ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

A new framework for Kmeans algorithm by combining the dispersions of clusters

Amruta S. Suryavanshi[1]
Computer Department
TSSM's Bhivarabai Sawant College of Engineering and Research, Pune – India

Prof. Anil D. Gujar[2]
Computer Department
TSSM's Bhivarabai Sawant College of Engineering and Research – India

Abstract: Kmeans algorithm performs clustering by using a partitioning method which partition data into different clusters in such a way that similar object are present in one cluster that is within cluster compactness and dissimilar objects are present in different clusters that is between cluster separations. Many of the Kmeans type clustering algorithms considered only similarities among objects but do not consider dissimilarities. In existing system extended version of Kmeans algorithm is described. Both cluster compactness within cluster and cluster separations between clusters is considered in new clustering algorithm. Existing work initially developed a group of objective function for clustering and then rules for updating the algorithm are determined. The new algorithm with new objective function to solve the problem of cluster compactness within cluster and cluster separations between clusters has been proposed. Proposed FCS algorithm works simultaneously on both i.e. similarities among objects and dissimilarities among objects. It will give a better performance over existing kmeans.

Keywords: Clustering, Data Mining, Feature weighting Kmeans, Fuzzy Compactness and Separation, Fuzzy C Mean.

I. INTRODUCTION

Clustering is a process which partition a given dataset into groups based on given features such that similar objects are stored in a group whereas dissimilar objects are stored in different groups. It is used in many applications such as text organization, image processing and gene analysis and community detection. One of the partitioning methods is Kmeans type clustering algorithm [2], in which the distance between data points and cluster centers is reduced to form a cluster that is it considers cluster compactness within clusters. Existing algorithms of these types are Basic Kmeans, automated variable weighting Kmeans (Wkmeans), attribute weighting clustering algorithms (AWA). In these algorithms the dispersions of cluster is considered for updating weights of feature. It shows, in a given dataset if the objects are similar then they must be evaluated with large weights and if the objects are dissimilar then they must be evaluated with small weights but in some conditions this does not work well. The between cluster separation is important to consider when distinguishing importance of different objects. So centroid of cluster should be away from global centroid [6] [7].

To consider both within cluster compactness and between cluster separations, a new set of clustering algorithms from developed kmeans algorithms is given. These algorithms are Extended kmeans, Extended Wkmeans, and Extended AWA, which extend basic kmeans, Wkmeans, and AWA.

II. LITERATURE SURVEY

By using four concepts literature survey of Kmeans type clustering is given. These four concepts are No Wkmeans algorithm, Vector Wkmeans algorithm, Matrix Wkmeans algorithm and Extensions of Kmeans.

No Wkmeans algorithm

No Wkmeans-type algorithms are divided in two parts: Without intercluster separation and with intercluster separation.

1) No Wkmeans-type algorithms without cluster separations between clusters:

Consider a set of n objects as X = {X1, X2,... Xn}. Object Xi = {xi1, xi2,. . . , xim} to which a set of m features is assigned. U is the membership matrix, it is n × k binary matrix, where uip= 1 shows that object i is belongs to cluster p and if it is not equal to one then it is not belongs to cluster p. A set of k vectors Z = {Z1, Z2,. . . ,Zk} shows the centroid of k clusters. The basic kmeans depends on reduction of the objective function [10].

$$P(U,Z)=\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{ip}(x_{ij}-z_{pj})^2 \qquad \textbf{(1)}$$

Subject to

$$u_{ip}\in\{0,1\}$$

U and Z can be solved by reducing the objective function.

Bisecting technique which is a hierarchical divisive version of kmeans is given by Steinbach et al. [10]. In this technique at each step objects are divided into two clusters, until K numbers of clusters are formed. The problem occurred in this method is to get the exact value of K [10].

2) No Wkmeans algorithms with cluster separations between clusters:

Some validity indexes are used in the clustering method [3] which consider both cluster compactness within cluster and cluster separations between clusters to get the exact value of number of clusters that is K. Yang et al. and Wu et al. [11] gives a fuzzy compactness and separation (FCS) algorithms for between cluster separation which find out the distances between the centers of the cluster and the global center. In the presence of noises and outliers, FCS performs well but the drawback is that all the features are considered equally in this process.

Vector Wkmeans algorithm

In the clustering process considering all features similar is a primary problem of No Wkmeans type algorithms then by adding weights to the features this problem is solved [4].

1) Vector Wkmeans algorithm without cluster separations between clusters:

Vector weighting algorithm is a Wkmeans where weights are added to Kmeans. It is equation as,

$$P(U,W,Z)=\sum_{p=1}^{k}\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_j^{\beta}(x_{ij}-z_{pj})^2 \qquad (2)$$

Subject to

$$u_{ip}\in\{0,1\},\sum_{p=1}^{k}u_{ip}=1,\sum_{j=1}^{m}w_j=1,0\le w_j\le 1$$

Where W is a weighting vector for the features.

A feature selection method called SYNCLUS was introduced by De Sarboet al. [4], in which several groups of features were created and then for each group weights were assigned.

2) Vector Wkmeans algorithm with cluster separations between clusters:

De Soete [5] proposed a technique to find optimal weight for each feature by calculating distance between all pairs of objects but disadvantage of this approach is that large computational cost is required and it is not useful to large dataset.

Matrix Wkmeans algorithm

This techniques group objects into clusters in different subsets of features for different clusters. It includes two tasks: 1) identification of the subsets of features where clusters can be found and 2) discovery of the clusters from different subsets of features.

1)Matrix Wkmeans algorithm without cluster separations between clusters:

Aggarwal et al. [13] proposed the PROjected CLUStering (PROCLUS) algorithm which is able to find a subset of features for each cluster. Using PROCLUS, a user, needs to specify the average number of cluster features.

AWA is a typical matrix weighting clustering algorithm, which can be formulated as,

$$P(U, W, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{ip} \sum_{j=1}^{m} w_{pj}^{\beta} (x_{ij} - z_{pj})^2 \qquad (3)$$

Subject to

$$u_{ip} \in \{0, 1\}, \sum_{p=1}^{k} u_{ip} = 1, \sum_{j=1}^{m} w_{pj} = 1, 0 \le w_{pj} \le 1$$

2)Matrix Wkmeans algorithm with cluster seperation between clusters:

Friedman and Meulman [14] proposed the clustering objects on subsets of features algorithm for matrix weighting clustering which involves the calculation of the distances between all pairs of objects at each iterative step. This results in a high-computational complexity O(tn^{2}m) where n, m, and t is the number of objects, features, and iteration.

*D. **Extensions of Kmeans***

1) Extended Kmeans (E-Kmeans)

Distance between center of cluster and an object that is cluster compactness within cluster are considered in Kmeans. In Ekmeans the concept of global centroid is introduced to get between cluster separations. It tries to reduce the distance between center of the cluster and objects and also try to maximize the distance between global centroid and centroids of clusters. As shown in fig. 1, *Z*0 is the global centroid and *Z*1, *Z*2 are the centers of cluster 1, cluster 2, respectively [1].

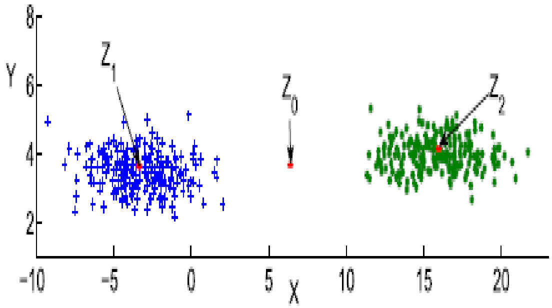


Fig. 1: Effect of between cluster separations

To integrate intracluster compactness and intercluster separation, modification in the objective function, as shown in (1), is done into,

$$P(U, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{ip} \sum_{j=1}^{m} \frac{(x_{ij} - z_{pj})^2}{(z_{pj} - z_{0j})^2} \qquad (4)$$

Subject to

$$u_{ip} \in \{0, 1\}, \quad \sum_{p=1}^{k} u_{ip} = 1.$$

z0j is the jth feature of the global centroid z0 of a data set. z0j can be calculated as

$$z_{0j} = \frac{\sum_{i=1}^{n} x_{ij}}{n}. \qquad (5)$$

Limitations:

Kmeans does not weight the feature that is all the features are treated equally.

2) Extension of Wkmeans (E-WKmeans)

All the objects are considered equal in basic kmeans and Ekmeans. But objects may have different weights. Weights are given to each object in Wkmeans algorithm by considering within cluster compactness. EWkmeans algorithm considers the within cluster compactness and the distances between the centers of all the clusters and global center simultaneously while assigning weights to the feature [1].

Let W ={w1,w2,...,wm} be the weights for m features and β be a parameter for tuning weight wj, then extend (2) into

$$P(U, W, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{ip} \left[\sum_{j=1}^{m} w_j^{\beta} \frac{(x_{ij} - z_{pj})^2}{(z_{pj} - z_{0j})^2} \right] \qquad (6)$$

Subject to

$$u_{ip} \in \{0, 1\}, \sum_{p=1}^{k} u_{ip} = 1, \sum_{j=1}^{m} w_j = 1, 0 \le w_j \le 1$$

Advantages:

E-Wkmeans weights the features with a vector, which means, each feature has a weight representing the importance of the feature in the entire data set.

Limitations:

The same feature in different clusters has the same weight.

3) Extension of AWA (E-AWA)

If the same feature is present in different clusters then the similar weight is assigned in Wkmeans and E-Wkmeans. But in real world applications, the same feature in different clusters has dissimilar weights [1]. This problem is solved in E-AWA by considering two methods that is cluster compactness within cluster and cluster separations between clusters.

Let W ={ W1,W2,...,Wk} be a weight matrix for k clusters. Wp={ wp1,wp2,...,wpm} denotes the feature weights in cluster p, then extend (3) into

$$P(U, W, Z) = \sum_{p=1}^{k} \sum_{i=1}^{n} u_{ip} \left[\sum_{j=1}^{m} w_{pj}^{\beta} \frac{(x_{ij} - z_{pj})^2}{(z_{pj} - z_{0j})^2} \right] \qquad (7)$$

Subject to

$$u_{ip} \in \{0, 1\}, \sum_{p=1}^{k} u_{ip} = 1, \sum_{j=1}^{m} w_{pj} = 1,\ 0 \leq w_{pj} \leq 1.$$

Advantages:

1. This method Consider both intracluster compactness and intercluster separation.

2. This method is robust because it does not introduce new parameter to balance intracluster compactness and intercluster separation.

3. This method Produce better clustering results than earlier methods because it utilize more information than traditional Kmeans type algorithm.

Limitations:

This method may produce errors when centroid of certain cluster is very close to global centroid and centroids among the clusters are not close to each other.

III. Proposed System

In a clustering process addition of the cluster separations between clusters is main task of proposed work. Many of the existing kmeans type algorithms only uses the within cluster compactness. On the other way, proposed work as shown in Fig. 2 consider both the within cluster compactness that is intracluster compactness and the cluster separation between clusters.

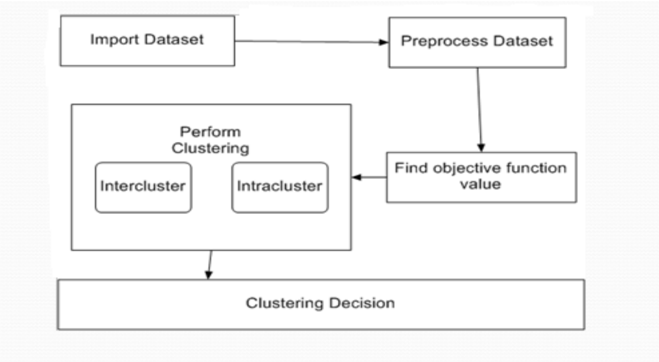


Fig. 2: Architecture of proposed system

Proposed work gives the modified approach based on FCM (Fuzzy C Mean) which consider the cluster compactness within cluster and cluster separation between clusters simultaneously.

IV. Implementation Details

The set of an s dimensional data set is considered as, X={x1,x2,…,xn}. Consider μ (x1)$_1$…… μ (x)$_c$ are fuzzy c-partitions μ_{ij} = μ_i(Xj) represent the degree that the data point xj belongs to cluster i. a1,…a$_n$ are the cluster centers. We can define a objective function as below.

$$J_{FCM}(\mu, a) = \sum_{j=1}^{n} \sum_{i=1}^{c} \mu_{ij}^{m} \left\| x_j - a_i \right\|^2$$

Where weighting factor m represents the degree of fuzziness.

Proposed system modifies the fuzzy compactness and separation algorithm based on fuzzy scatter matrix. It adds penalized term to the existing work as below.

$$J_{FCS}(\mu, a) = \sum_{j=1}^{n} \sum_{i=1}^{c} \mu_{ij}^{m} \left\| x_j - a_i \right\|^2 - \sum_{j=1}^{n} \sum_{i=1}^{c} \eta_i \mu_{ij}^{m} \left\| a_i - \bar{x} \right\|^2$$

Where parameter η_i>= 1, It is known that the J_{FCS}=J_{FCM} When η_i= 0. System will consider the following equation which minimizes the objective functions.

$$\mu_{ij} = \left(\left\|x_j - a_i\right\|^2 - \eta_i \left\|a_i - \overline{x}\right\|^2\right)^{\frac{-1}{m-1}} \Big/ \sum_{k=1}^{c}\left(\left\|x_j - a_k\right\|^2 - \eta_k \left\|a_k - \overline{x}\right\|^2\right)^{\frac{-1}{m-1}}$$

And

$$a_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} x_j - \eta_i \sum_{j=1}^{n} \mu_{ij}^{m} \overline{x}}{\sum_{j=1}^{n} \mu_{ij}^{m} - \eta_i \sum_{j=1}^{n} \mu_{ij}^{m}} \quad ,$$

$$\eta_i = \frac{(\beta/4)\min_{i' \neq i} \left\|a_{i'} - a_i\right\|^2}{\max_k \left\|a_k - \overline{x}\right\|^2}, 0 \leq \beta \leq 1.0$$

Proposed algorithm for Fuzzy Compactness and Separation

Input: Data set

Output: Clusters

1. Import Dataset.
2. Normalize dataset.
3. For all data points in dataset
4. Calculate value of J_{fcs}
5. Compute the value of μ_{ij} for x_j
6. If μ_{ij} has good degree for cluster a_i
7. Add x_j to a_i cluster.
8. Else
9. Check for other clusters centers
10. End for
11. Return clusters

Where,

$J_{fcs=}$ Objective Function by using FCS

$\mu_{ij=}$ Degree to which object x_j belong to cluster a_i

Mathematical Model

Mathematical model of a proposed system is given below.

Let S, be a system such that,

S={s, e, I, O, Fmain, Fs, DD, NDD, ϕ}

Where,

S- Proposed System

s- Initial State i.e. Preprocess Dataset.

PD is a function to preprocess dataset. Let D be the input dataset.

PreD = PD (D)

PreD = Preprocessed Data

D= Input Data

e- End State i.e. Clustering.

CS is clustering function to which outputs the clustered results.

CD = CS (μ_{ij}, ai, xj)

CD = Clustered Data

I- Input of System i.e. Input Dataset.

O- Output of System i.e. Clustered Data.

Fmain- Main function resulting into outcome O. In a given system it will be Fuzzy Compactness and Separation (FCS) algorithm.

OF is objective function which outputs value of objective function.

J_{fcs} = OF(μ_{ij} , ai, xj)

J_{fcs}= objective function

μ_{ij} = Degree by which data point xj belongs to cluster ai

ai = cluster I

xj = Data point

Fs- Supportive function. In given system it will be Fuzzy c-mean (FCM)

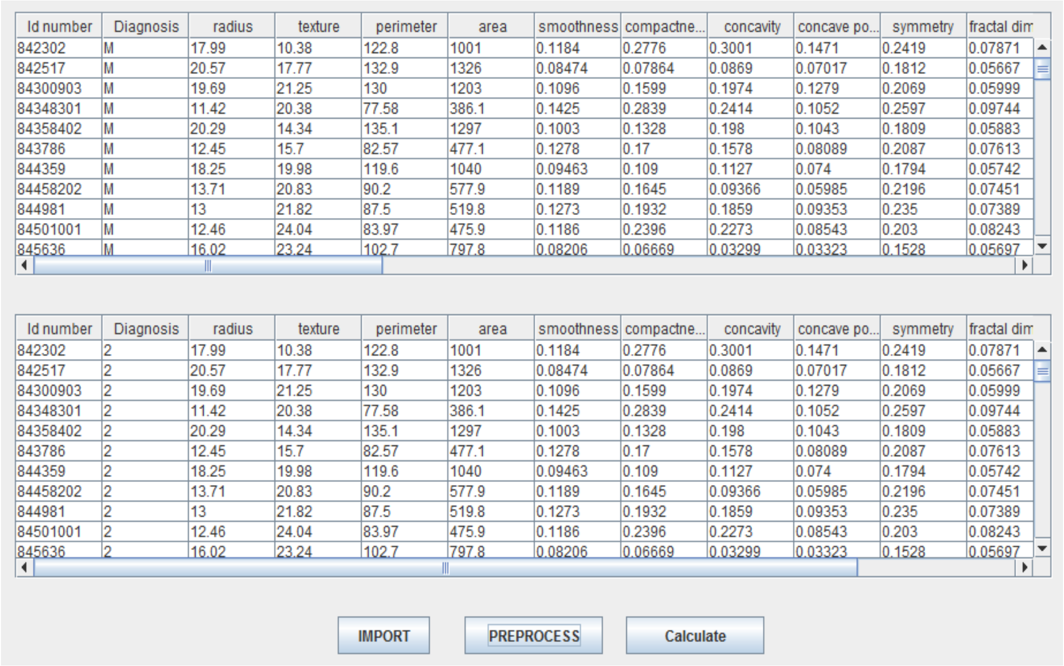
DD- Deterministic Data, i.e. data to be clustered is deterministic.

NDD- Non Deterministic Data i.e. number of clusters is Non Deterministic.

V. Result and Discussion

Description of Dataset

To find out the performance of proposed algorithm real life data set called Wisconsin Diagnostic Breast Cancer (WDBC) is used. Improvement in the Clustering results is done by considering cluster separation between clusters. Ten real-valued features are computed. These are area, smoothness, radius, texture, perimeter, compactness, concavity, concave points, symmetry, and fractal dimension. Attributes used are ID number and Diagnosis.



Id number	Diagnosis	radius	texture	perimeter	area	smoothness	compactne...	concavity	concave po...	symmetry	fractal dim
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697

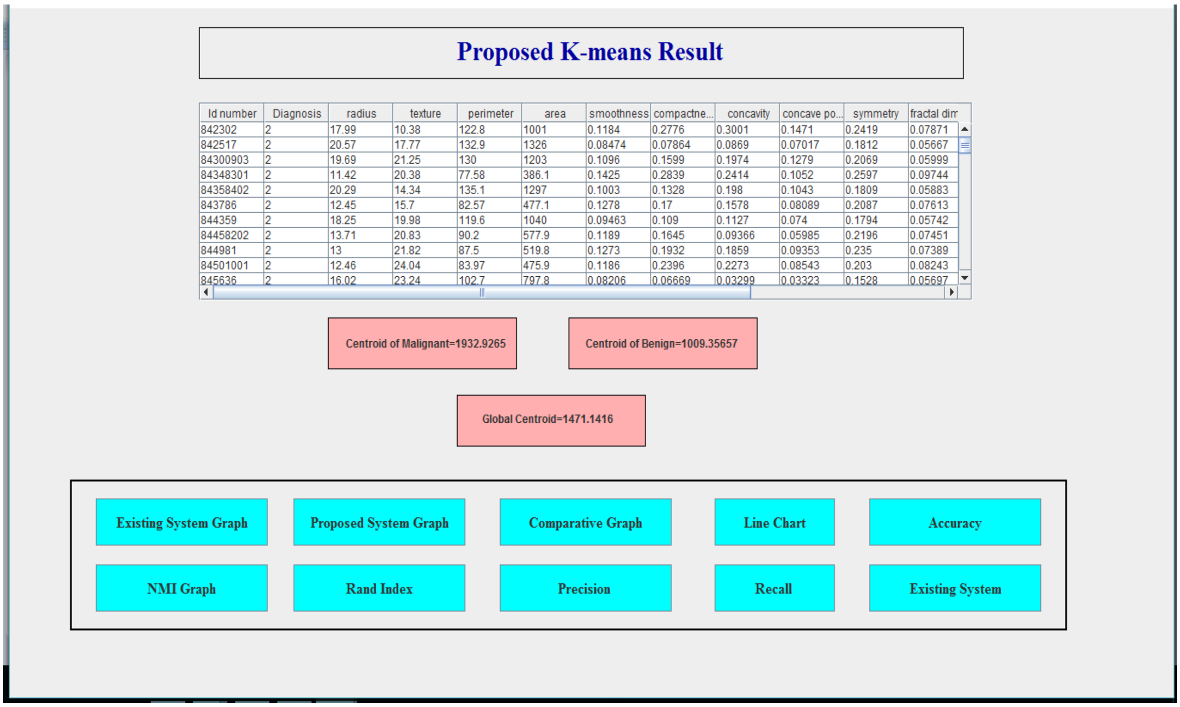
Id number	Diagnosis	radius	texture	perimeter	area	smoothness	compactne...	concavity	concave po...	symmetry	fractal dim
842302	2	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871
842517	2	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
84300903	2	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999
84348301	2	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744
84358402	2	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883
843786	2	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613
844359	2	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742
84458202	2	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451
844981	2	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389
84501001	2	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243
845636	2	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697

Fig. 3: WDBC Dataset

Performance of Metrics

The performance metric used in proposed algorithm is accuracy. Accuracy gives the percentage of the objects that are correctly discovered in a result.

Experimental Result



Proposed K-means Result

Id number	Diagnosis	radius	texture	perimeter	area	smoothness	compactne...	concavity	concave po...	symmetry	fractal dim
842302	2	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871
842517	2	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
84300903	2	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999
84348301	2	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744
84358402	2	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883
843786	2	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613
844359	2	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742
84458202	2	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451
844981	2	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389
84501001	2	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243
845636	2	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697

Centroid of Malignant=1932.9265

Centroid of Benign=1009.35657

Global Centroid=1471.1416

Fig. 4: GUI of proposed system

Comparative Analysis

WDBC Dataset	Existing system	Proposed system
Malaignant centroid	1933.5765	1932.9265
Benign centroid	1009.7533	1009.3565
Global centroid	1471.6649	1471.1416
Accuracy of Malaignant	**92.85**	**96.42**
Accuracy of Benign	**86.20**	**92.59**

Comparative Graph

The performance of proposed system by comparing it with existing system is analyzed in this section.

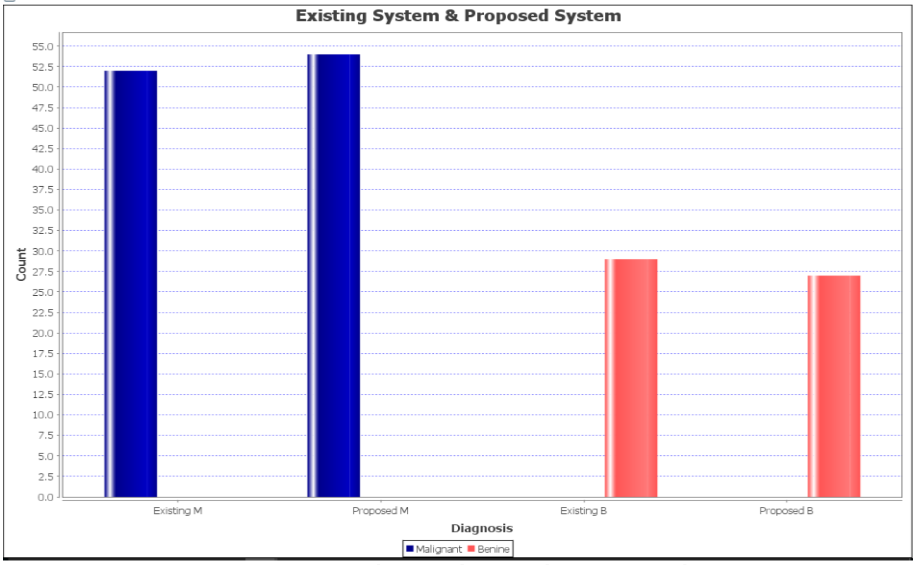


Fig 5: existing and proposed system object count graph

Figure below shows clustering accuracy for proposed system.

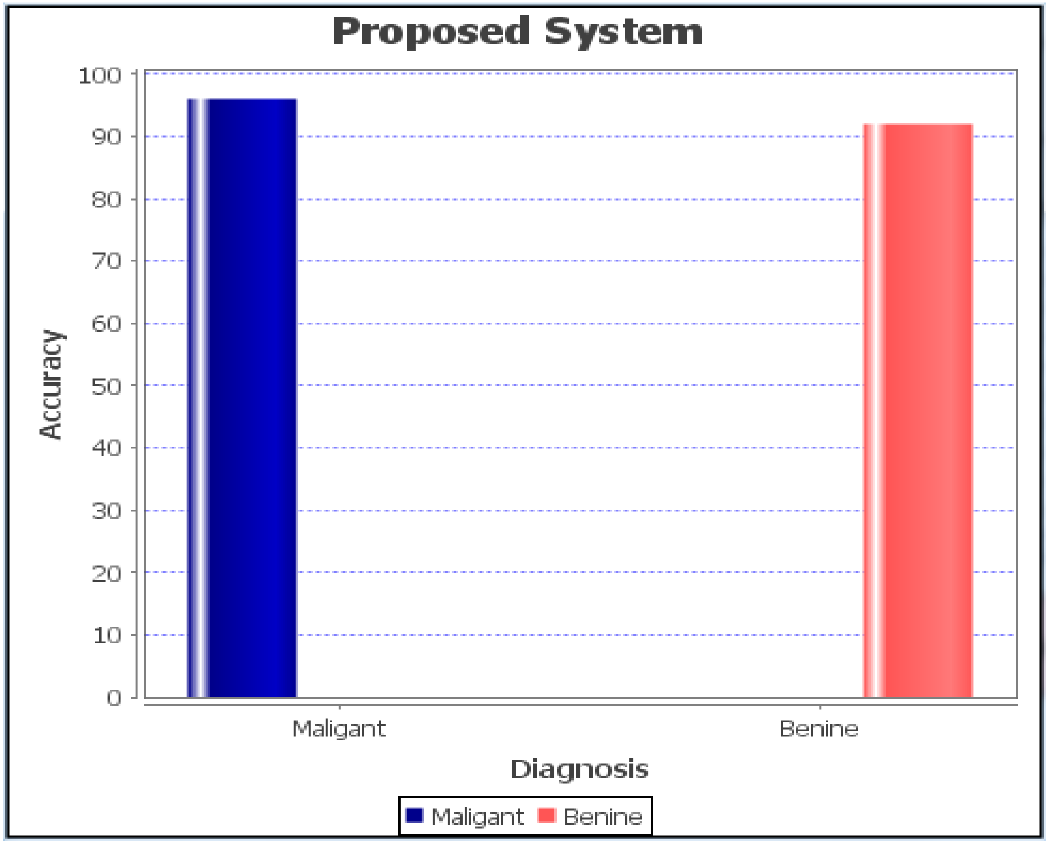


Fig 6: Accuracy graph for proposed system

Figure below shows scatter graph for proposed system that shows malignant cluster and benign cluster.

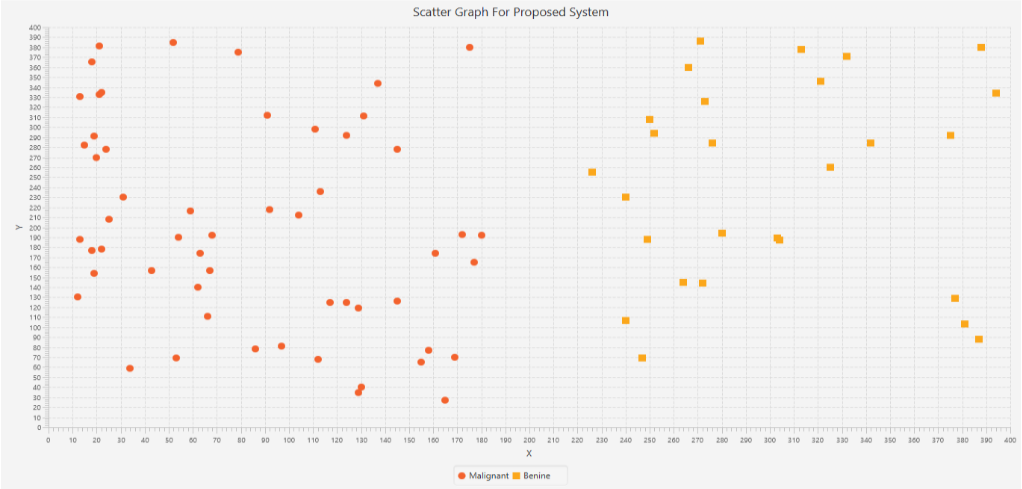


Fig 7: Scatter graph for proposed system

VI. Conclusion

A new framework for kmeans-type algorithms which include both cluster compactness within cluster and the cluster separations between clusters are proposed. Three extensions of kmeans type algorithms E-Kmeans, E-Wkmeans, E-AWA by integrating both intracluster compactness and intercluster separation are given. Kmeans type algorithms are improved by integrating these techniques and new objective functions based Fuzzy Compactness and Separation is proposed.

The extending algorithms are able to produce better clustering results in comparison to other algorithms. Therefore Fuzzy Compactness and Separation (FCS) deliver the best performance in comparison to other algorithms.

VII. Future Scope

As a future work, proposed Kmeans algorithm can be used for applications like gene data clustering and image processing. The dataset used for proposed algorithm is real dataset. In the future, we can use categorical dataset for Kmeans clustering.

References

1. Xiaohui Huang, Yunming Ye, and Haijun Zhang "Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation", IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 25, NO. 8, AUGUST 2014.

2. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. San Mateo, CA, USA: Morgan Kaufmann, 2011.

3. S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," IEEE Trans. Syst., Man Cybern., A, Syst. Humans, vol. 38, no. 1,pp. 218–237, Jan. 2008.

4. W. De Sarbo, J. Carroll, L. Clark, and P. Green, "Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables," Psychometrika, vol. 49, no. 1, pp. 57–78, 1984.

5. G. De Soete, "Optimal variable weighting for ultrametric and additive tree clustering," Qual. Quantity, vol. 20, no. 2, pp. 169–180, 1986.

6. M. Al-Razgan and C. Domeniconi, "Weighted clustering ensembles," in Proc. SIAM Int.Conf. Data Mining, 2006, pp. 258–269.

7. X. Chen, Y. Ye, X. Xu, and J. Zhexue Huang, "A feature group weighting method for subspace clustering of high-dimensional data," Pattern Recognit., vol. 45, no. 1, pp. 434–446, 012.

8. A. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, 2010.

9. R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Trans. Neural Netw., vol.16, no. 3, pp. 645–678, May 2005.

10. M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in Proc. KDD Workshop Text Mining, vol. 400. 2000, pp. 525–526

11. D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in Proc. 17th Int. Conf. Mach. Learn., San Francisco, CA, USA, 2000, pp. 727–734.

12. K. Wu, J. Yu, and M. Yang, "A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests," Pattern Recognit. Lett. vol. 26, no. 5, pp. 639–652, 2005.

13. C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park, "Fast algorithms for Projected clustering," ACM SIGMOD Rec., vol. 28, no. 2, pp. 61–72, 1999.

14. J. Friedman and J. Meulman, "Clustering objects on subsets of attributes," J. R. Statist. Soc., Ser. B, vol. 66, no. 4, pp. 815–849, 2004.

AUTHOR(S) PROFILE

Amruta S. Suryavanshi, is currently pursuing M.E (Computer) from Department of Computer Engineering, Bhivarabai Sawant College of Engineering and research, Savitribai Phule Pune University, Pune, Maharashtra, India -411007. She received her B.E (Information Technology) Degree from D. Y. Patil college of Engineering and Technology, Shivaji University, Kolhapur, Maharashtra, India -416113. Her area of interest is Data Mining.

Anil D. Gujar, received the M.Tech. (IT) degree from the Department of Information Technology, Bharati Vidyapeeth University, Pune, Maharashtra, India. He is currently working as Asst. Professor with Department of Computer Engineering, Bhivarabai Sawant College of Engineering and Research, Pune, Maharashtra, India.