IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-6, Issue-2 E-ISSN: 2347-2693

Balanced Data Clustering Algorithm for Both Hard and Soft Clustering

Purnendu Das[1], Bishwa Ranjan Roy[2*], Saptarshi Paul[3]

[1]Dept. of Computer Science, Assam University, Silchar, India
[2*]Dept. of Computer Science, Assam University, Silchar, India
[3]Dept. of Computer Science, Assam University, Silchar, India

**Corresponding Author: brroy88@gmail.com, Tel.: +91-94356-73134*

Available online at: www.ijcseonline.org

Received: 20/Jan/2018, Revised: 30/Jan/2018, Accepted: 14/Feb/2018, Published: 28/Feb/2018

***Abstract*—** Clustering is a widely studied problem in a variety of application domains such as neural network and statistics. It is the process of partitioning or grouping a set of patterns into disjoint clusters which show that patterns belonging to the same cluster are same or alike and patterns in different cluster are different. There are many ways to deal with the above problem of clustering. *K*-means is the simple and effective algorithm in producing good clustering results for many practical applications. However, they are sensitive to the choice of starting points and are inefficient for solving clustering problems in large datasets. Recently, incremental approaches have been developed to resolve difficulties with the choice of starting points. The global *k*-means and the fast global *k*-means algorithms are based on such an approach. They iteratively add one cluster center at a time. Fuzzy C- means is also very popular for fuzzy based data clustering. But all such clustering algorithms are hugely effected by the imbalanced nature of data values. Each data in the dataset has multiple attributes and the value of some attributes may be so large that the importance of other attributes values may be completely ignored during the clustering process. In this paper we proposed a data balancing technique for both fast global *k*-means and fuzzy c-means algorithm. We balanced the attributes values of each data in such a way that all the attributes get importance during the clustering process.

***Keywords*—** k-Means, Global k-Means, Fast Global k-Means, Data Streaming

I. INTRODUCTION

A fundamental problem that frequently arises in a great variety of fields such as pattern recognition, image processing, machine learning and statistics is the clustering problem [1], [2], [3], [4]. In its basic form the clustering problem is defined as the problem of finding homogeneous groups of data points in a given data set. Each of these groups is called a cluster and can be defined as a region in which the density of objects is locally higher than in other regions.

The simplest form of clustering is partitional clustering which aims at partitioning a given data set into disjoint subsets (clusters) so that specific clustering criteria are optimized. The most widely used criterion is the clustering error criterion which for each point computes its squared distance from the corresponding cluster center and then takes the sum of these distances for all points in the data set. A popular clustering method that minimizes the clustering error is the *k*-means algorithm. However, the *k*-means algorithm is a local search procedure and it is well known that it suffers from the serious drawback that its performance heavily depends on the initial starting conditions [1].

Different approaches to improve the efficiency of the *k*-means algorithm have been proposed [2], of which incremental ones are among the most successful. In these approaches clusters are computed incrementally by solving all intermediate clustering problems. The global *k*-means algorithm (GKM) proposed in [5] and the modified global *k*-means algorithm (FSGK) proposed in [6] are incremental clustering algorithms. Results of numerical experiments presented in [6] show that these algorithms allow one to find global or a near global minimizer of the cluster (or error) function.

Global *k*-means clustering algorithm (GKM), which constitutes a deterministic effective global clustering algorithm for the minimization of the clustering error that employs the *k*-means algorithm as a local search procedure. The algorithm proceeds in an incremental way: to solve a clustering problem with M clusters, all intermediate problems with 1, 2, . . . , M 1 clusters are sequentially solved. The basic idea underlying the proposed method is that an optimal solution for a clustering problem with M clusters can be obtained using a series of local searches (using the k-means algorithm). At each local search the M 1 cluster centers are always initially placed at their optimal positions corresponding to the clustering problem with M 1 clusters. The remaining M^{th} cluster center is initially placed at several positions within the data space. Since for $M = 1$ the optimal solution is known, we can iteratively apply the above procedure to 2nd optimal solutions for all k-clustering problems $k = 1, . . . , M$. In addition to effectiveness, the method is deterministic and does not depend on any initial conditions or empirically adjustable parameters. These are significant advantages over all clustering approaches mentioned above.

A new version of the modified global k-means algorithm (FSGK) is proposed in [6]. An auxiliary cluster function has been applied to generate a set of starting points lying in different parts of the dataset. The k-means algorithm is applied starting from these points to minimize the auxiliary cluster function and the best solution is selected as a starting point for the next cluster center. Exploit the information gathered in previous iterations of the incremental algorithm to avoid computing the whole affinity matrix. Also the tri-angle inequality for distances is used to avoid unnecessary computations. The results demonstrate that the FSGK is far more efficient than the GKM.

In hard clustering like K-Means, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters. One of most popular fuzzy clustering algorithm is fuzzy C- means algorithm.

In a dataset, where each data is a vector of having n attributes. It may be possible that some attributes of each data are so large that the clustering algorithm ignores the other attributes having lesser value. Ignoring some attribute values results incorrect clustering, which is a major problem faced by all standard clustering algorithm like k-means, global k-means, fast global k-means and fuzzy c means. This paper we proposed an data balancing technique to balance each data in the dataset so that each attribute gets equal importance during the data clustering process. The technique is already implemented for k-means [7], [8]. We used the technique for fast global k-means and fuzzy c-means.

The rest of the paper is organized as follows. Section II gives the background details required for this paper. We explained our proposed algorithm in section III. The experimental comparisons and analysis are given in section IV. Finally we conclude the paper in section VI

II. BACKGROUND

A. Global k-means algorithm (GKM)

Given a data set $D = \{d_1, . . . , d_N \}$, $d_n \in R^d$, the R- clustering problem aims at partitioning this data set into R disjoint subsets (clusters) $L_1, . . . , L_R$, such that a clustering criterion is optimized. The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point d_i and the centroid c_k (cluster center) of the subset L_k which contains d_i. This criterion is called clustering error and depends on the cluster centers $c_1, . . . , c_R$:

$$F\ (c1,...,cN\ ) = \sum_{i=j}^{N}\sum_{k=1}^{R} B(di\ \in\ Lk)||di\ -\ ck||^2, \quad (1)$$

Where $B(X) = 1$ if D is true and 0 otherwise.

The global k-means clustering algorithm constitutes a deterministic global optimization method that does not depend on any initial parameter values and employs the k-means algorithm as a local search procedure. Instead of randomly selecting initial values for all cluster centers as is the case with most global clustering algorithms, global k-means proceeds in an incremental way attempting to optimally add one new cluster center at each stage. More specifically, to solve a clustering problem with R clusters the method proceeds as follows. Start with one cluster ($r = 1$) and find its optimal position which corresponds to the centroid of the data set D. In order to solve the problem with two clusters ($r = 2$), perform N executions of the k-means algorithm from the following initial positions of the cluster centers: the first cluster center is always placed at the optimal position for the problem with $r = 1$, while the second center at execution n is placed at the position of the data point d_n, ($n = 1, . . . , N$). The best solution obtained after the N executions of the k-means algorithm is considered as the solution for the clustering problem with $r = 2$. In general, let $\left(c_1^*(k), ..., c_N^*(k)\right)$ denote the final solution for k-clustering problem. Once the solution for the (k1)-clustering problem is found, the solution of the k-clustering problem is as follows: Perform N runs of the k-means algorithm with k clusters where each run n starts from the initial state$(c_1^*(k1), ..., c_{k-1}^*(k1), d_n)$). The best solution obtained from the N runs is considered as the solution $\left(c_1^*(k), ..., c_k^*(k)\right)$ of the k-clustering problem. The above algorithm finally obtain a solution with *R* clusters having also found solutions for all *k*-clustering problems with $k < R$.

B. The fast global k-means algorithm (FSGK)

To make the execution of global *k*-means algorithm faster a modified global *k*-means algorithm called fast global *k*-means (FSGK) has been proposed in [1]. In this algorithm, during each iteration of the incremental procedure, instead of executing *k*-means for all the data variables in the data set and decide the next cluster, it selects a single data from the entire data set as the initial center for the next cluster and continue with *k*-means algorithm. The selection of the single data from the data set is done by the following procedure. In order to compute an initial center, define x_i for each object d_i as following:

$$x_i = \sum_{j=1}^{N} \frac{dist_{ij}}{\sum_{l=1}^{N} dist_{jl}}, i = 1, \dots \dots, N \qquad (2)$$

The point that minimizes x_i is the one which has a com with the minimum x_i tends to be the best center of a cluster. Another parameter is required to obtain the next initial cluster center. Suppose that the solution of the $(k-1)$-clustering problem is $\left(c_1^*(k-1), \dots, c_{k-1}^*(k-1)\right)$ and a new cluster center (i.e., the k^{th} initial center) is added at the location d_i that minimizes l_i as defined in Equation 3. Then we execute the *K*-means algorithm to obtain the solution with *k* clusters.

$$l_i = \frac{x_i}{\sum_{j=1}^{k-1} dist\left(d_i, c_j{}^{(k-1)}\right)}, i = 1, \dots \dots, n. \qquad (3)$$

The addition of the parameter (i.e. the denominator of f_i) ensures that the new cluster center could be far away from the existing cluster centers. It should be noted that the new center we computed it by Equation 3 is an optimal initial cluster center.

The algorithm can be described as follows:

1) (Initialization) Calculate the distance between each pair of all the objects based on Euclidean distance, then calculate x_i for each object as defined in Equation2. Select the point that minimize x_i as the first center.

   Set r = 1.

2) (Update centroids) Apply k-means algorithm and pre serve the best r-partition obtained and their cluster centers ($c_1, c_2, \dots, c_r$).

3) (Stopping criterion) Set r = r+1. If r > R, then stop.

4) (Select the new cluster center) Calculate l_i for object d_i as defined in Equation 3. Select the point which has the minimum value of l_i as the new cluster center, now the initial center is ($c_1, c_2, \dots, c_r, d_i$) and go to Step2.

This version of the GKM algorithm has an excellent feature that it requires much less calculation amount and shows less computational complexity. The distance between each pair of objects is computed only once, which contributes to the excellent feature. At the same time, the selection of the next cluster initial center can avoid the impact of noisy data on the clustering result. This proposed algorithm will be compared with GKM algorithm and its variation in the next section.

C. FCM algorithm

Consider a data set $D = d_1, d_2, d_3, \dots, d_n$, the FCM algorithm partitions *D* into *M* fuzzy clusters and find out each clusters center so that the cost function (objective function) of dissimilarity measure is minimization or below a certain threshold. FCM analyze membership value of each data in each cluster, it is presented as follows:

Objective function:

$$J_m(U, c) = \sum_{k=1}^{n} \sum_{i=1}^{M} (u_{ik})^m (dist_{ik})^2 \qquad (4)$$

U and *v* can be calculated as:

$$u_{ik} = \frac{1}{\sum_{j=1}^{M} \left(\frac{dist_{ik}}{dist_{jk}}\right)^{2/(m-1)}} \qquad (5)$$

$$c_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \qquad (6)$$

Where u_{ik} is the membership value of the k^{th} data x_k in the i^{th} cluster. $dist_{ik} = \|d_k - c_i\|$ is the Euclidean distance between data d_k and the cluster centroid $ci, 1 \le i \le M$, $1 \le k \le n$, exponent $m > 1$.

The FCM algorithm determines the cluster centroid c_i and the membership matrix *U* through iterations using the following steps:

1. Initialize the membership matrix *U* , u_{ik} randomly comes from (0, 1) and satisfy:

$$\sum_{i=1}^{c}(u_{ik}) = 1, 1 \leq k \leq n$$

2. Calculate M fuzzy clusters c_i, $i = 1, \ldots, M$ using Equation 6.
3. Compute the objective function according to Equation 4. Stop if objective function of dissimilarity measure is minimization or concentrate on a certain value, or its improvement over previous iteration is below a certain threshold, or iterations reach a certain tolerance value.
4. Compute a new U using Equation 5. Go to step 2.

III. PROPOSED DATA BALANCING TECHNIQUE

In this section, we present our algorithm. And we start from the disadvantage of standard *k*-means. I think many authors meet the problem as Fig 1 shows when we use *K*- means as clustering algorithm. In a dataset, some values of features is so large while the others is so small. If we use *K*-means as clustering algorithm, the large value will play an important role in the clustering results while the small values can be ignored which is the disadvantage of *k*-means. If we use adaptive *k*- means algorithm, it is difficult that we find the weights and the complexity is also very high. For solving this problem, we think that we project all the values of features to a fixed rang of from 0 to 1. Or we normalize all the values. So we can solve the problem. The idea is so simple but it is effective. Transforming formula is as follow:

$$values(t) = \frac{O(values) - MIN(f)}{MAX(f) - MIN(f)} \pm \sigma$$

Where $values\ (t) \in [0,\ 1]$, f are feature values and σ is smoothing value if we want to use it.

In [1] the ides has been implemented for standard *K*- means algorithm only. In this paper we extended the work and proposed modified algorithm for both fast global *K* means and fuzzy C-means, using the concept of the above mentioned data balancing. The advantage of global *K*-means over standard *k*-means and fast global *k*-means over global *k*-means is already described in Section I. The proposed balanced fast global *k*-mean and balanced fuzzy C means are given next.

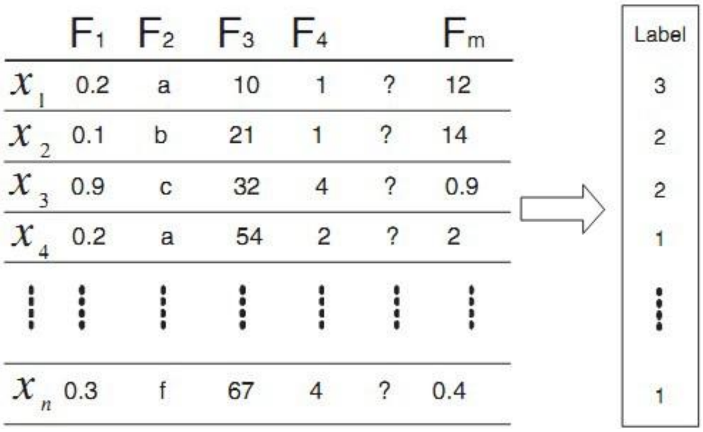
The algorithm can be described as follows:

1) For each data d in the dataset D, where d has n number of attributes, do the following:

$$d_i = \frac{d_i - MIN(d_i)}{MAX(d_i) - MIN(d_i)}$$

Where, $MIN\ (d_i)$ means the minimum attribute of d_i and $MAX\ (d_i)$ means the maximum attribute of d_i.

2) Run the Fast Global K-means algorithm or Fuzzy C-means algorithm with the modified (balanced) dataset.



	F_1	F_2	F_3	F_4		F_m	Label
x_1	0.2	a	10	1	?	12	3
x_2	0.1	b	21	1	?	14	2
x_3	0.9	c	32	4	?	0.9	2
x_4	0.2	a	54	2	?	2	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_n	0.3	f	67	4	?	0.4	1

Figure 1. The values of features is not balance, the values of F2 is so large while the values of F1 is so small

IV. EXPERIMENTAL ANALYSIS

In this section, we run experiments on seven datasets from UCI machine learning repository. The numbers of objects, features and classes in each data set are listed in Figure 2. For evaluation, we use micro precision to measure accuracy of the cluster with respect to the true labels: the micro precision $M\ P\ = \sum_{h=1}^{k} ah/n$, where k is the number of clusters and n is the number of objects, denotes the number clusters and n of objects in cluster h that are correctly assigned to the corresponding class. We identify the "corresponding class" for consensus cluster h as the true class with the largest overlap with the cluster, and assign all objects in cluster h to that class. Note that $0 \leq M\ P \leq\ 1$, with 1 indicating the best possible clustering, which has to be in full agreement with the class labels. The results of experiment are showed next where the Maximum and average MP on different data- sets by running different cluster algorithms are listed. For fuzzy C-means we consider a data belongs to a cluster if the corresponding membership value is maximum.

We compare both Fast Global K-Means and Fuzzy C- Means with our proposed data balancing technique. We ran each algorithm for different cluster sizes. The results are compared in terms of MP and the time consumed.

A. *Comparing Fast Global K-means with our proposed technique*

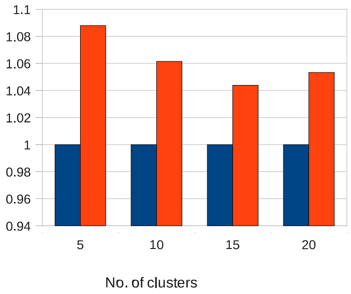


Figure 3. Comparing the average execution time (for all the datasets) of the Balanced Fast K-means algorithm (proposed) with original fast global k-means algorithm (Baseline)

We compare our proposed data balancing technique with original Fast Global K-means algorithm. Table I, II, III and IV shows comparison in terms of MP as well as the execution time for different number of clusters. In each Table we can see that the execution time is improving in all cases. The MP on the other hand not showing improvement in all cases but average improvement is more than 13%. Figure 3 and Figure 4 shows the improvement of our proposed technique over fast global k-means. Figure 3 shows the execution time improvement while Figure 4 shows the improvement in MP. Both figure takes the corresponding average value of all the datasets for a particular cluster size.

B. *Comparing Fuzzy C-means with our proposed technique*

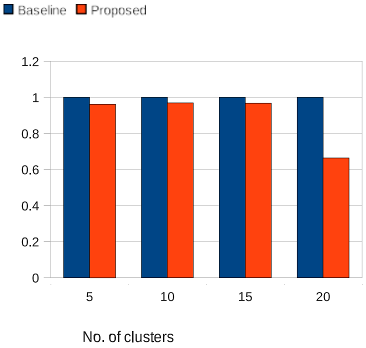
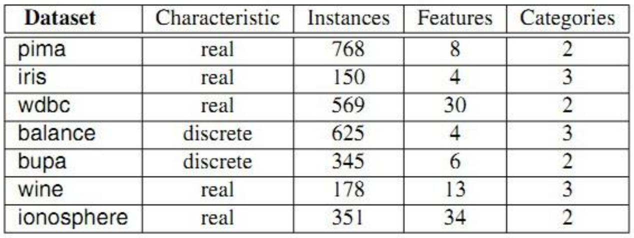


Figure 4. Comparing the average MP (for all the datasets) of the Balanced Fast K-means algorithm (proposed) with original fast global k-means algorithm (baseline).

We compare our proposed data balancing technique with original Fuzzy C-means algorithm. Table V, VI, VII and VIII shows comparison in terms of MP as well as the execution time for different number of clusters. In each Table we can see that the execution time is improving in all cases. The MP on the other hand not showing improvement in all cases but average improvement is more than 5%. Figure 5 and Figure 6 shows the improvement of our proposed technique over fast global k-means. Figure 5 shows the execution time improvement while Figure 6 shows the improvement in MP. Both figure takes the corresponding average value of all the datasets for a particular cluster size.



Dataset	Characteristic	Instances	Features	Categories
pima	real	768	8	2
iris	real	150	4	3
wdbc	real	569	30	2
balance	discrete	625	4	3
bupa	discrete	345	6	2
wine	real	178	13	3
ionosphere	real	351	34	2

Figure 2. THE NUMBER OF THE INSTANCES, FEATURES, AND CLASSES IN EACH DATASET.

Table IComparing Blanaced Fast Global K-means (Proposed) and Fast Global K-means (Baseline) in terms of max MP and time consumed. Number of cluster is 5.

DATASETS			Baseline			
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.2171	295	0.1767	355	22.86	16.90
wine	0.0617	111	0.0449	121	37.42	8.26
pima	0.6223	3258	0.65	3370	-4.26	3.32
iris	1	42	0.83	43	20.48	2.33
bupa	0.59	314	0.57	328	3.51	4.27
balance	0.198	1151	0.2	1160	-1.00	0.78
Average	**0.448183**	**861.8333333**	**0.411933**	**896.1666667**	**13.17**	**5.98**

Table II Comparing Blanaced Fast Global K-means (Proposed) and Fast Global K-means (Baseline) in terms of max MP and time consumed. Number of cluster is 10.

DATASETS			Baseline			
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.308	546	0.2525	622	21.98019802	12.21864952
wine	0.11	198	0.06	215	83.33333333	7.906976744
pima	0.62	5985	0.65	6157	-4.615384615	2.793568296
iris	1	65	0.93	68	7.52688172	4.411764706
bupa	0.59	575	0.57	592	3.50877193	2.871621622
balance	0.198	2097	0.2	2118	-1	0.991501416
Average	**0.471**	**1577.666667**	**0.44375**	**1628.666667**	**18.4556334**	**5.199013717**

Table III Comparing Blanaced Fast Global K-means (Proposed) and Fast Global K-means (Baseline) in terms of max MP and time consumed. Number of cluster is 15.

DATASETS			Baseline		Proposed	
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wine	0.1741	284	0.089	309	95.61797753	8.090614887
pima	0.6223	8712	0.651	8975	-4.408602151	2.930362117
iris	1	89	0.94	93	6.382978723	4.301075269
bupa	0.5913	840	0.5797	881	2.001035018	4.653802497
balance	0.198	3063	0.2175	3174	-8.965517241	3.497164461
Average	**0.51714**	**2597.6**	**0.49544**	**2686.4**	**18.12557438**	**4.694603846**

Table IV Comparing Blanaced Fast Global K-means (Proposed) and Fast Global K-means (Baseline) in terms of max MP and time consumed. Number of cluster is 20.

DATASETS			Baseline		Proposed	
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wine	0.2303	373	0.1235	398	86.47773279	6.281407035
pima	0.6223	5985	0.65	11764	-4.261538462	49.12444747
iris	1	114	0.94	119	6.382978723	4.201680672
bupa	0.59	1110	0.57	1156	3.50877193	3.979238754
balance	0.2	3996	0.2255	4017	-11.30820399	0.522778193
Average	**0.52852**	**2315.6**	**0.5018**	**3490.8**	**16.1599482**	**12.82191042**

Table V Comparing Blanaced Fuzzy C-means (Proposed) and Fuzzy C-means (Baseline) in terms of max MP and time consumed. Number of cluster is 5.

DATASETS			Baseline		Proposed	
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.1	23	0.17	420	-41.18	94.52
wine	0.067	15	0.033	191	103.03	92.15
pima	0.62	625	0.65	1185	-4.62	47.26
iris	1	34	0.89	80	12.36	57.50
bupa	0.59	135	0.57	891	3.51	84.85
balance	0.1984	34	0.19	237	4.42	85.65
Average	**0.429233**	**144.3333333**	**0.417167**	**500.6666667**	**12.92**	**76.99**

Table VI Comparing Blanaced Fuzzy C-means (Proposed) and Fuzzy C-means (Baseline) in terms of max MP and time consumed. Number of cluster is 10.

DATASETS	Proposed		Baseline			
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.18	31	0.2	2355	-10.00	98.68
wine	0.08	37	0.06	501	33.33	92.61
pima	0.62	809	0.65	3614	-4.62	77.61
iris	1	62	0.92	143	8.70	56.64
bupa	0.59	70	0.57	1860	3.51	96.24
balance	0.1984	83	0.2	562	-0.80	85.23
Average	**0.444733**	**182**	**0.433333**	**1505.833333**	**5.02**	**84.50**

Table VII Comparing Blanaced Fuzzy C-means (Proposed) and Fuzzy C-means (Baseline) in terms of max MP and time consumed. Number of cluster is 15.

DATASETS	Proposed		Baseline			
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.15	55	0.25	2708	-40.00	97.97
wine	0.16	34	0.09	733	77.78	95.36
pima	0.62	1063	0.65	8960	-4.62	88.14
iris	1	44	0.96	210	4.17	79.05
bupa	0.59	91	0.58	2418	1.72	96.24
balance	0.1984	78	0.2	955	-0.80	91.83
Average	**0.453067**	**227.5**	**0.455**	**2664**	**6.38**	**91.43**

Table VIII Comparing Blanaced Fuzzy C-means (Proposed) and Fuzzy C-means (Baseline) in terms of max MP and time consumed. Number of cluster is 20.

DATASETS	Proposed		Baseline			
		Time Consumed	Max MP	Time Consumed	Improvement in MP (in %)	Improvement in Time (in %)
wpbc	0.29	82	0.31	6826	-6.45	98.80
wine	0.13	84	0.11	840	18.18	90.00
pima	0.62	322	0.65	9919	-4.62	96.75
iris	1	62	0.95	346	5.26	82.08
bupa	0.62	144	0.58	4746	6.90	96.97
balance	0.2	123	0.2	1311	0.00	90.62
Average	**0.476667**	**136.1666667**	**0.466667**	**3998**	**3.21**	**92.54**

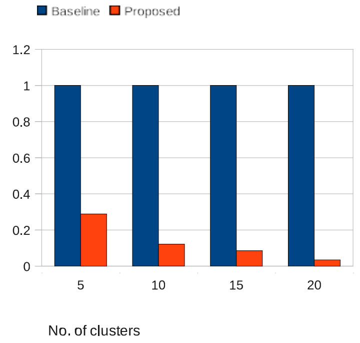


Figure 5. Comparing the average execution time (for all the datasets) of the Balanced Fuzzy C-means algorithm (proposed) with original fuzzy C-means algorithm (baseline).

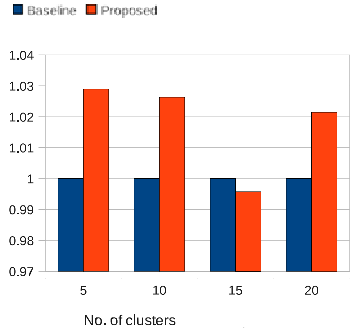


Figure 6. Comparing the average MP (for all the datasets) of the Balanced Fuzzy C-means algorithm (proposed) with original fuzzy C-means algorithm (baseline).

VII. Future Work

Data balancing of global Fuzzy C-means is an possible extension of the proposed work. Also the work can be extend for high dimensional datasets. The streaming dataset need to handle separately in clustering. Balancing the streaming datasets is considered as future work.

VIII. Conclusion

Clustering is a widely studied problem in a variety of application domains such as neural network and statistics. It is the process of partitioning or grouping a set of patterns into disjoint clusters which show that patterns belonging to the same cluster are same or alike and patterns in different cluster are different. There are many ways to deal with the above problem of clustering. *K*-means is the simple and effective algorithm in producing good clustering results for many practical applications. However, they are sensitive to the choice of starting points and are inefficient for solving clustering problems in large datasets. Recently, incremental approaches have been developed to resolve difficulties with the choice of starting points. The global *k*-means and the fast global *k*-means algorithms are based on such an approach. They iteratively add one cluster center at a time. Fuzzy C-means is also very popular for fuzzy based data clustering. But all such clustering algorithms are hugely effected by the imbalanced nature of data values. Each data in the dataset has multiple attributes and the value of some attributes may be so large that the importance of other attributes values may be completely ignored during the clustering process.

In this paper we proposed an data balancing technique for both fast global *k*-means and fuzzy c-means algorithm. We balanced the attributes values of each data in such a way that all the attributes get importance during the clustering process.

REFERENCES

[1] L. Bai, J. Liang, C. Sui, and C. Dang, “Fast global k-means clustering based on local geometrical information,” *Information Sciences*, vol. 245, no. 0, pp. 168 – 180, 2013.

[2] A. Jain and R. Dubes, Eds., *Algorithms for Clustering Data*. Prentice Hall, 1988.

[3] R. Wan, X. Yan, and X. Su, “A weighted fuzzy clustering algo rithm for data stream,” in *Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management - Volume 01*, ser. CCCM ’08, 2008, pp. 360–364.

[4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS ’02, 2002, pp. 1–16.

[5] A. Likas, M. Vlassis, and J. Verbeek, “The global k-means clustering algorithm,” *Pattern Recognition*, vol. 35, no. 2, pp. 451–461, 2003.

[6] A. Bagirov, “Modified global k-means algorithm for sum-of-squares clustering problem,” *Pattern Recognition*, vol. 41, pp. 3192–3199, 2008.

[7] H. Wang, J. Qi, W. Zheng, and M. Wang, “Balance k-means algorithm,” in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, Dec 2009, pp. 1–3.

[8] R. He, W. Xu, J. Sun, and B. Zu, “Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem,” in *Proceedings of the 2009 Third International Symposium on Intelligent Information Technology Application - Volume 03*, ser. IITA ’09. IEEE Computer Society, 2009, pp. 87–90. [Online]. Available: http://dx.doi.org/10.1109/IITA.2009.307

Authors Profile

Dr. Purnendu Das is an assistant professor of the Department of Computer Science, Assam University, Silchar. He has pursued Ph.D. degree from Tripura University. He has published researched papers in many reputed journals.

Bishwa Ranjan Roy is an assistant professor of the Department of Computer Science, Assam University, Silchar. He has pursued M.Tech. degree from NIT Silchar. He has published researched papers in many reputed journals.

Saptarshi Paul is an assistant professor of the Department of Computer Science, Assam University, Silchar. He has pursued M.Tech. degree from VIT University. He has published researched papers in many reputed journals.