour in K-Means. For proposed model, we first adopted type division method to reduce the dense multidimensional data space. We looked up for techniques, which could give better results than K-Means and found FCM as the solution. Genetic algorithm (GA) has the limitation of unguided mutation. Hence, we used PSO. In this article experiment performed on Movielens dataset illustrated that the proposed model may deliver high performance related to veracity, and deliver more predictable and personalized recommendations. When compared to already existing methods and having 0.78 mean absolute error (MAE), our result is 3.503 % better with 0.75 as the MAE, showed that our approach gives improved results.

Keywords Recommender system · Computational intelligence · Fuzzy C-means · Collaborative filtering · Movie

✉ Rahul Katarya
rahulkatarya@dtu.ac.in

Om Prakash Verma
opverma.dce@gmail.com

¹ Department of Computer Science & Engineering, Delhi Technological University, Delhi, India

1 Introduction

In recent times, we have seen numerous applications of recommendation system. The “People you may know” column suggest friends on Facebook and the “recommended channel” column in YouTube recommend the videos according to our interests and past historical searches. These are some of the examples of recommendation used in web network technology. Movie recommendation applications are widely used nowadays which are amalgamated with web and multimedia devices that adopted operating systems such as windows and android. A recommender system is an information filtering technology which is used to give useful recommendations to a group of users for items or services that might interest them [4, 28]. It is based on a utility function that automatically predicts how a user will like an item, depending upon their past behavior, relations with other users, item similarity, context, and various other factors [23, 26]. The recommendation is an instance of data mining where data sets are discovered and arranged in large patterns. Recommendation systems use different techniques, but two of them used mostly, one is content-based (CB) which examine properties of the items recommended [35]. CB based recommender systems deliver the description of the item and knowledge of the user’s preference; another one is collaborative filtering systems that recommend items based on similarity measures among users and items. It builds a model from a user’s past behavior such as items previously purchased; numerical ratings were given to those items [16, 17, 23, 27, 46]. Collaborative filtering (CF) is based on the theory that people who agreed on same decisions in the past will agree with similar decisions in the future, and they will like similar items as they liked in the past [11, 17, 29, 41]. However, it faces various challenges and limitations such as data sparsity [20, 45, 46], whose role is to the evaluation of large itemsets. Another limitation is hard to make predictions based on nearest neighbor algorithms, third is scalability in which numbers of users and numbers of items both increases [18, 19, 25, 41] and last one is the cold start where poor relationship among like-minded people [23, 40, 46]. To address encounters, above-mentioned, a hybrid model based movie recommendation approach is proposed to improve the concerns of both high dimensionality and data sparsity. We moved to other approaches of collaborating filtering, and we landed up on model-based collaborative filtering [20, 29, 33]. It is memory based and uses the entire user-item database to generate prediction and also uses a statistical technique to find nearest neighbors. The clustering technique of model-based collaborative filtering is used because clustering reduces the high dimension of data sparse problem and improves scalability. Clustering group the like-minded people together and thus prevent us from searching the whole separate user space. Then there’s another problem that clustering requires quite large computational time [46]. We use the entire system in two phases: online and offline phase. In offline phase, clustering model results in a relatively small dimensional space and divides active users into different clusters. In online phase, due to predicted ratings of movies, an active user is presented with top-N movie recommendation list. The FCM algorithm is employed to prepare the required clusters because, in FCM a user does not belong to only one cluster but more than one cluster with a different degree of membership, this ensure more accurate recommendation [5, 14, 18, 31]. Besides this, particle swarm optimization (PSO) is adopted, that is an optimization technique to provide better results of FCM. We used PSO instead of GA because it provides a guided change in particles as compared to a random mutation in GA [6–8, 12, 16, 34, 37, 38, 42, 43]. We have used K-means to facilitate basic elements of PSO [8, 14, 15, 24, 30, 36]. We have implemented and perform an experiment on movielens dataset. The results we have obtained have achieved more success than existing

cluster based collaborative filtering method. The rest of this article is organized as follows: Section 2 is dedicated to literature survey which gives details about the related works already done. In section 3 brief overview of PSO, K-Means and FCM are taken into consideration. Fundamentals and methods are also discussed in section 3. In Section 4; we highlighted our proposed framework PSOKM-FCM, which specifies that how the algorithms and techniques behaved in our system. Section 5 specifies the experiment and result performed on movielens dataset, and disused what we have achieved by applying our framework to movielens database. The last section 6 is dedicated to a conclusion and future work with the explanation of achievements and scope of our proposed framework.

2 Related work

Bobadilla et al. [4] performed massive literature survey on recommender systems (RS) and suggested that RS as a valuable tool for the internet that provides relevant information. Nowadays lots of Webs and mobile applications have adopted recommendation approaches for suggesting relevant recommendations. E-commerce and mobile applications now totally depend upon the recommendation techniques and methods. A recommender system is broadly classified into three types of methods: content-based, collaborative based and hybrid based and classified them according to heuristic and model based [1]. Traditional collaborative filtering (TCF) approaches are the most popular approach in a recommendation system research domain. Then, using Pearson correlation coefficient (PCC) measurement, users are grouped such that those who have more similarity are together in a cluster. Next, predicted value of items is calculated and recommended to the user. Then at the end, the top N highest predictive items are selected to recommend the target user [40]. Context based recommender systems are adopting nowadays by various e-commerce industries. In recommendation research authors also applied factorization machines to the context-aware recommender systems [39]. They developed an iterative optimization technique that analytically found the least-square answer for one parameter given the other ones. Factorization Machines (FMs) were applied to model contextual information and deliver context-aware rating predictions. This method resulted in fast context-aware recommendations because the model equation of FMs could be computed in linear time with both the number of context variables and the factorization size. In another research work, a recommender system combined with content-based and collaborative methods was suggested for items of interest to users, and also, exploit item semantics [13]. Authors described a hybrid method in which a user-specific recommendation mechanism was learned that use similarity measures between users and also measures the attributes of items that make them interesting to particular users. A multitask clustering framework was projected for the activity of daily living analysis from visual data gathered from wearable cameras [48]. In their framework, rather than clustering data from different users separately, researchers proposed to look for clustering partitions which were coherent among related tasks. A strategy was suggested that automatically select meaningful semantic concepts for the event detection task based on both the events-kit text descriptions and the concepts high-level feature descriptions [50]. Moreover, authors introduced an event oriented dictionary representation based on the selected semantic concepts. They attempted to learn an event oriented dictionary representation based on the selected semantic concepts. A flexible graph-guided multi-task learning (FEGA-MTL) framework was offered for categorizing the head pose of a person who passages freely in an atmosphere monitored by multiple, large field-of-view surveillance

cameras [47, 49]. The FEGA-MTL framework naturally extended to a weakly supervised setting where the target’s walking route is employed as a proxy instead of head orientation. Some researchers also considered the problem of egocentric activity recognition from unlabelled data within a multi-task clustering framework [10]. Two multi-task clustering (MTC) algorithms were proposed and evaluated on first-person vision (FPV) datasets.

3 Material and methods

The collaborative filtering is based on the user preference generated from the Pearson correlation coefficient (PCC), that compute the similarity between n user and other users [28]. Pearson correlation computes the statistical correlation among two item’s mutual ratings to determine their similarity. The correlation is calculated as following as shown in Eq. 1.

$$S(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \tag{1}$$

Where $S(u, v)$, is the value between -1 and 1 . If a value equals to -1 , there is a perfect negative indication correlation between the two values, vice versa. Here u and v are the users whose similarity has to be calculated. i is the item belonging to a set of items that both the users have rated. $r_{u,i}$ is the rating of the i^{th} item by user u and $r_{v,i}$ is the rating of the i^{th} item by user v . \bar{r}_u is the average rating of user u and \bar{r}_v is the average rating of user v . In [46], it converted the original high dimensional data space into a relatively small dimensional data space for carrying denser information. The main idea is to convert and obtain new coordinate space from the original data, which is denoted by a principal component of data with the largest eigenvalue. Let’s assume we have $m \times n$ matrix as the user-rating and in this n -dimension vector specifies user’s profile. It turns out then principal components after performing eigenvalue decomposition, and we select the only first d components ($d \leq n$) to keep in the new data space which is based on the value of accumulated proportion of 90 % of the origin alone. According to [2, 9, 15, 21, 24, 30, 36, 44], the simplest clustering algorithm used is k -means, which categorizes items into k clusters. Initially, each k cluster contains random items. Then, for each cluster, a centroid (or center) is computed. The distance of each item from the centroids are then calculated and checked. If an item is discovered to be closer to another cluster, meaning that if the distance between them is less then, it is moved to that cluster. Centroids are calculated, again and again; thus, all item distances are checked. This is done until the stability is reached (that is when no items move in the different cluster during an iteration), and thus algorithm ends. In [46], suggested a common drawback of K -means algorithm is the selection of initial seeds (initial centroid). K -means algorithm aims at reducing an objective function, in this case, a squared error function. The objective function is:

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \tag{2}$$

Where $\|x_i^{(j)} - c_j\|^2$ is a selected distance measure among a data point $x_i^{(j)}$ and the cluster centre c_j , i is an indicator of the distance of the n data points from their respective cluster centres.

The k-means algorithm is composed of the following steps: a) Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids. b) Assign each object to the group that has the closest centroid. c) When all objects have been assigned, recalculate the positions of the K centroids. d) Repeat Steps b and c until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated. Our experimental outcomes have been compared with the techniques adopted by authors on the same dataset [46]. The initial centroid could affect the final output and can easily lead to produce local optimum. Particle swarm optimization (PSO) was inspired by the social behavior as it is an evolutionary computation technique [22]. In PSO, the possible resolutions called “particles”, fly around in a multi-dimensional search space, to determine an optimal, or sub-optimal, answer by competition as well as support for themselves. The system primarily has a population of random responses. The individual particle is given a random velocity and is drifted through the d -dimensional problem space. The position or location (x_{id}) and velocity (v_{id}) of every particle i in dimension d is revised based on its previous velocity, the previous best particle location (p_{id} or p_{best}), and the previous global best location or position of a particle in the population (p_{gd} or g_{best}). The key approach of PSO lies in accelerating each particle towards its p_{best} and g_{best} locations at each time step. The authorized velocity and position of i th particle are given in Eqs. 3 and 4, respectively [8]:

$$v_{id} = w \times v_{id} + c_1 \times \text{rand}_1() \times (p_{id} - x_{id}) + c_2 \times \text{rand}_2() \times (p_{gd} - x_{id}). \quad (3)$$

$$x_{id} = x_{id} + v_{id} \quad (4)$$

Where w denotes the inertia weight factor and is usually set to a value in the range of 0.5–1. p_{id} is the location of the particle that understands the best fitness value. p_{gd} is the position of the particles that involve a global best fitness value. c_1 and c_2 are constants recognized as social acceleration coefficients. d signifies the aspect of the problem space. rand_1 , rand_2 are random values in the scope of (0, 1), which ensures wide search through problem space. The inertia weight factor w delivers the essential diversity to the swarm by altering the momentum of particles to escape the stagnation of particles at the local optima. Equation 5 needs each particle to record its existing coordinate x_{id} . The velocity v_{id} specifies the speed of its movement sideways the dimensions in a problem space, and the coordinates p_{id} and p_{gd} . Where the best fitness values were computed [8]:

$$P_i(t+1) = \begin{cases} P_i(t) & f(X_i(t+1)) \leq f(X_i(t)) \\ X_i(t+1) & f(X_i(t+1)) > f(X_i(t)) \end{cases} \quad (5)$$

Where the symbol f represents the fitness function; $P_i(t)$ stands for the finest fitness values and the coordination where the significance was calculated. t denotes the generation step. Finally, PSO algorithm can be outlined as a) Initialize every particle in the population pool within the search space of initial velocities; b) Calculate the fitness value of each of these particles. c) If this value is better than its best fitness value recorded until this time, which is also called as population best (p_{id}), then set the current value as the new p_{id} . d) The global best (p_{gd}) is selected from the candidates, which is the best fitness value of all of them in the pool. This is updated in every iteration after this comparison as the new p_{gd} . e) Calculate the particle velocity according to Eq. 2. f)

Refresh the particle position according to Eq. 3. g) Carry out stages b to e until minimum error criteria are attained. The fuzzy c-means algorithm [3, 5, 14, 31, 32] is a clustering algorithm where each item may belong to two or more clusters, not just one. Here, the degree of membership is considered for each item which is given by a probability distribution over the clusters. Fuzzy c-means (FCM) is a clustering algorithm that is useful when an item’s similarity matches with more than one cluster, and we do not want just to consider but various other clusters too [31]. FCM is different in the sense that it does not decide the entire membership of a data point to a given group; instead, it calculates the likelihood (the degree of membership). Since there is no absolute membership in one cluster but the various percentage of membership in different clusters, FCM can be extremely fast. This is because high accurateness can be achieved with a large number of iterations with this clustering method. FCM is grounded on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty \tag{6}$$

Where m is some real number larger than 1, u_{ij} is the degree of association of x_i in the cluster j . x_i is the i th of d -dimensional dignified data, c_j is the d -dimension center of the cluster. $\|*\|$ is somewhat norm expressing the similarity between any measured data and the center. Fuzzy partitioning is supported out by an iterative optimization of the objective function mentioned above, with the apprise of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \tag{7}$$

This iteration will stop when, $\max_{ij} \{u_{ij}^{(k+1)} - u_{ij}^k\} < \epsilon$ where ϵ is a termination criterion between 0 and 1, whereas k is the iteration steps. This process converges to a local minimum or a saddle point of J_m .

4 PSO-KM-FCM based collaborative filtering framework

In this section, our aim is to develop a hybrid clustering model to improve movie prediction accuracy and recommendation to users. We have used some well-known clustering algorithm such as fuzzy c-mean, k-means clustering algorithm and PSO along with an approach named ‘type division’ method. Initial data contains two parameters ‘movie id’ and its ‘rating’ by a user. Type division method distributes the ratings given over the types that particular movie belongs. Then they are used as initials to a k-mean algorithm which outputs the centers for particular algorithm fuzzy c-mean algorithm which need optimal initial centers for better results, the centers produced by k means are further optimised by using them in fuzzy c-mean. After processing fuzzy c-mean algorithm, final results are combined back and used for predicting ratings for futures

movies given by a particular user based on previous data. As shown in Fig. 1, we developed type division method for movies and adopted particle swarm optimization (PSO). In our proposed recommendation system, we have developed a master collaboration of PSO, K-means and fuzzy c-means. We have converted our initial dataset to a new form in which users are divided based on types of movies they watched. Once we have our new dataset we use it on PSO and K-means combination to find initial centres, which are much accurate and precise centres than the assigning of centres randomly, these centres are used by fuzzy c-means for optimization. Type division method is a fundamental approach that divides users according to their interest in different kinds of movies; it uses initial data set generate 19 files, each containing users having an interest in that particular type of movie. A single movie can belong to many types or categories. After generation of files, we use our algorithm on each file to combine users more precisely by their rating on particular movie belonging to that type. By dividing users according to the categories or type of movies, we find particular user interest and their rating for that particular type. The type division method simply segregated the dataset making the recommendation system more accurate for calculation.

Type division method was applied to the database and that had no effect on K-means being applied first except that K-Means had to be applied for each movie kind in the dataset. The initial centres and clusters given by K-Means provided an alternate to the random allocation of input to PSO for optimized center calculation. The final output of K-Means and PSO was a set of centres for each database file. The centres given by K-Means and PSO were used by FCM algorithm to form the final clusters which were directly used for result calculation process. Figure 1 shows that our approach for proposed movie recommendation system. It initially involves breaking the dataset movies into the 19 types given and applying a series of clustering and optimization algorithms explained below.

4.1 Adopting K-means clustering for centres

K-means algorithm is one of the most popular and commonly used clustering algorithms because of factors like flexibility, simplicity and computation efficiency which arises when considering large amounts of data. K-means one by one calculate 'k' cluster centres to assign users to nearest cluster based on the distance calculated and applied by the Type division method in which movies types are categorized according to their users. When there is no more change in centres, the code comes to convergence. For the movie and their users, K-means algorithm aims to partition these users into 'k' groups automatically.

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - C_j \right\|^2 \quad (8)$$

Where J = objective function, k = number of clusters, n = number of cases, C_j = centroid for cluster j, $x_i^{(j)}$ = case i, $\|x_i^{(j)} - C_j\|^2$ = Distance function.

- (a) Clusters the Movie data into k groups.
- (b) Select k points at random, as cluster centres (C_j where $j = 1, 2..k$).
- (c) Assign objects to their closest cluster centre according to the Euclidean distance function.
- (d) Compute the centroid or mean of all objects in each cluster.
- (e) Repeat steps b, c and d until the same points are assigned to each cluster in consecutive rounds.

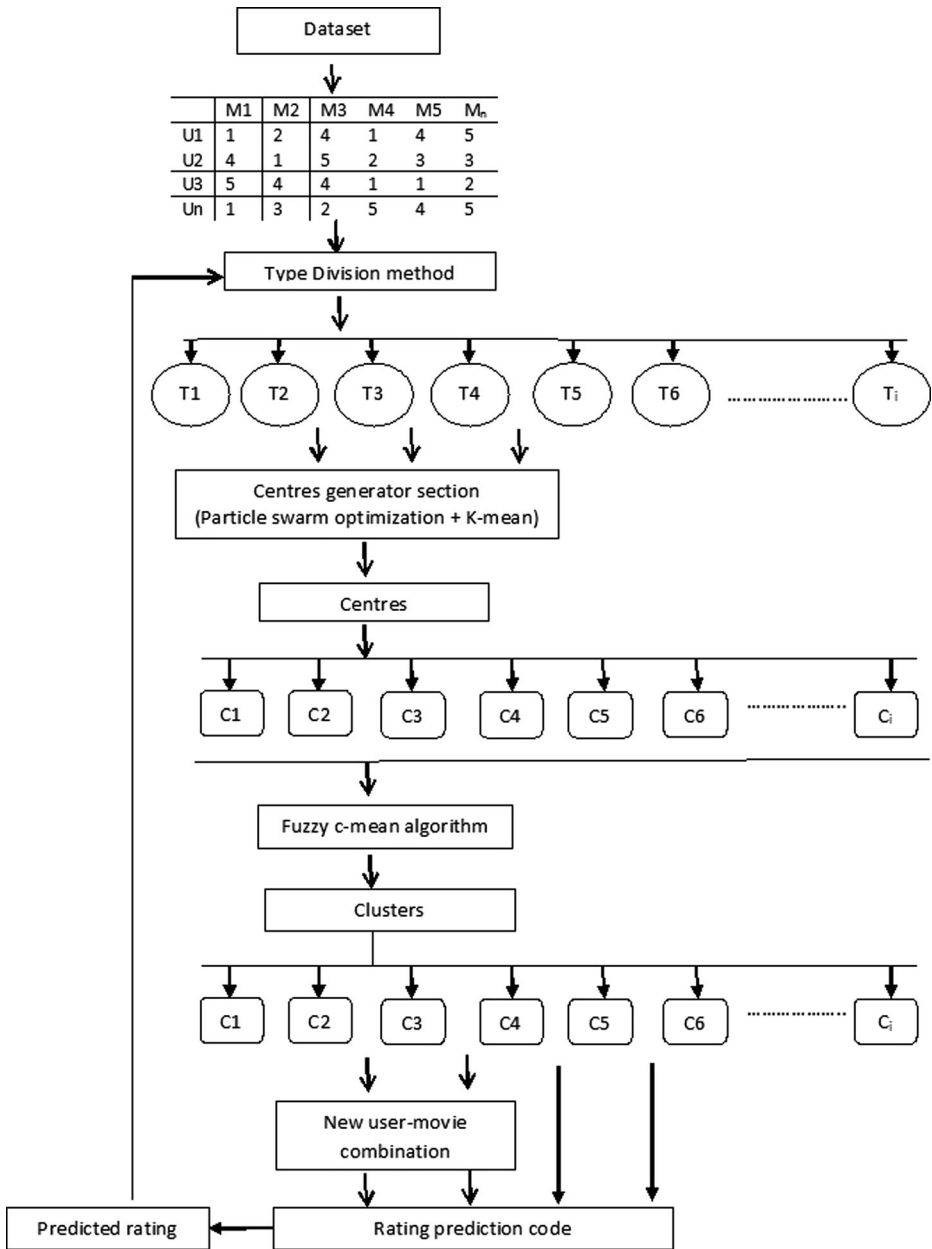


Fig. 1 PSO-KM-FCM based collaborative filtering framework

4.2 Particle swarm optimization application for movie

Let S be the number of users in the swarm, each having a position $x_i \in R^n$ in the defined search-space and having a velocity $v_i \in R^n$. Let p_i is the best known position of user i and g is best position of the whole swarm. A PSO algorithm application for movie is shown in Fig. 2.

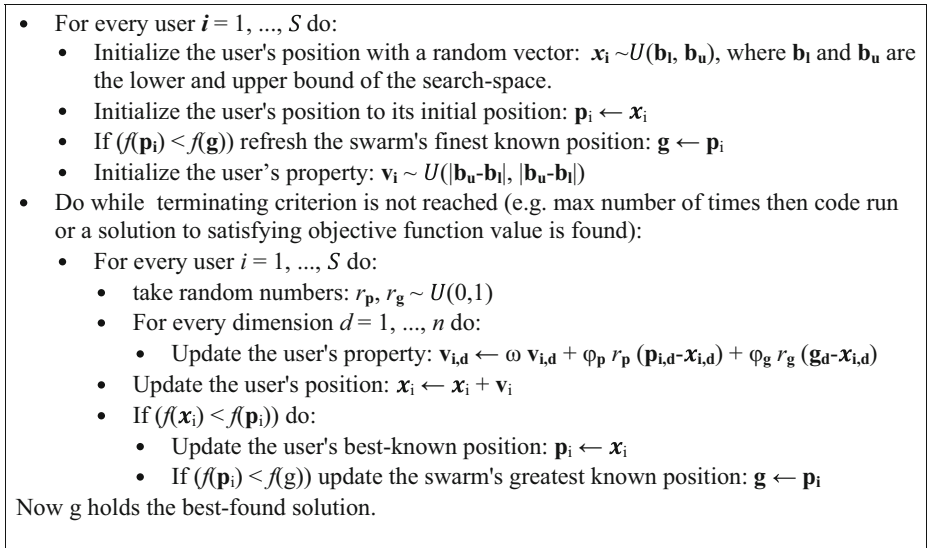


Fig. 2 Particle swarm optimization application for the movie

The parameters ϕ_p , ω and ϕ_g , are nominated by the practitioner. These parameters control the behaviour and efficiency of the PSO algorithm.

4.3 Application of fuzzy c-means and rating prediction

This FCM works by allowing membership to each data point correlating to each cluster center by the distance among the cluster center and the data point. More the data is near to the cluster center more is its membership towards the particular cluster center. Clearly, the summation of membership of each data point should be equal to one. After each iteration membership and cluster, centers are updated according to users and set of centers in Eq. 9, and steps are:

- a) Take cluster centers generated by K-Mean-PSO code.
- b) Calculate the fuzzy membership μ_{ij} using Eq. 9:

$$\mu_{ij} = 1 / \sum_{k=1}^c (d_{ij} / d_{ik})^{\left(\frac{2}{m} - 1\right)} \tag{9}$$

Where, μ_{ij} =fuzzy membership, c = number of cluster centre, d_{ij} =Euclidean distance between i^{th} data and j^{th} cluster center, 'm' is the fuzziness index $m \in [1, \infty]$.

- c) Calculate the fuzzy centres v_j with Eq. 10:

$$v_{ij} = \left(1 / \sum_{i=1}^n (\mu_{ij})^m x_i \right) / \left(\sum_{i=1}^n (\mu_{ij})^m \right), \forall j = 1, 2, \dots, C \tag{10}$$

- d) Repeat step b and c until the minimum 'J' value is achieved

$$\|U^{(k+1)} - U^{(k)}\| < \beta. \text{ Where,}$$

k is the iteration step. β is the termination criterion which is between $[0, 1]$.
 $U = (\mu_{ij})_{n \times c}$ is the fuzzy membership matrix. J is the objective function.

We divided that movie in their types then finds a user based on the previous rating in corresponding types, and then all this rating are combined to get a final predicted rating.

5 Experiments and results

In this section, we discuss the experimental design and empirically investigate our novel proposed movie recommendation system via K-Means-PSO-FCM technique. We evaluate the performance of the proposed method using Mean Absolute Error. Finally, the results are analyzed and discussed here. We carried out all of our experiments and techniques on Pavilion DV6 2.6 GHz, 6.0 GB RAM and Java using Netbeans IDE 7.3.1 to simulate the model.

5.1 Data set and evaluation criteria

We consider the Movielens dataset to conduct the experiments, which is available online including 100,000 ratings by 943 users on 1682 movies and assigned to a discrete scale of 1–5 (<http://grouplens.org/datasets/movielens/100k/>). Every user has rated at least 20 movies. We divide the dataset into 19 types of movies given for each file in the dataset. We have taken two features for identifying the user preferences as input to recommendation system, and they are ‘movies id’ that tells the categories or types it represents and ‘ratings’ given to the movie. We have 19 different kinds in which all the movies gets categories. Each user rating for a particular movie is distributed among the types that movie belongs to. The distribution task is done by the ‘type division’ method, which goes for individual clustering and gets combined again at the end of final results. We implemented our proposed system on the dataset to make the predictions using the final clusters. To check the quality of recommendation, we used the mean absolute error (MAE) as evaluation measure which has been widely used to compare and measure the performance of recommendation systems. The MAE is a statistical accuracy metric which measures the average of the absolute difference between the predicted ratings according to the technique used and actual ratings on test users as shown in Eq. 10. A lower MAE value corresponds to a more accurate prediction [46].

$$\text{MAE} = \frac{\sum |\tilde{P}_{ij} - r_{ij}|}{M} \quad (11)$$

Where M is the total number of predicted movies, P_{ij} represents the predicted value for user i on item j and r_{ij} is the true rating. To understand whether users are interested in the recommendation movies, we can employ the precision and recall metrics which are widely used in recommender systems to understand intelligence level of recommendations. Precision is the ratio of interesting movies retrieved by a recommendation method to the number of recommendations. Recall gives the ratio of interesting movies retrieved that is considered interesting by the user in the actual system. These two measures clearly conflict in nature because increasing the size of recommended movies N leads to an increase in the recall but decrease the precision. However, these evaluations can only be measured on a real online recommender system. The precision and recall for Top- N (N is the number of predicted movies) recommendations are as follows [4]:-

5.2 Results and discussion

The sparse of user-item rating matrix makes it hard to find real neighbours to form the final recommendation list. In our experiments, we compare the performances and some trends of the existing baseline CF movie recommendation systems with our approach, while the neighbourhood size varies from 5 to 60 in an increment of 5. A detailed result is mentioned in Table 1, as it demonstrates efficient experimental results. The mean absolute error is calculated for K-Means-PSO-FCM on different dataset files. We first calculate the MAE using the above-given method on various dataset files namely u1.base, u2.base, u3.base, u4.base, u5.base, ua.base and ub.base. These files contain a different number of movies. However, we get the mean absolute error for each dataset which is later combined to find the MAE of the entire MovieLens dataset.

Table 1 shows the values of mean absolute error for our approach on the MovieLens dataset.

Figure 3 shows the variation of mean absolute error over different files of the dataset and after that we calculated the Mean, Variation and the standard deviation of the MAE calculated: Mean = 0.7458933, Variance = 3.78615 e-5, Standard Deviation = 6.15317 e-3. Mean absolute error comparison between different techniques. We first try to evaluate the movie recommendation quality with the traditional cluster-based CFs. We compare different methods based on the MAE calculated with a different number of neighbours ranging from 5 to 60.

Figure 4 shows the variations in mean absolute error for a different number of neighbours in clusters for various methods. It shows that all methods try to reach the optimum prediction values where the neighbourhood size varies from 15 to 20, and it becomes relatively stable around 60 nearest neighbours. Without the first step of dimensional reduction, GA-KM and SOM gave very close MAE values, and it seems that GAKM produces slightly better prediction than SOM. When coupling with PCA technique, GAKM shows a distinct improvement in recommendation accuracy compared with SOM. Moreover, the proposed K-Means-PSO-FCM produces the smallest MAE values continually where the neighbour size varies. All K-means clustering CF generate increasing MAE values which indicate the decreasing quality for recommendation due to the sensitiveness of the algorithm. Traditional user-based CF produces relatively worse prediction compared with the basic clustering-based methods. However, our method gave the lowest MAE amongst all methods shown above because of the fuzzy logic used.

Table 2 shows the comparison between the variations in the mean absolute error calculated by different techniques by comparing their mean and standard deviations. Our proposed method has the best MAE among all the methods used and a reasonable variation still keeping the value below all the other methods used. Our work is novel in the sense that, this hybrid of PSO, K-Means and Fuzzy-c-means delivered better

Table 1 MAE of separate files of dataset

Dataset	Mean absolute error
U1.base	0.755745
U2.base	0.745370
U3.base	0.738844
U4.base	0.739792
U5.base	0.744568
Ua.base	0.742731
Ub.base	0.754202

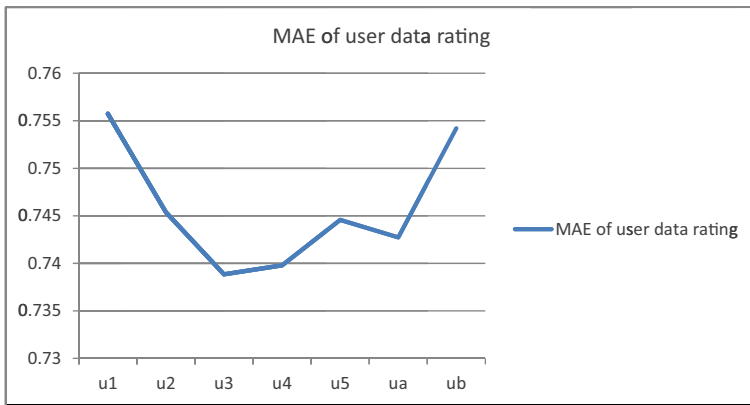


Fig. 3 MAE of separate files of the dataset

results. We initially use FCM, but to make it more efficient, we needed an optimization algorithm. The centers obtained by PSO was used as input for FCM. PSO needs initial clusters and initial centers. For initial clusters, we made random clusters with no similarity factor. For leading centers, we took the average of the no. of movie and rating values and used them as centers. However, the differences between these centers were too small. This produced poor results. So we used K-Means, and the clusters of K-means are used as initial clusters of PSO, and the average of the values is taken as first centers for PSO. We found that the combination of FCM with PSO and K-Means gives lower MAE values than other recommendation systems used before.

6 Conclusion and future work

In this paper, we developed a novel hybrid model based on a collaborative filtering (CF) approach that produces movie recommendations in which we developed type division system for movies and adopted this with particle swarm optimization and clustering algorithms as

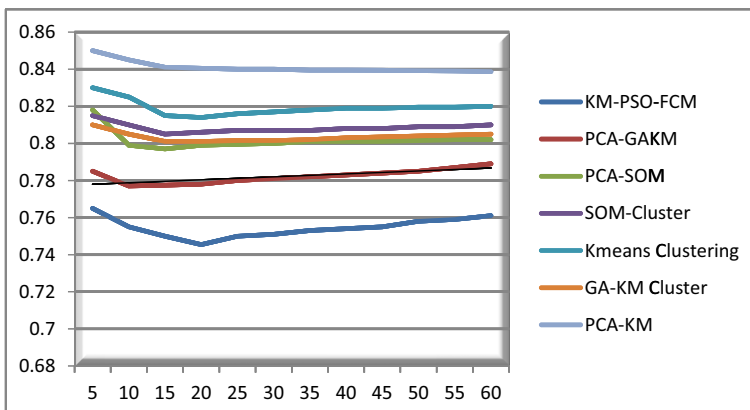


Fig. 4 MAE comparison between different techniques

Table 2 MAE comparison between different techniques

Method	Mean	Standard deviation
Proposed	0.7547	5.067 e-03
PCA-GAKM	0.7821	4.747 e-03
PCA-SOM	0.8018	5.900 e-03
SOM-clustering	0.8078	2.589 e-03
KM-clustering	0.8175	2.244 e-03
GA-KM clustering	0.8040	2.786 e-03
PCA-KM	0.8412	2.845 e-03

fuzzy c-mean and with K-mean algorithm application. In our approach, we divided user-movie according to their movie type, which in return make our dataset more precise, dense, clear and reliable. We used fuzzy c-means (FCM) as the main algorithm for finding a neighborhood for users. To improve the accuracy of FCM we used PSO and K-means algorithm, which give more precise centers for fuzzy c-means and give denser neighborhood selections. We have done the experiment on Movielens dataset; this dataset has many categories. Note that this dataset will only have the ratings of movies rated by users. Users are fully independent and do not have social relationships. Our proposed approach proves that it is capable of generating better prediction accuracy and provide more reliable movie recommendations to users comparing to the present clustering-based collaborative filtering (CCF) approaches. As for future work, we will continue to improve our work approach that can deal with a much larger dataset with higher dimension or attribute. We will add other attributes of users like age, occupation, etc. to give more accurate and reliable rating.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17:734–749. doi:10.1109/TKDE.2005.99
2. Ahmad A, Dey L (2007) A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowl Eng* 63:503–527. doi:10.1016/j.datak.2007.03.016
3. Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. *Comput Geosci* 10:191–203. doi:10.1016/0098-3004(84)90020-7
4. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl Based Syst* 46:109–132. doi:10.1016/j.knosys.2013.03.012
5. Cannon RL, Dave JV, Bezdek JC (1986) Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Trans Pattern Anal Mach Intell* 8:248–255
6. Chen C, Ye F (2004) Particle swarm optimization algorithm and its application to clustering analysis. *IEEE Netw Sens Control* 2:789–794
7. Chen D, Zhao C (2009) Particle swarm optimization with adaptive population size and its application. *Appl Soft Comput* 9:39–48. doi:10.1016/j.asoc.2008.03.001
8. Cui X, Potok TE (2005) Document clustering analysis based on hybrid PSO+K-means algorithm. *Engineering* 185–191
9. Ding C, He X (2004) K-means clustering via principal component analysis. *Twentyfirst Int Conf Mach Learn ICML 04 CI*:29. doi: 10.1145/1015330.1015408
10. Eccv A (2014) Recognizing daily activities from first-person videos with multi-task clustering. *Accv2014*. doi: 10.1007/978-3-319-16817-3_34
11. Ekstrand MD (2010) Collaborative filtering recommender systems. *Found Trends® Hum Comput Inter* 4: 81–173. doi:10.1561/1100000009

12. Fourie PC (2002) The particle swarm optimization algorithm in size and shape optimization. *Struct Multidiscip Optim* 23:259–267. doi:[10.1007/s00158-002-0188-0](https://doi.org/10.1007/s00158-002-0188-0)
13. Garden M, Dudek G (2006) Mixed collaborative and content-based filtering with user-contributed semantic features. *Proc Natl Conf Artif Intell* 21:1307
14. Ghosh S, Dubey S (2013) Comparative analysis of K-means and fuzzy c-means algorithms. *IJACSA* 4:35–39. doi:[10.14569/IJACSA.2013.040406](https://doi.org/10.14569/IJACSA.2013.040406)
15. Hartigan JA, Wong MA (1979) A K-means clustering algorithm. *J R Stat Soc* 28:100–108
16. Hassan R, Cohanin B (2005) A comparison of particle swarm optimization and the genetic algorithm. 1st AIAA Multidiscip Des Optim Spec Conf 1–13. doi: [10.2514/6.2005-1897](https://doi.org/10.2514/6.2005-1897)
17. Herlocker J, Konstan J (1999) An algorithmic framework for performing collaborative filtering. *Proc 22nd ... 8*. doi: [10.1145/312624.312682](https://doi.org/10.1145/312624.312682)
18. Jafar OAM, Sivakumar R (2013) A comparative study of hard and fuzzy data clustering algorithms with cluster validity indices. *Int Conf Emerg Res Comput Inf Commun Appl*
19. Jones MT (2013) Recommender systems, part 2: introducing open source engines explore software for building a recommendation capability. 1–13
20. Kannan R, Ghinea G, Swaminathan S (2015) What do you wish to see? A summarization system for movies based on user preferences. *Inf Process Manag* 51:286–305. doi:[10.1016/j.ipm.2014.12.001](https://doi.org/10.1016/j.ipm.2014.12.001)
21. Kanungo T, Mount DM, Netanyahu NS et al (2002) An efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans Pattern Anal Mach Intell* 24:881–892. doi:[10.1109/TPAMI.2002.1017616](https://doi.org/10.1109/TPAMI.2002.1017616)
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. *IEEE International Conference on Neural Networks, Proceedings* 4:1942–1948. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968)
23. Konstan JA, Riedl J (2012) Recommender systems: from algorithms to user experience. *User Model User Adapt Inter* 22:101–123. doi:[10.1007/s11257-011-9112-x](https://doi.org/10.1007/s11257-011-9112-x)
24. Krishna K, Murty MN (1999) Genetic K-means algorithm. 29:433–439
25. Lekakos G, Caravelas P (2008) A hybrid approach for movie recommendation. *Multimed Tools Appl* 36:55–70. doi:[10.1007/s11042-006-0082-7](https://doi.org/10.1007/s11042-006-0082-7)
26. Lü L, Medo M, Yeung CH et al (2012) Recommender systems. 97
27. Man KF, Tang KS, Kwong S (1996) Genetic algorithms: concepts and applications. *IEEE Trans Ind Electron* 43:519–534. doi:[10.1109/41.538609](https://doi.org/10.1109/41.538609)
28. Melville P, Sindhvani V (2010) Recommender systems. *Encycl Mach Learn* 829–837. doi: [10.1162/153244302760200641](https://doi.org/10.1162/153244302760200641)
29. Mukherjee R, Sajja N, Sen S (2003) A movie recommendation system—an application of voting theory in user modeling. *User Model User Adap Inter* 13:5–33. doi:[10.1023/A:1024022819690](https://doi.org/10.1023/A:1024022819690)
30. Nazeer K (2009) Improving the accuracy and efficiency of the K-means clustering algorithm. *Proc World Congr I*:1–5
31. Pal R, Nikhil BCJ (1995) On cluster validity for the fuzzy c-means model.pdf. 370–379
32. Pal NR, Pal K, Keller JM, Bezdek JC (2005) A possibilistic fuzzy c-means clustering algorithm. *IEEE Trans Fuzzy Syst* 13:517–530. doi: [10.1109/TFUZZ.2004.840099](https://doi.org/10.1109/TFUZZ.2004.840099)
33. Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. *Expert Syst Appl* 39:10059–10072
34. Parsopoulos KE, Vrahatis MN (2008) Multi-objective particles swarm optimization approaches. *IGI Glob* 20–42. doi: [10.4018/978-1-59904-498-9.ch002](https://doi.org/10.4018/978-1-59904-498-9.ch002)
35. Pera MS, Ng Y-K (2013) A group recommender for movies based on content similarity and popularity. *Inf Process Manag* 49:673–687. doi:[10.1016/j.ipm.2012.07.007](https://doi.org/10.1016/j.ipm.2012.07.007)
36. Pham DT, Dimov SS, Nguyen CD (2005) Selection of K in K-means clustering. *Proc Inst Mech Eng Part C J Mech Eng Sci* 219:103–119. doi:[10.1243/095440605X8298](https://doi.org/10.1243/095440605X8298)
37. Poli R (2008) Analysis of the publications on the applications of particle swarm optimisation. *J Artif Evol Appl* 2008:1–10. doi:[10.1155/2008/685175](https://doi.org/10.1155/2008/685175)
38. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. *Swarm Intell* 1:33–57. doi:[10.1007/s11721-007-0002-0](https://doi.org/10.1007/s11721-007-0002-0)
39. Rendle S, Gantner Z, Freudenthaler C, Schmidt-Thieme L (2011) Fast context-aware recommendations with factorization machines. *Proc 34th Int ACM SIGIR Conf Res Dev Inf* 635–644. doi: [10.1145/2009916.2010002](https://doi.org/10.1145/2009916.2010002)
40. Rosli AN, You T, Ha I et al (2014) Alleviating the cold-start problem by incorporating movies facebook pages. *Clust Comput* 18:187–197. doi:[10.1007/s10586-014-0355-2](https://doi.org/10.1007/s10586-014-0355-2)
41. Shani G, Gunawardana A (2011) Evaluating recommendation systems. *Recomm Syst Handb* 257–297
42. Shi Y, Eberhart R (1999) Empirical study of particle swarm optimization. *Proc Congr Evol Comput* 1945–1950. doi: [10.1109/CEC.1999.785511](https://doi.org/10.1109/CEC.1999.785511)
43. Van Der Merwe DW, Engelbrecht AP (2003) Data clustering using particle swarm optimization. *Congr Evol Comput*. doi:[10.1109/CEC.2003.1299577](https://doi.org/10.1109/CEC.2003.1299577)

44. Wagstaff K, Cardie C, Rogers S, Schroedl S (2001) Constrained K-means clustering with background knowledge. *Int Conf Mach Learn* 577–584. doi: [10.1109/TPAMI.2002.1017616](https://doi.org/10.1109/TPAMI.2002.1017616)
45. Wang L-C, Meng X-W, Zhang Y-J (2012) Context-aware recommender systems. *J Softw* 23:1–20. doi: [10.3724/SP.J.1001.2012.04100](https://doi.org/10.3724/SP.J.1001.2012.04100)
46. Wang Z, Yu X, Feng N, Wang Z (2014) An improved collaborative movie recommendation system using computational intelligence. *J Vis Lang Comput* 25:667–675. doi: [10.1016/j.jvlc.2014.09.011](https://doi.org/10.1016/j.jvlc.2014.09.011)
47. Yan Y, Member S, Ricci E (2007) A multi-task learning framework for head pose estimation under target motion. 6:1–15. doi: [10.1109/TPAMI.2015.2477843](https://doi.org/10.1109/TPAMI.2015.2477843)
48. Yan Y, Ricci E, Liu G et al (2015) Egocentric daily activity recognition via multitask clustering. *Image Process IEEE Trans* 24:2984–2995. doi: [10.1109/TIP.2015.2438540](https://doi.org/10.1109/TIP.2015.2438540)
49. Yan Y, Ricci E, Subramanian R et al (2013) No matter where you are: flexible graph-guided multi-task learning for multi-view head pose classification under target motion. *Proc IEEE Int Conf Comput Vis* 1177–1184. doi: [10.1109/ICCV.2013.150](https://doi.org/10.1109/ICCV.2013.150)
50. Yan Y, Yang Y, Meng D et al (2015) Event oriented dictionary learning for complex event detection. 24: 1867–1878



Rahul Katarya is working as an Assistant Professor in Department of Computer Science & Engineering in Delhi Technological University, Delhi India.



Dr. O. P. Verma is HOD and Professor in Department of Computer Science & Engineering in Delhi Technological University, Delhi India.