Asian Journal of Information Technology 13 (4): 222-234, 2014
ISSN: 1682-3915

Data Clustering using GA with Non-Negative Matrix Factorization

[1]K.S. Kavitha, [2]K.V. Ramakrishnan and [3]Manoj Kumar Singh
[1]Auden Technology and Management Academy, Bangalore, India
[2]C.M.R. Institute of Technology, Bangalore, India
[3]Manuro Technology Research, Bangalore, India

Abstract: In this study, a new approach has applied to define the clustering using factorizing the original data set matrix into two lower dimension matrices namely, two dimensional features data set and a transformation matrix with the help of non negative matrix factorization. This two dimensional feature data set is having the more separation in available different categories and also provide approximated visual information about possible clusters available in data set along with correlation available among them. Two dimension feature sets are a used to obtain the final clusters using optimizing the minimum quantizing error with help of Genetic algorithm. Comparisons are made with other well established algorithms like particle swarm optimization. Benefits of features matrix is also shown in compare to raw data set in terms of obtained cluster performance. K-means algorithm is also applied independently before and after matrix factorization and comparisons are made with other obtained results. Cluster performance indexes are defined in terms of F-measure and purity.

Key words: Data mining, Genetic algorithm, PSO, K-Means algorithm, clustering, matrix factorization

INTRODUCTION

The tremendous growth of scientific databases put a lot of challenges before the researches to extract useful information from them using traditional data base techniques. Hence, effective mining methods are essential to discover the implicit knowledge from huge data warehouses. Data warehouses provide a great deal of opportunities for performing data mining tasks such as classification and clustering. Cluster analysis is one of the major data mining techniques, widely used for many practical applications in various emerging areas like bioinformatics, engineering, biology, medicine and data mining. Clustering is an unsupervised method that subdivides an input data set into a desired number of subgroups so that the objects of the same subgroup will be similar (or related) to one another and different from (or unrelated to) the objects in other groups (Dash *et al*., 2010). A good clustering method will produce high quality clusters with high intra-cluster similarity and low inter-cluster similarity (Xu and Wunsch, 2005). The quality of a clustering result depends on both the similarity measure used by the method and its implementation and also by its ability to discover some or all of the hidden patterns. Cluster analysis is the general task to be solved which means that it is not one specific algorithm. It is an result of various algorithms itself, in order to be efficient at clustering. It is distinguished by various type of clustering: hierarchical (nested) versus partitioned (un-nested). Hierarchical versus partial will be discussed more among different clusters, whether the set of clusters is nested or un nested. In more traditional terminology, it has known as hierarchical or partitional. A partitional is a division of one data set exactly in one subset. If the cluster has sub clusters it obtains the hierarchical clustering which is a set of nested clusters that are organized as a tree. The main node (root) is a cluster and each node is an sub cluster except leaves which sometimes are singleton cluster of individual data objects. In general when we make a comparison between Hierarchical algorithm and Partitioning Methods, the fact is that Hierarchical algorithms cannot provide optimal partitions for their criterion. However, partitional methods assume given the number of clusters to be found and then look for the optimal partition based on the objective function. As researchers mentioned earlier, the most important difference between hierarchical and partitional approach is that hierarchical methods produce a nested series of partitions while partitional methods produce only one. K-means clustering is very simple and fast efficient. This is most popular one and it is developed by Mac Queen (Dembele and Kastner, 2003). The easiness of K-Means Clustering algorithm made this algorithm used in several fields. The K-Means algorithm is effective in producing clusters for many practical applications but the computational complexity of the original K-means algorithm is very high, especially for large data sets. The K-means Clustering algorithm is a Partitioning Clustering Method that separates the data into K groups. One drawback in the K-Means algorithm is that of a priori

Corresponding Author: K.S. Kavitha, Auden Technology and Management Academy, Bangalore, India

fixation of number of clusters (Al-Shboul and Myaeng, 2009; Zhang and Xia, 2009; Yuan *et al*., 2004; Yedla *et al*., 2010). The main objective in cluster analysis is to group objects that are similar in one cluster and separate objects that are dissimilar by assigning them to different clusters. One of the most popular clustering methods is K-means Clustering algorithm (Zhang and Xia, 2009; Fahim *et al*., 2006; Bhattacharya and De, 2008; Yedla *et al*., 2010). It classifies object to a pre-defined number of clusters which is given by the user (assume K clusters). The idea is to choose random cluster centres, one for each cluster. These centres are preferred to be as far as possible from each other. In this algorithm mostly Euclidean distance is used to find distance between data points and centroids (Dembele and Kastner, 2003). The K-means Method aims to minimize the sum of squared distances between all points and the cluster centre. The drawback in the K-means algorithm is that of a priori fixation of number of clusters (Al-Shboul and Myaeng, 2009; Zhang and Xia, 2009; Yuan *et al*., 2004; Yedla *et al*., 2010), sensitive to initial value for different initial value there may be different clusters generated and unable to handle noisy data and outliers. Genetic algorithm (Goldberg, 1989) is a biologically inspired search algorithm. The GA uses and manipulates a population of potential solutions to find the optimal solutions. A generation is completed after each individual in the population has performed the genetic operators. The individuals in the population will be better adapted to the objective/fitness function as they have to survive in the subsequent generations. At each step, the GA selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generation, the population evolves toward an optimal solution. This advantage of GA is used to find the suitable cluster for new data to be inserted in database and the fitness function can be altered to change the behavior of the algorithm.

LITERATURE REVIEW

Numerous researches in literature related to this area have motivated the research work. A way of clustering using biological inspired Genetic algorithm was developed by Kamble (2010) which clusters data in dynamic form. The database is assumed to be clustered initially and every new element is added as without need of changing existing clustered database, another ways of improvement for K-Means Cluster algorithm offers improved simulation results which offers but also offers clustering result is more accurate and effective (Zhang and Fang, 2013). Niknam *et al*. (2008) presented in their study an efficient Hybrid Evolutionary Optimization algorithm based on combining Ant Colony Optimization (ACO) and Simulated Annealing (SA) called ACO-SA for cluster analysis. In this algorithm, the simulated Annealing algorithm as a local searcher for each colony is considered. To evaluate the performance of the hybrid algorithm, it is compared with other stochastic algorithms viz., the original ACO, SA and K-means algorithms on several well known real life data sets. In the new proposed Hybrid Evolutionary algorithm to solve nonlinear partitional clustering problem (Niknam and Amiri, 2010). It is the combination of FAPSO (fuzzy Adaptive Particle Swarm Optimization), ACO (Ant Colony Optimization) and K-Means algorithms called FAPSO-ACO-K which can find better cluster partition. The performance of the proposed algorithm is evaluated through several benchmark data sets. A widely acclaimed research paper developed an algorithm called K-Modes to extend the K-means paradigm to categorical domains. A new dissimilarity measures to deal with categorical objects, replace means of clusters with modes and use a frequency based method to update modes in the clustering process to minimize the clustering cost function. Tested with the well known soybean disease data set. In the conventional paper related to an algorithm (Huang, 1997) called K-Modes to extend the K-means paradigm to categorical domains. Here, it introduces new dissimilarity measures to deal with categorical objects, replace means of clusters with modes and use a frequency based method to update modes in the clustering process to minimize the clustering cost function. Tested with the well known soybean disease data set. Dash and Dash (2012) compared K-Means and Genetic algorithm based data clustering which have been compared on the basis of their working principle, advantage and disadvantage with suitable examples.

NON-NEGATIVE MATRIX FACTORIZATION

Non-negative Matrix Factorization (NMF) is a popular matrix factorization approach that approximates a non-negative matrix X by the product of two non-negative low-rank factor matrices W and H. The recent years have witnessed a surge of interests on Non-negative Matrix Factorization (NMF) from the artificial intelligence field. Different from traditional spectral decomposition methods such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), NMF is usually additive which results in a better interpretation ability, does not require the factorized latent spaces to be orthogonal which allows more flexibility to adapt the representation to the data set. Different to other matrix factorization approaches, NMF takes into account the

fact that most types of real-world data, particularly all images or videos are non-negative and maintain such non-negativity constraints in factorization.

Non-negative Matrix Factorization (NMF) problem which can be stated in generic form as follows. Given a non-negative matrix $A \epsilon R^{m \times n}$ and a positive integer $k<\min\{m, n\}$, find nonnegative matrices $W \epsilon R^{m \times k}$ and $H \epsilon R^{k \times n}$ to minimize the functional:

$$f(W,H)=\frac{1}{2}\left\|A - WH\right\|_F^2 \quad (1)$$

The product WH is called a non-negative matrix factorization of A although, A is not necessarily equal to the product WH. Clearly the product WH is an approximate factorization of rank at most k but researchers will omit the word "approximate" in the remainder of this study. An appropriate decision on the value of k is critical in practice but the choice of k is very often problem de-pendent. In most cases however, k is usually chosen such that $k<<\min(m, n)$ in which case WH can be thought of as a compressed form of the data in A.

Multiplicative update algorithms: The Prototypical Multiplicative algorithm originated with Seung and Lee (2001). Their multiplicative update algorithm with the mean squared error objective function is provided.

Multiplicative update algorithm for NMF:

```
W = rand (m, k);          % initialize W as random dense matrix
H = rand (k, n);          % initialize H as random dense matrix

    for  i = 1 : Max_iteration
                H = H×(W^TA)/(W^TWH+C);
                H = H×(AH^T)/(WHH^T+C);
     end
```

Alternating Least Squares algorithms: Another class of NMF algorithms is the Alternating Least Squares (ALS) class. In these algorithms, a least squares step is followed by another least squares step in an alternating fashion thus giving rise to the ALS name. ALS algorithms exploit the fact that while the optimization problem of Eq. 1 is not convex in both W and H, it is convex in either W or H. Thus, given one matrix, the other matrix can be found with a simple least squares computation. An elementary ALS algorithm.

ALS algorithm for NMF:

```
W = rand (m, k);        % initialize W as random dense matrix

    for  i = 1 : Max_iteration
            Solve for H in matrix equation W^T WH = W^TA
            Set all negative elements in H to 0
            Solve for W in matrix equation HH^T WT = HA^T
            Set all negative elements in W to 0
    end
```

The ALS algorithms are more flexible allowing the iterative process to escape from a poor path. Depending on the implementation, ALS algorithms can be very fast. The implementation shown above requires significantly less work than other NMF algorithms and slightly less work than an SVD implementation.

GENETIC ALGORITHM

Evolutionary computation, offers practical advantages to the researcher facing difficult optimization problems. These advantages are multi-fold including the simplicity of the approach, its robust response to changing circumstance, its flexibility and many other facets. The evolutionary approach can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. As a result, evolutionary computations have received increased interest, particularly with regards to the manner in which they may be applied for practical problem solving. In nature, evolution is mostly determined by natural selection or different individuals competing for resources in the environment. Those individuals that are better are more likely to survive and propagate their genetic material. The encoding for genetic information (genome) is done in a way that admits asexual reproduction which results in offspring that are genetically identical to the parent. Sexual reproduction allows some exchange and re-ordering of chromosomes, producing offspring that contain a combination of information from each parent. This is the recombination operation which is often referred to as crossover because of the way strands of chromosomes cross over during the exchange. The diversity in the population is achieved by mutation. Evolutionary algorithms are ubiquitous nowadays having been success-fully applied to numerous problems from different domains including optimization, automatic programming, machine learning, operations research, bioinformatics and social systems. In many cases the mathematical function which describes the problem is not known and the values at certain parameters are obtained from simulations. In contrast to many other optimization techniques an important advantage of evolutionary algorithms is they can cope with multi-modal functions. A typical flowchart of a Genetic Algorithm (GA) is shown in Fig. 1.

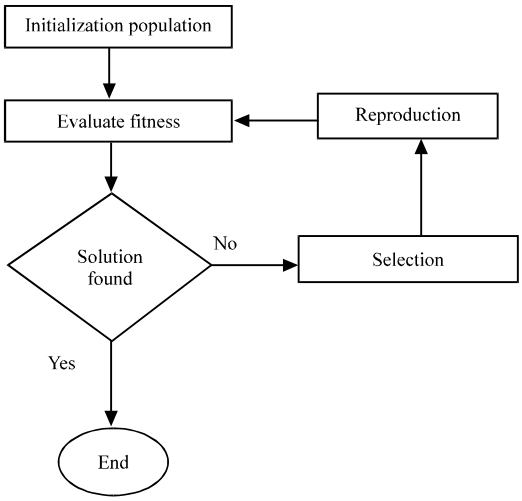

Fig. 1: A flow chart of a Genetic Algorithm (GA)

PARTICLE SWARM OPTIMIZATION (PSO)

PSO's precursor was a simulator of social behavior that was used to visualize the movement of a birds' flock. Several versions of the simulation model were developed incorporating concepts such as nearest-neighbor velocity matching and acceleration by distance. When it was realized that the simulation could be used as an optimizer, several parameters were omitted, through a trial and error process, resulting in the first simple version of PSO. PSO is similar to EC techniques in that a population of potential solutions to the problem under consideration is used to probe the search space. However, in PSO each individual of the population has an adaptable velocity (position change) according to which it moves in the search space. Moreover, each individual has a memory, remembering the best position of the search space it has ever visited. Thus, its movement is an aggregated acceleration towards its best previously visited position and towards the best individual of a topological neighborhood. Two variants of the PSO algorithm were developed. One with a global neighborhood and one with a local neighborhood. According to the global variant, each particle moves towards its best previous position and towards the best particle in the whole swarm. On the other hand, according to the local variant each particle moves towards its best previous position and towards the best particle in its restricted neighborhood. In the following paragraphs, the global variant is exposed (the local variant can be easily derived through minor changes).

Suppose that the search space is D dimensional then the ith particle of the swarm can be represented by a D-dimensional vector, Xi = [xi1, xi2, ..., xiD]. The velocity (position change) of this particle can be represented by another D-dimensional vector V_i = [vi1, vi2, ..., viD]. The best previously visited position of the i-th particle is denoted as Pi = [pi1, pi2, ..., piD]. Defining g as the index of the best particle in the swarm (i.e., the gth particle is the best), n is the best seen by that particular particle and let the superscripts denote the iteration number then the swarm is manipulated according to the following Eq. 2 and 3:

$$V(n+1)\,id = \chi\left[\begin{matrix} wVnid+C1\ r1\left(Pnid-Xnid\right)+ \\ C2\ r2\left(Pngd-Xnid\right)\end{matrix}\right] \quad (2)$$

$$X(n+1)id = Xnid+V(n+1)\,id \quad (3)$$

Where:

w = Called inertia weight
C1, C2 = Two positive constants
r1 = Called cognitive parameter
r2 = Called social parameter
χ = Constriction factor

In the local variant of PSO, each particle moves towards the best particle of its neighborhood. Indeed, the swarm in PSO performs space calculations for several time steps. It responds to the quality factors implied by each particle's best position and the best particle in the swarm, allocating the responses in a way that ensures diversity. Moreover, the swarm alters its behavior (state) only when the best particle in the swarm (or in the neighborhood, in the local variant of PSO) changes thus it is both adaptive and stable.

K-MEANS ALGORITHM

One of the most important components of a clustering algorithm is the measure of similarity used to determine how close two patterns are to one another. K-means clustering group's data vectors into a predefined number of clusters, based on euclidean distance as similarity measure. Data vectors within a cluster have small euclidean distances from one another and are associated with one centroid vector which represents the midpoint of that cluster. The centroid vector is the mean of the data vectors that belong to the corresponding cluster. Standard K-means algorithm is summarized as:

- Randomly initialize the NC cluster vectors
- Repeat

- For each data, assign the vector to the class with the closed centroid where the distance to centroid is determined using:

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (4)$$

Where, k subscripts the dimension

- Recalculate the cluster centroid vectors using until a stopping criterion is satisfied:

$$m_j = \frac{1}{n_j} \sum_{\forall z_p \in c_j} z_p \quad (5)$$

Where:

N_d = The input dimension
N_c = Number of cluster centriods
Z_p = The pth data vector
m_j = Centroid of cluster j
n_j = Number of data vectors in cluster j
C_j = The subset of data vectors that form cluster j

In this case, when there is little change in the centroid vectors over a number of iterations.

FITNESS CRITERIA

In this study, fitness criteria for all cases are taken as quantization error which is defined as given in Eq. 6:

$$J_e = \frac{\sum_{j=1}^{N_c} \left[\frac{\sum_{\forall z_p \in C_j} d(z_p, m_j)}{|C_{ij}|} \right]}{N_c} \quad (6)$$

Where:

d = Distance to centroid
$|C_{ij}|$ = The number of data vectors belonging to cluster C_{ij}, i.e., frequency of that cluster

EVALUATION METHODS OF CLUSTERING

Quality of clustering in this study are measured according to the three criteria:

- The quantization error
- F-measure
- Purity

Researchers used the F-measure and purity values to evaluate the accuracy of the Clustering algorithms. The F-measure is a harmonic combination of the precision and recall values used in information retrieval. We can calculate the precision P(i, j) and recall R(i, j) of each cluster j for each class i.

F-measure of cluster: If n_i is the number of members of class i, n_j is the number of member of cluster j and n_{ij} is the number of class i in cluster j then P(i, j) and R(i, j) can be defined as:

$$p(i,j) = \frac{n_{ij}}{n_j}; \quad R(i,j) = \frac{n_{ij}}{n_i} \quad (7)$$

The corresponding F-measure F(i, j) is defined as:

$$F(i,j) = \frac{2 \times p(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \quad (8)$$

Then F-measure of the whole clustering result is defined as:

$$F = \sum_i \frac{n_i}{n} \max_j \left(F(i,j) \right) \quad (9)$$

where, n is the total number of data in the data set. In general, the larger the F-measure is the better the clustering results.

Purity of cluster: Purity of a cluster represents the fraction of cluster corresponding to the largest class of data assigned to that cluster thus the purity of cluster j is defined as:

$$\text{Purity}(j) = \frac{1}{n_j} \max_i \left(n_{ij} \right) \quad (10)$$

The purity of the whole clustering result is a weighted sum of the cluster purities:

$$\text{Purities} = \sum_j \frac{n_j}{n} \text{purity}(j) \quad (11)$$

In general the larger the purity value is better the clustering result is.

EXPERIMENT DATA SET

There are five data sets have taken among them two are synthetic data sets where as other three are real data sets.

Data set 1 (Synthetic data set): The 15 dimensional data set having three clusters and in each cluster there are 100 data vectors. Each cluster contains value through following method:

$$\text{Class} = \begin{Bmatrix} 1 \text{ then } 1:5 \in [10\ 20], & \text{others} \in [0\ 3] \\ 2 \text{ then } 6:10 \in [10\ 20], & \text{others} \in [0\ 3] \\ 3 \text{ then } 11:15 \in [10\ 20], & \text{others} \in [0\ 3] \end{Bmatrix}$$

Data set 2 (Iris data set): This is well understood database from UCI repository with 4 inputs and 3 cases. There are 150 data vectors and having interaction among classes.

Data set 3 (Heart disease data set): This is a complex data set from clustering point of view contains 270 data vectors of 2 classes with 13 inputs. Huge interactions available among classes. This data set is available in UCI repository.

Data set 4 (Bolt data set): This data set is available publically in Stat Lib Datasets Archive. This data from an experiment on the affects of machine adjustments on the time to count bolts. There are 40 data vectors of 8 inputs. No information available about how many categories are available there clearly.

EXPERIMENT PROCESS AND PERFORMANCE WITH NNMF

Factorization are defined with two different algorithms namely Multiplicative algorithm and Alternate Least Square algorithm independently to each data set to get a comparative analysis in terms of root mean square residue.

Maximum number of iteration in Multiplicative and ALS algorithms is taken as 200. Performance comparison of factorization between both algorithms in terms of RMS (Root Mean Square) residue have shown in Table 1-4. There are 10 independent experiment trail has given and mean rms residue error is taken as parameter for quality measure. From the result, it is clear that for all data set, ALS algorithm has delivered lesser rms residue error compare to multiplicative algorithm. Hence, ALS algorithm has taken for final clustering operation. rms residue errors with iteration for a single experiment in all case are also shown in Fig. 3, 5 and 7.

Graphical representation show very high convergence rate for all cases. Representation of feature data set after factorization in two dimension mapping is also shown in Fig. 3, 5, 7 and 9, respectively for all four data sets. It is clear with observation with visual means to understand the total No. of clusters available in raw data set.

Table 1: NNMF for data set 1

	Multiplicative		Alternate least square	
Data sets	Iterations	RMS residue	Iterations	RMS residue
1	57	4.36360	200	4.35506
2	50	4.35478	200	4.36415
3	81	4.36360	200	4.35506
4	54	4.35478	200	4.35506
5	83	4.36360	200	4.35506
6	103	4.35478	200	4.35506
7	81	4.36360	200	4.35506
8	36	4.36360	200	4.36415
9	59	4.35478	200	4.35506
10	69	4.36360	200	4.35506
Mean	67.3	4.36010	200	4.35690
SD	19.8049	0.00460	0	0.00380

Table 2: NNMF for data set 2

	Multiplicative		Alternate least square	
Data sets	Iterations	RMS residue	Iterations	RMS residue
1	200	0.161911	140	0.1611
2	200	0.161488	140	0.1611
3	200	0.162018	139	0.1611
4	200	0.161716	140	0.1611
5	200	0.161600	136	0.1611
6	199	0.161092	139	0.1611
7	200	0.161778	140	0.1611
8	200	0.162341	139	0.1611
9	133	0.161218	140	0.1611
10	200	0.161526	139	0.1611
Mean	193.2	0.161700	139.1	0.1611
SD	21.1545	0.000400	1.1972	0

Table 3: NNMF for data set 3

	Multiplicative		Alternate least square	
Data sets	Iterations	RMS residue	Iterations	RMS residue
1	200	6.3407	153	6.2691
2	200	6.3217	153	6.2691
3	200	6.2806	152	6.2691
4	200	6.2965	153	6.2691
5	200	6.2905	154	6.2691
6	200	6.3574	155	6.2691
7	200	6.3224	151	6.2691
8	200	6.3581	151	6.2691
9	200	6.3483	153	6.2691
10	200	6.3703	149	6.2691
Mean	200	6.3287	152.4	6.2691
SD	0	0.0313	1.7127	0

Table 4: NNMF for data set 4

	Multiplicative		Alternate least square	
Data sets	Iterations	RMS residue	Iterations	RMS residue
1	200	5.6547	106	5.6418
2	200	5.6461	111	5.6418
3	200	5.6449	95	5.6418
4	200	5.6492	108	5.6418
5	200	5.6482	106	5.6418
6	200	5.6525	111	5.6418
7	200	5.6476	93	5.6418
8	200	5.6496	100	5.6418
9	200	5.6452	116	5.6418
10	200	5.6429	92	5.6418
Mean	200	5.6481	103.8	5.6418
SD	0	0.0036	8.3506	0

CLUSTERING EXPERIMENTS

Genetic algorithm and PSO are applied before and after factorization for all the data sets. Population size for both algorithms is taken as 20 and 100 for dataset after factorization and before factorization. Tournament selection is applied in GA which have the mutation probability equal to 0.3. Both process are terminated after 1000 iteration. For both algorithms 10 independent trails have given for estimation of clusters. Parameters value for PSO in all cases of simulation defined as: $C_1 = C_2 = 0.5$, $\chi = 0.75$ and inertia weight value decreases from 1.2 towards 0 with iterations. Performances of all algorithms have shown correspondingly in Table 5-8:

- Experiment with data set 1: Table 1-5, Fig. 2-12
- Experiment with data set 2: Table 6, Fig. 13-15
- Experiment with data set 3: Table 7, Fig. 16-18
- Experiment with data set 4: Table 8, Fig. 19-21

Performance analysis of clustering: For all data set experiments are divided into two categories: without

Table 5: Performance comparison in terms of F-measure and purity of cluster for data set 1

Data sets	MFGA (F/P)	MFPSO (F/P)	MFK means (F/P)	GA (F/P)	PSO (F/P)	K-means (F/P)
1	1.0/1.0	1.0/1.0	1.0/1.0	0.72/0.67	0.70/0.71	0.67/0.66
2	1.0/1.0	0.98/0.97	1.0/1.0	0.72/0.68	0.72/0.72	1.0/1.0
3	1.0/1.0	1.0/1.0	1.0/1.0	0.69/0.73	0.73/0.71	1.0/1.0
4	0.98/0.99	1.0/1.0	0.67/0.66	0.75/0.72	0.69/0.67	0.61/0.60
5	1.0/1.0	0.99/0.97	1.0/1.0	0.71/0.73	0.67/0.68	1.0/1.0
6	1.0/1.0	1.0/1.0	1.0/1.0	0.71/0.73	0.75/0.77	1.0/1.0
7	1.0/1.0	0.67/0.78	1.0/1.0	0.74/0.71	0.73/0.75	0.59/0.61
8	1.0/1.0	1.0/1.0	0.7/0.72	0.77/0.78	0.77/0.74	1.0/1.0
9	0.99/0.99	1.0/1.0	1.0/1.0	0.71/0.70	0.70/0.70	0.66/0.67
10	1.0/1.0	1.0/1.0	1.0/1.0	0.72/0.74	0.72/0.69	0.67/0.67

Table 6: Performance comparison in terms of F-measure and purity of cluster for data set 2

Data sets	MFGA (F/P)	MFPSO F/P	MFK means F/P	GA F/P	PSO F/P	K-means F/P
1	1.0/1.0	0.95/0.94	0.96/0.98	0.82/0.87	0.70/0.71	0.87/0.86
2	1.0/1.0	0.97/0.96	0.92/0.91	0.82/0.88	0.72/0.71	0.88/0.89
3	0.97/0.96	1.0/1.0	0.95/0.91	0.89/0.83	0.73/0.72	0.87/0.80
4	0.99/0.98	1.0/1.0	0.89/0.90	0.85/0.82	0.69/0.65	0.81/0.86
5	1.0/1.0	0.92/0.91	0.9/0.90	0.81/0.82	0.67/0.66	0.88/0.81
6	1.0/1.0	1.0/1.0	0.96/0.95	0.83/0.79	0.74/0.72	0.79/0.80
7	0.97/0.98	0.97/0.98	1.0/1.0	0.76/0.79	0.73/0.75	0.79/0.83
8	1.0/1.0	1.0/1.0	0.67/0.72	0.75/0.78	0.78/0.75	0.82/0.82
9	0.98/0.99	1.0/1.0	0.94/0.89	0.77/0.74	0.70/0.72	0.85/0.81
10	1.0/1.0	0.92/0.93	0.89/0.90	0.78/0.75	0.72/0.74	0.87/0.77

Table 7: Performance comparison in terms of F-measure and purity of cluster for data set 3

Data sets	MFGA (F/P)	MFPSO (F/P)	MFK means (F/P)	GA (F/P)	PSO (F/P)	K-means (F/P)
1	0.63/0.70	0.62/0.67	0.66/0.68	0.64/0.70	0.61/0.70	0.57/0.56
2	0.65/0.66	0.65/0.67	0.65/0.66	0.66/0.66	0.62/0.66	0.58/0.59
3	0.64/0.65	0.64/0.64	0.66/0.65	0.64/0.65	0.64/0.62	0.54/0.50
4	0.66/0.63	0.66/0.63	0.67/0.66	0.63/0.64	0.63/0.63	0.51/0.56
5	0.67/0.68	0.66/0.67	0.64/0.64	0.65/0.66	0.62/0.67	0.58/0.51
6	0.66/0.67	0.63/0.65	0.63/0.66	0.64/0.65	0.63/0.66	0.59/0.50
7	0.62/0.63	0.61/0.64	0.62/0.64	0.63/0.62	0.62/0.64	0.59/0.53
8	0.65/0.64	0.63/0.65	0.66/0.63	0.65/0.64	0.65/0.64	0.52/0.52
9	0.62/0.67	0.62/0.68	0.63/0.67	0.61/0.65	0.64/0.63	0.55/0.51
10	0.63/0.64	0.64/0.62	0.63/0.65	0.63/0.63	0.63/0.61	0.57/0.57

Table 8: Performance comparison in terms of F-measure and purity of cluster for data set 4

Data sets	MFGA (F/P)	MFPSO (F/P)	MFK-means (F/P)	GA (F/P)	PSO (F/P)	K-means (F/P)
1	1.0/1.0	1.0/1.0	1.0/1.0	0.76/0.80	0.75/0.78	1.0/1.0
2	1.0/1.0	1.0/1.0	1.0/1.0	0.72/0.79	0.72/0.78	1.0/1.0
3	1.0/1.0	1.0/1.0	1.0/1.0	0.80/0.81	0.79/0.80	1.0/1.0
4	1.0/1.0	1.0/1.0	1.0/1.0	0.75/0.79	0.76/0.80	1.0/1.0
5	1.0/1.0	1.0/1.0	1.0/1.0	0.78/0.80	0.77/0.78	1.0/1.0
6	1.0/1.0	1.0/1.0	1.0/1.0	0.74/0.78	0.75/0.77	1.0/1.0
7	1.0/1.0	1.0/1.0	1.0/1.0	0.78/0.80	0.75/0.78	1.0/1.0
8	1.0/1.0	1.0/1.0	1.0/1.0	0.76/0.78	0.73/0.74	1.0/1.0
9	1.0/1.0	1.0/1.0	1.0/1.0	0.79/0.79	0.77/0.76	1.0/1.0
10	1.0/1.0	1.0/1.0	1.0/1.0	0.77/0.79	0.75/0.78	1.0/1.0

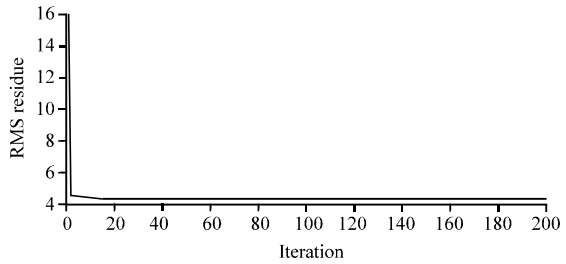

Fig. 2: ALS performance over rms residue

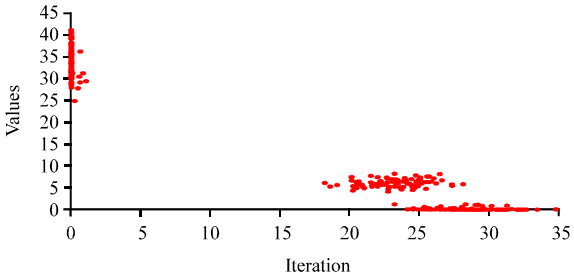

Fig. 3: Transformed 2D data of data set 1

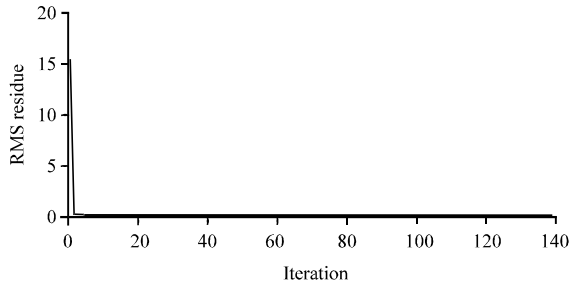

Fig. 4: ALS performance over RMS residue

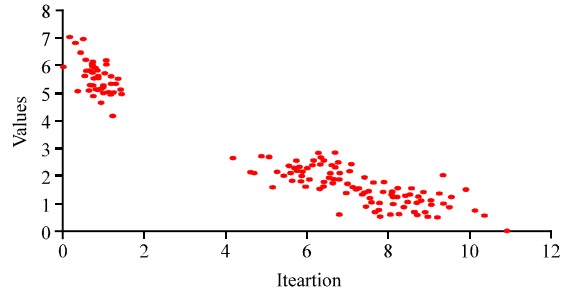

Fig. 5: Transformed 2D data of data set 2

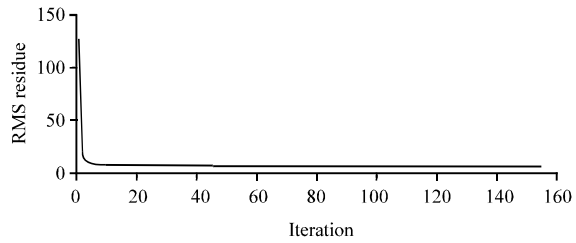

Fig. 6: ALS performance over rms residue

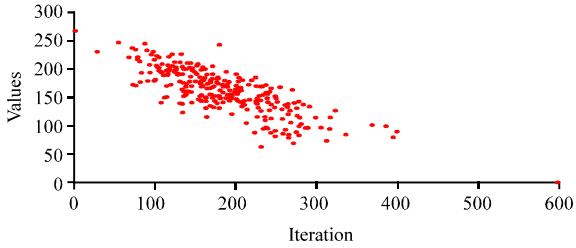

Fig. 7: Transformed 2D data of data set 3

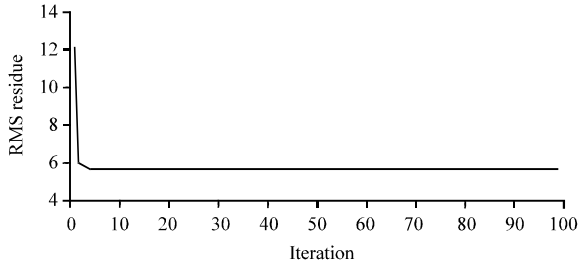

Fig. 8: ALS performnce over RMS residue

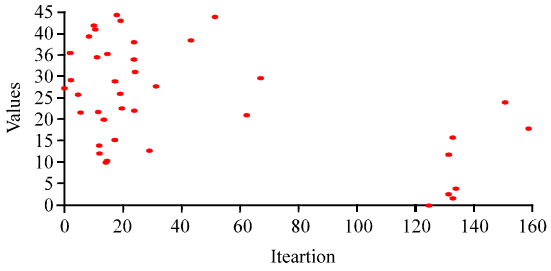

Fig. 9: Transformed 2D data of data set 4

factorization and with factorization. In both case three different algorithms namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and K-means are applied with defined number of clusters to obtain the centroid of clusters.

Result without factorization: By observing the F-measure and purity of clusters in Table 5-8, it is clear that performance deliver by GA is better compare to PSO and k-means for all data sets except for data set 4 in all 10 independent trails. Quantization error optimization performance between GA and PSO for all data set are given in Fig. 9, 12, 15 and 18. For all case GA based clustering is not only deliver minimum quantization error but also faster convergence.

Result with factorization using ALS: Feature matrix obtained in factorization by ALS algorithm is taken as

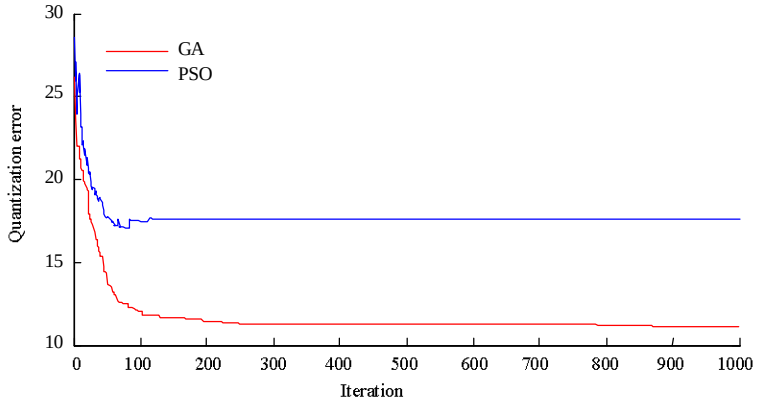

Fig. 10: Quantization error minimization by PSO and GA in raw data set 1

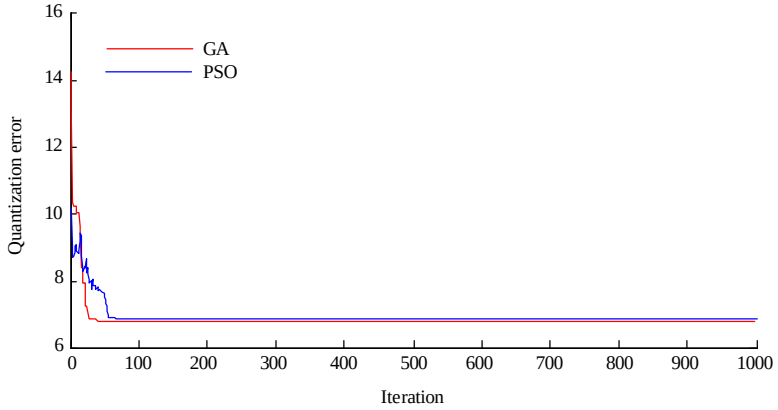

Fig. 11: Quantization error minimization by PSO and GA in factorize data for data set 1

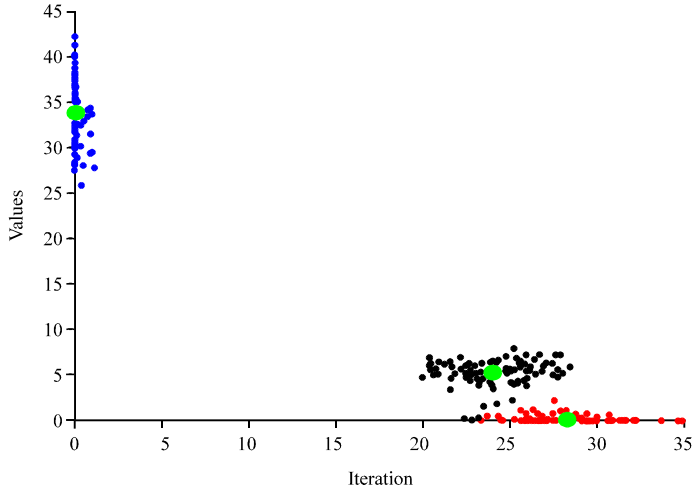

Fig. 12: Cluster and centroid development for factorize data by GA for data set 1

input for same GA, PSO and K-means as it applied in without factorization mode and experiments are given for all data set in 10 independent trails. For understanding purpose here we are saying as namely MFGA, MFPSO and MFKMeans. Obtained results in terms of F-measure and Purity of clusters are given in Table 5-8. When researchers compare with without factorization method in any case, performance improved with high quality with

factorization. Quality to deliver the better clusters by GA is maintained as in case without factorization. Quantization error performance between PSO and GA are shown in Fig. 10, 13, 16 and 19. Final clusters and respective centroid are shown in Fig. 11, 14, 17 and 20.

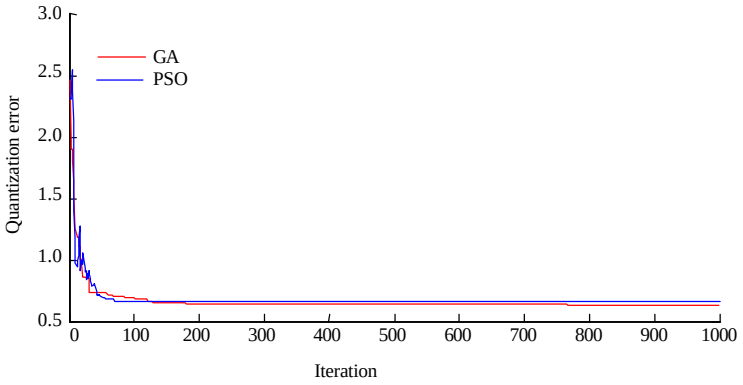

Fig. 13: Quantization error minimization by PSO and GA in raw data set 2

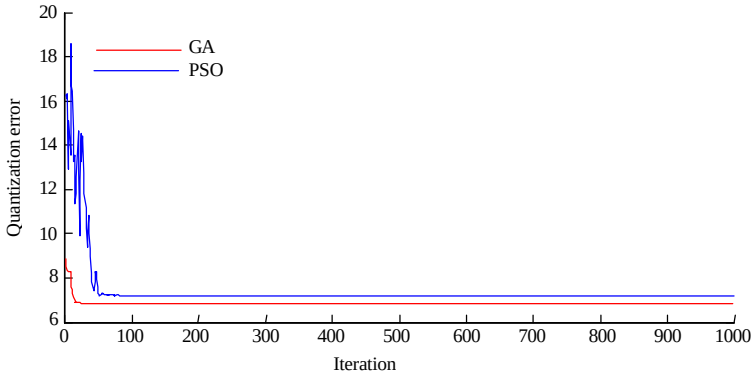

Fig. 14: Quantization error minimization by PSO and GA in factorize data for data set 2

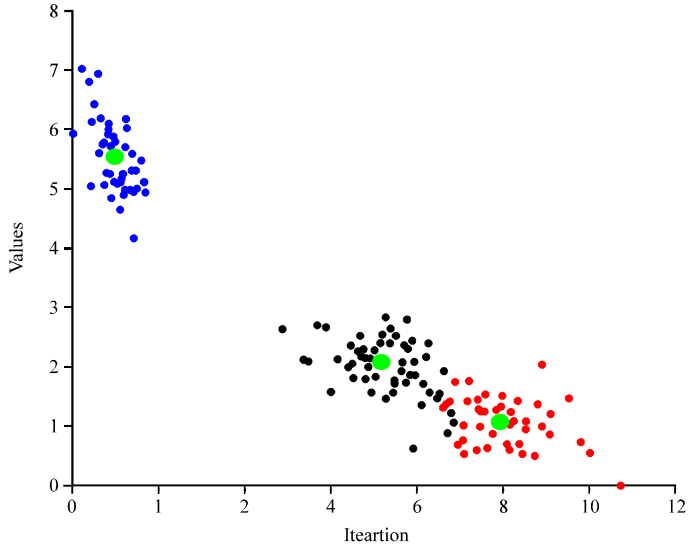

Fig. 15: Cluster and Centroid development for factorize data by GA for data set 2

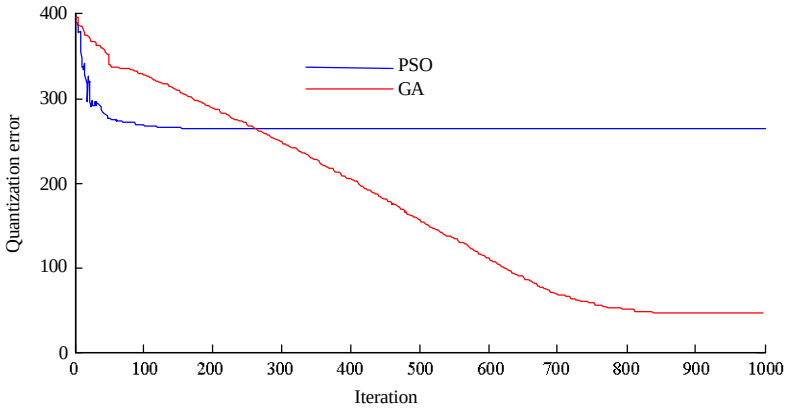

Fig. 16: Quantization error minimization by PSO and GA in raw data set 3

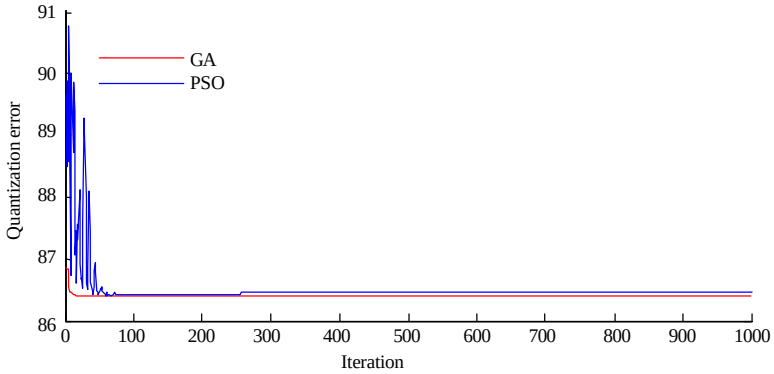

Fig. 17: Quantization error minimization by PSO and GA in factorize data for data set 3

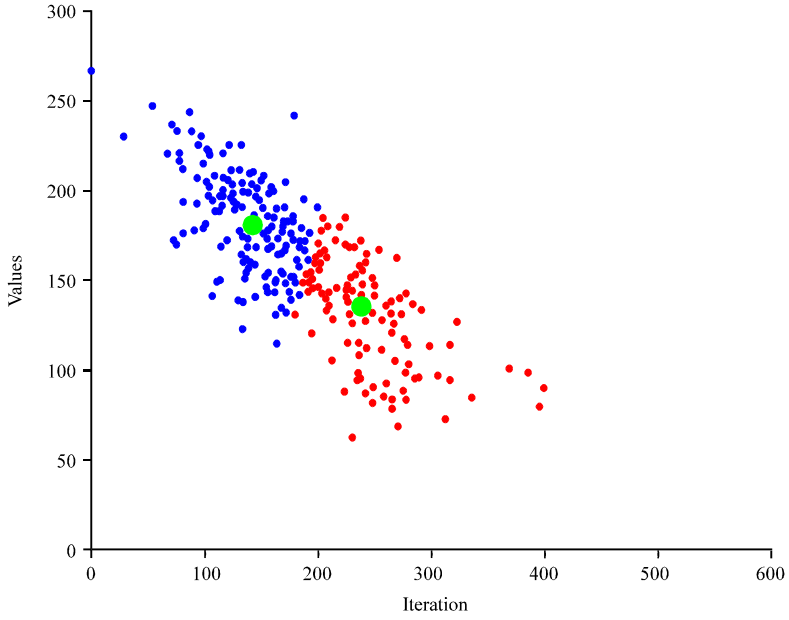

Fig. 18: Cluster and centroid development for factorize data by GA for data set 3

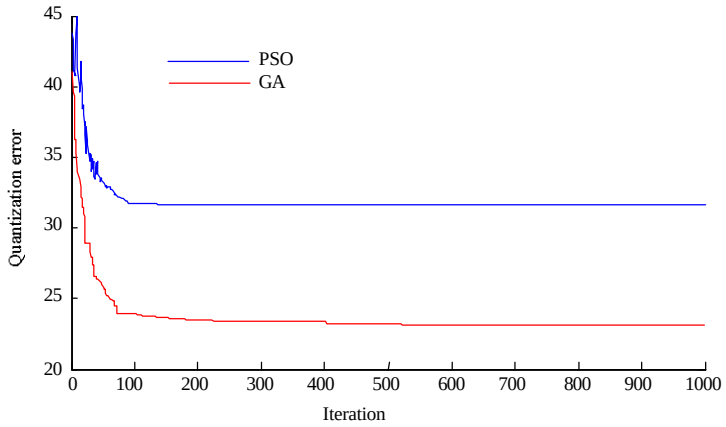

Fig. 19: Quantization error minimization by PSO and GA in raw data set 4

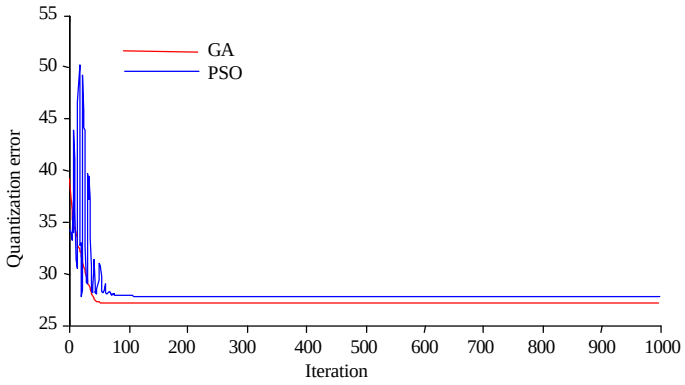

Fig. 20: Quantization error minimization by PSO and GA in factorize data fir data set 4

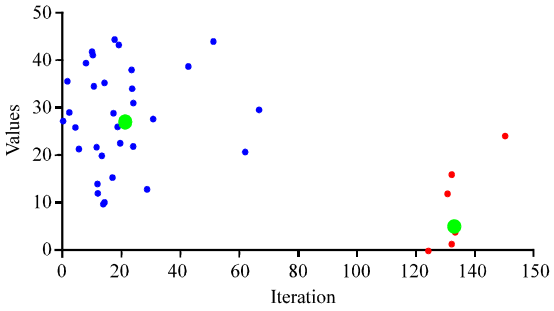

Fig. 21: Cluster and centroid development for factorize data by GA for data set 4

CONCLUSION

Performance of clustering has enhanced with the use of NNMF by transforming raw matrix data set into two smaller matrixes, among them one is two dimensional. This provides number of advantages like reduction in dimensionality, separation of data comes under the different category and visual identification of possible clusters. Among two different non-negative matrix factorization, ALS provides lesser value of rms residue hence final clustering has given with the outcomes of ALS. Comparisons are made between various possibilities with respect to quality of clusters obtain namely by F-measure and purity of clusters. It is observed with experiments that Genetic algorithm in association of ALS based non negative matrix factorization outperformed the PSO and k-means based clustering with and without NNMF.

REFERENCES

Al-Shboul, B. and S.H. Myaeng, 2009. Initializing K-means using genetic algorithms. World Academy of Science, Engineering and Technology, pp: 114-118. http://www.waset.org/journals/waset/v54/v54-21.pdf.

Bhattacharya, A. and R.K. De, 2008. Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes: Detecting varying patterns in expression profiles. Bioinformatics, 24: 1359-1366.

Dash, B., D. Mishra, A. Rath and M. Acharya, 2010. A hybridized K-means clustering approach for high dimensional dataset. Int. J. Eng. Sci. Technol., 2: 59-66.

Dash, R. and R. Dash, 2012. Comparative analysis of K-means and genetic algorithm based data clustering. Int. J. Adv. Comput. Math. Sci., 3: 257-265.

Dembele, D. and P. Kastner, 2003. Fuzzy C-means method for clustering microarray data. Bioinformatics, 19: 973-980.

Fahim, A.M., A.M. Salem, F.A. Torkey and M.A. Ramadan, 2006. An efficient enhanced K-means clustering algorithm. J. Zhejiang Univ. Sci. A, 7: 1626-1633.

Goldberg, D.E., 1989. Genetic algorithms in Search Optimization and Machine Learning. Addison-Wesley, New York, USA.

Huang, Z., 1997. A fast clustering algorithm to cluster very large categorical data sets in data mining. Proceedings of ACM Workshop on Research Issues on Data Mining and Knowledge Discovery, December 1997, Tucson, AZ., pp: 526-529.

Kamble, A., 2010. Incremental clustering in data mining using genetic algorithm. Int. J. Comput. Theory Eng., 2: 1793-8201.

Niknam, T. and B. Amiri, 2010. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied Soft Comput., 10: 183-197.

Niknam, T., B.B. Firouzi and M. Nayeripour, 2008. An efficient hybrid evolutionary algorithm for cluster analysis. World Applied Sci. J., 4: 300-307.

Seung, D. and L. Lee, 2001. Algorithms for non-negative matrix factorization. Adv. Neural Inform. Process. Syst., 13: 556-562.

Xu, R. and D. Wunsch, 2005. Survey of clustering algorithms. IEEE Trans. Neural Networks, 16: 645-678.

Yedla, M., S.R. Pathakota and T.M. Srinivasa, 2010. Enhancing K-means clustering algorithm with improved initial center. Int. J. Comput. Sci. Inform. Technol., 1: 121-125.

Yuan, F., Z.H. Meng, H.X. Zhang and C.R. Dong, 2004. A new algorithm to get the initial centroids. Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, August 26-29, 2004, Shanghai, China, pp: 1191-1193.

Zhang, C. and S. Xia, 2009. *K*-means clustering algorithm with improved initial center. Proceedings of the 2nd International Workshop on Knowledge Discovery and Data Mining, January 23-25, 2009, Moscow, Russia, pp: 790-792.

Zhang, C. and Z. Fang, 2013. An improved K-means clustering algorithm. J. Inform. Comput. Sci., 10: 193-199.