Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 167 (2020) 113–120

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

K-RMS Algorithm

Avishek Garain*, Dipankar Das

Computer Science and Engineering, Jadavpur University, Kolkata, India

Abstract

Clustering is an unsupervised learning problem in the domain of machine learning and data science, where information about data instances may or may not be given. K-Means algorithm is one such clustering algorithms, the use of which is widespread. But, at the same time K-Means suffers from a few disadvantages such as low accuracy and high number of iterations. In order to rectify such problems, a modified K-Means algorithm has been demonstrated , named as K-RMS clustering algorithm in the present work. The modifications have been done so that the accuracy increases albeit with less number of iterations and specially performs well for decimal data compared to K-Means. The modified algorithm has been tested on 12 datasets obtained from UCI web archive, and the results gathered are very promising.

Keywords: clustering;distortion-error;rms-value;multi-component analysis; unsupervised-learning

1. Introduction

Clustering can be loosely defined as the process of organizing objects into groups whose members are similar, in some way. A cluster, therefore, is a collection of objects which are similar amongst them and are dissimilar with respect to objects belonging to other clusters. Clustering algorithms are generally based on Unsupervised Learning technique. The basic objective of cluster analysis is to discover natural groupings of objects.

A clustering algorithm tackles datasets containing many features scaled down to 2-D or 3-D data with various types of preprocessors. Labels are generally absent and only information that might be present are, the type of data and the number of classes it contains. In most cases this data is unavailable too. For such instances, some other methods are used to find this piece of information.

Cluster analysis techniques have been used in many areas such as qualitative interpretation and data compression, process monitoring, local model development, analysis of chemical compounds for combinatorial chemistry, discov-

* Avishek Garain. Tel.: +91-8436154826
E-mail address: avishekgarain@gmail.com

10.1016/j.procs.2020.03.188

ering of clusters in DNA dinucleotides, etc., to name a few. It has recently also found success in telecommunication network (calling patterns and fault management systems).

K-Means is one such efficient clustering algorithm that has been used in various disciplines like information retrieval and image analysis. But, it suffers from some disadvantages such as high number of iterations and low accuracy for signed data. This is due to the fact that, K-Means algorithm considers average value to calculate the centroid of a cluster.

To counter this, the K-Means algorithm has been modified, by changing the way clustering is achieved by allocating the centroid. The new algorithm has been named as K-RMS algorithm and the details of it will be discussed in Section 4. To test the effectiveness of the K-RMS algorithm, it has been tested on 12 datasets, obtained from sklearn[1] package and UCI[2].

The paper has been organized as follows. In section 2 a brief survey of the previous work done in this field has been provided. In section 3, a small description on the working of K-Means algorithm has been provided. In section 4, the proposed K-RMS algorithm and its various constraints have been discussed in details. The advantages of using this proposed algorithm over traditional K-means algorithm have also been discussed. Also, the datasets, obtained from sklearn package and UCI, that has been used to test the algorithm in section 5 have been described. Finally, the experimental results and visual comparison with the state of art has been shown in section 6. This is followed by concluding remarks in Section 7

2. Literature Survey

Atashpaz-Gargari and Lucas [2] proposed the ICA (Imperialist Competitive Algorithm) for optimization inspired by the imperialistic competition. The algorithm starts with an initial population called countries. The countries are of two types: colonies and imperialists and together they form some empires. Imperialistic competition among these empires form the basis of this algorithm. During this competition, weak empires collapse and powerful ones take possession and in turn converges to a state in which there exist only one empire. The imperialist empire and it's colonies have the same cost. The authors showed the ability of this algorithm in dealing with different types of optimization problems.

In order to improve the convergence velocity and accuracy of the ICA, Niknam et al. [8] recommended a modified imperialist competitive algorithm (MICA). They found that premature convergence may occur under different situations: the population converges to local optima of the objective function or the search algorithm proceeds slowly or does not proceed at all. The authors stated that mutation is a powerful strategy which diversifies the ICA population and improves the ICAs performance on preventing premature convergence to local minima. The authors used a new mutation operator.

Global Kernel K-Means algorithm was proposed by Tzortzis and Likas [11]. It mapped the data set points from input space to a higher dimensional feature space, with the help of a kernel matrix. Also, the authors used this algorithm to find the near-optimal solution to the clustering problem, by incrementally and deterministically adding a new cluster at each stage and by applying kernel k-means as a local search procedure instead of initializing all clusters at the beginning of the execution.

K-Modes algorithm was proposed by He et al. [4]. In the basic algorithm the total cost against the whole data set is calculated, each time when a new Mode is obtained. To make the computation more efficient the following algorithm instead is used in practice. k initial modes, one for each cluster.An object is allocated to the cluster whose mode is the nearest to it.The mode of the cluster is updated after each allocation. After all objects have been allocated to clusters, the dissimilarity of objects is retested against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, the object is reallocated to that cluster and the modes of both clusters are updated.This process is repeated until no object has changed clusters after a full cycle test of the whole data set.

[1] https://scikit-learn.org/stable/
[2] https://archive.ics.uci.edu/ml/datasets

3. K-Means Algorithm

The proposed algorithm K-RMS is based on the principle of K-Means algorithm which is a clustering algorithm. It starts working on k initial cluster centers and improves the accuracy of clustering [7] in iterative fashion. Initially, every point is allocated to some cluster, by finding minimum distance of that point from the centroid. Then centroid of allocated points is calculated using the following formula.

$$X = (X_1 + X_2 + X_n)/n$$
$$Y = (Y_1 + Y_2 + Y_n)/n$$

The centroids are then updated and the data points are reallocated accordingly. But, the disadvantage of this algorithm is the large number of iterations while handling large amount of data. Moreover, it also gives very less efficiency on predicting clusters in datasets containing instances with small decimal values. It also suffers from decreasing accuracy for signed data along with increasing iterations.

For example, let

$$X, Y = (4, 5), (6, 9), (-4, -5), (-6, -9)$$

Now let's allocate the points $(4, 5), (-4, -5)$ to 1^{st} centroid and $(6, 9), (-6, -9)$ to the 2^{nd} one.

$$(X_c, Y_c) = (0, 0)$$

In case of K-Means, this occurs for both the centroids.
Now, the fact is that the cancellation process continues and increases the number of iterations. In order to cope up with this disadvantage, the following algorithm named as K-RMS was proposed .

4. K-RMS Clustering Algorithm

The algorithm "K-RMS" is devised such that it solves issues like the handling signed data problem. It also decreases the number of iterations and increases the accuracy to a great extent.

4.1. Probable Advantages and Hypothesis

It is observed that if RMS(Root Mean Square) value is used instead of average value, it is expected that the number of iterations will decrease significantly for large datasets. This is because RMS value is much more exact and fast converging in every field of science be it chemistry(V_{RMS} or Root Mean Square Velocity) or some other fields like electrical circuits, etc. It also takes care of negative values in datasets. The degree of changes that takes place during the workflow of the algorithm is lesser compared to that when average value is used.

4.2. Algorithm

STEP-1: Let, there be n number of data points and the Feature1 of the dataset be denoted by set

$$\alpha = \{X_1, X_2, X_3....., X_n\}$$

and Feature2 of the dataset by set,

$$\Upsilon = \{Y_1, Y_2, Y_3.., Y_n\}$$

Initially, it randomly generates the number of centroids where the count is same as the number of clusters, the data is to be divided into. Let, the number of clusters be denoted by M and the centriods be denoted by ξ, where, $\xi = \{(X_1, Y_1), (X_2, Y_2), .., (X_M, Y_M)\}$ Now, these randomly generated centroids have random initial values. If a given point (x,y) is to be allocated to one of the centroids, the average euclidean distance of all given data points is calculated individually from each of the centroids and the distance be denoted by δ.
For example, with respect to a point (X_i, Y_i) and a centroid (X_j, Y_j), where $1 \leq i \leq n, 1 \leq j \leq M$.*Now*,$\delta$ between (X_i, Y_i) and (X_j, Y_j) is denoted by δ_{ij}.

$$\delta_{ij} = \sqrt{((X_i - X_j)^2 + (Y_i - Y_j)^2)/n}$$

The Euclidean distance has been normalized by the number of data points to bring it closer to the threshold that decides its convergence. It is observed that it significantly reduces the number of iterations without hampering the accuracy. Now, for each (X_i, Y_i) , $\min(\delta_{ij})$ gives the centroid (X_j, Y_j) to which the the point is allocated and same is applied for other datapoints and centroids.

STEP-2: The RMS value [10][1] of the points allocated to a centroid is found out respectively for x and y coordinates of those data points and this new value (X_{RMS}, Y_{RMS}) is assigned as the new centroid where,

$$X_{rms} = \sqrt{((X_1{}^2 + X_2{}^2 + ..X_n^2)/n)}$$

$$Y_{rms} = \sqrt{((Y_1{}^2 + Y_2{}^2 + ..Y_n{}^2)/n)}$$

This process continues for each of the initialized centroids.

STEP-3: Now, the distance between initial and final positions of each and every centroid is measured respectively. This distance is called as shift. A threshold for shift has already been defined. If the shift is less than that threshold, then no further iterations take place for that particular centroid.

STEP-4: Now, as a new set of centroids has been obtained, apply the part of **STEP-1** is applied again where the distance from all points is found and assigned accordingly. The maximum and minimum errors have been computed too. This is nothing but δ_{max} and δ_{min} found after some K number of iterations of the average squared distance between each point and its cluster centroid. This is also known as the distortion cost.This distortion cost is compared with threshold while going for the next iteration.

To portray the advantage of this algorithm over centroid allocation, let us look at the same example given in Section 3.

For example, let

$$X, Y = (4, 5), (6, 9), (-4, -5), (-6, -9)$$

Now let's allocate $(4, 5), (-4, -5)$ to 1^{st} centroid and $(6, 9), (-6, -9)$ to the 2^{nd} one.

$$(X_c, Y_c) = (4, 5)$$

for K-RMS This decreases distortion error computation that results in relatively faster convergence.

4.3. Precautions:

1. The algorithm may get stuck at some local minima so the whole process is iterated and the best of results are shown. A program for this purpose is specifically made too, which is to be used in determining k before running the main program.
2. Enough number of markers are allocated for any basic dataset to be displayed. Things to be manually entered with data are:

    - k-Number of clusters
    - Cutoff(Here 0.2 or .12)-Varies with dataset(to be determined by experiment)
    - Dimension of data(2 or 3) if requires visual analysis else can work with any number of dimensions

5. Datasets

The following datasets, from sklearn package were used to test the hypothesis.

1. Iris dataset-sklearn.datasets
2. Wine dataset-sklearn.datasets
3. Man-Woman shopping statistics dataset

Additionally, some more datasets were collected from UCI website https://archive.ics.uci.edu/ml/machine-learning-databases/. The datasets were:
R-15 dataset, D-31 dataset, Compound dataset, S-originals dataset, S-originals dataset1, S-originals dataset2, Parkinson-1 dataset, Parkinson-Control dataset, Path Based dataset, Frog dataset

The figures under subsection Graphs and Visual Analysis of results, namely Fig-1,2 describes the accuracy gained and iterations reduced over K-Means algorithm.The graphs following namely from Fig-3 till 7, visualize the proper clustering done by the K-RMS algorithm. A figure showing efficient and accurate clustering of multidimensional data has also been included, as in Fig-8.All the figures are generated using the plot.ly python library.

5.1. Description of Datasets

The Iris dataset (compiled by biologist Ronald Fisher), describes particular characteristics of Iris flowers, specifically, length and width of pedals and sepals. It has 4 features and has 3 classes.

The wine dataset has 13 features based on chemical constituents present in 3 types of wines in the same region of Italy.

Rest of the datasets are synthetic datasets obtained from UCI website, as mentioned above. The synthetic datasets have varying features and give rise to complex types of distribution on graphical plotting.

6. Results and Experiments

Results with subscript 1 show best accuracy enhancement and iteration reduction and subscript 2 shows same accuracy.On experiment best results obtained are displayed in the tables below. Features are selected accordingly.

DATASET	Sl. No.	FEATURES(X,Y)	ACCURACY(%)		ITERATIONS	
			K-Means	K-RMS	K-Means	K-RMS
IRIS (5000 epochs,cutoff=0.2)	1	(0,3)	94.60%	96.67%	9095	7505
	2	(1,3)	96.60%	97.34%	8491	63941
WINE (5000 epochs,cutoff=0.2)	1	(7,8)	51.12%	$83.15\%_1$	6097	5161
	2	(3,7)	67.97%	$91.57\%_1$	15648	128871
	3	(0,7)	91.01%	94.38%	7435	6206
R-15 (5000 epochs,cutoff=0.02)	1	(0,1)	97.83%	99.00%	20643	25282
COMPOUND (5000 epochs,cutoff=0.02)	1	(0,1)	55.13%	56.14%	45246	417401
D-31 (1000 epochs,cutoff=0.1)	1	(0,1)	48.90%	$94.99\%_1$	6522	28451
S-ORIGINALS (100 epochs,cutoff=0.1)	1	(0,1)	81%	$91\%_1$	1981	2019
S-ORIGINALS (1) (100 epochs,cutoff=0.1)	1	(0,1)	66%	$84\%_1$	2645	2439
PARKINSON-1 (1 epoch,cutoff=0.1)	1	(0,1)	68%	$76\%_1$	18	6
	2	(0,5)	82%	86%	16	15
	3	(1,3)	71.37%	$88.85\%_1$	15	15
	4	(1,4)	74.80%	$92.29\%_1$	31	6
	5	(3,4)	84.05%	$96.53\%_1$	28	9
PATH BASED (5000 epochs,cutoff=0.02)	1	(0,1)	48%	58%	48876	437461
PARKINSON-CONTROL (1 epoch ,cutoff=.1)	1	(0,1)	88%	89%	50	281
	2	(0,4)	76%	81%	36	201
	3	(1,5)	67%	$85\%_1$	5	21
	4	(3,4)	86%	$94\%_1$	10	41
	5	(3,5)	$98\%_2$	$98\%_2$	5	4

Table 1: Comparison of Accuracy and Iterations

DATASET	ELAPSED TIME(SECS)		ERROR			
	K-Means	K-RMS	K-Means		K-RMS	
			Lowest	Highest	Lowest	Highest
IRIS(5000 epochs,cutoff=0.2)	14.31	12.81	0.22	0.58	0.11	0.66
	11.84	10.695	0.14	0.71	0.07	0.41
WINE(5000 epochs,cutoff=0.2)	12.1	12.75	0.11	0.35	0.05	0.21
	22.61	23.74	2.31	4.79	1.16	2.57
	10.27	11.38	0.08	0.29	0.04	0.15
R-15(5000 epochs,cutoff=0.02)	471.506	607.271	0.18	2.73	0.09	1.48
COMPOUND (5000 epochs,cutoff=0.02)	280.41	289.16	9.69	27.06	4.88	13.85
D-31 (1000 epochs,cutoff=0.1)	553.43	231.09	1.13	6.85	0.56	2.28
S-ORIGINALS (100 epochs,cutoff=0.1)	322.81	373.79	1.78E+07	6.22E+07	8.94E+08	3.64E+09
S-ORIGINALS (1) (100 epochs,cutoff=0.1)	441.64	436.18	2.65E+08	7.38E+08	1.33E+09	3.44E+09
PARKINSON-1 (1 epoch,cutoff=0.1)	3.34	2.54	5992.48	5992.48	3086.09	3086.09
	2.12	1.47	4902.49	4902.49	2657.32	2657.32
	1.9	2.014	5.99E+07	5.99E+07	2.99E+07	2.99E+07
	2	2.23	3187.96	3187.96	1667.07	1667.07
	2.76	1.67	4493.22	4493.22	2296.38	2296.38
PATH BASED (5000 epochs,cutoff=0.02)	112.82	124.42	29.86	41.71	15.2	21.35
PARKINSON-CONTROL (1 epoch,cutoff=.1)	3.34	2.54	5992.48	5992.48	3086.09	3086.09
	2.78	2.05	13937.41	13937.41	7139.09	7139.09
	1.14	0.88	2.32E+07	2.32E+07	1.16E+07	1.16E+07
	1.21	1.12	18362.89	18362.89	9221.39	9221.39
	1.02	1.03	2.32E+07	2.32E+07	1.16E+07	1.16E+07

Table 2: Comparison of Time elapsed and mainly Distortion Error

When number of iterations is 1, the lowest and highest errors are same. This is due to the fact that the program is run only once and hence there can't be any variation of errors. The reasons due to such accuracies at some places is the distribution of data itself as can be seen in the next section. In the datasets such as D-31 dataset, R-15 dataset, S-originals dataset etc., the data points are clearly seperated in the X-Y plane and thus makes detection of clusters easier and efficient resulting in higher accuracies and lower number of iterations.

6.1. Graphs and Visual Analysis of Results

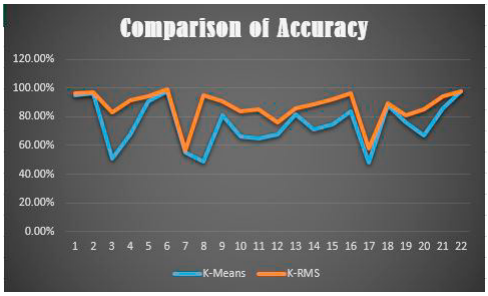

Fig. 1: Average Accuracy gained = 11.8%

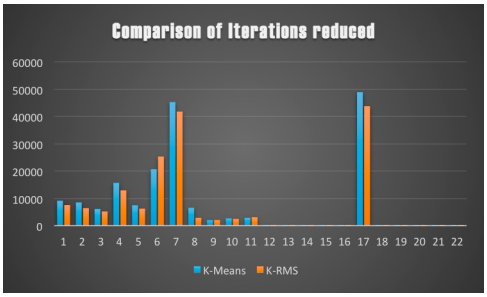

Fig. 2: Average Iteration Percentage reduced = 26.88%

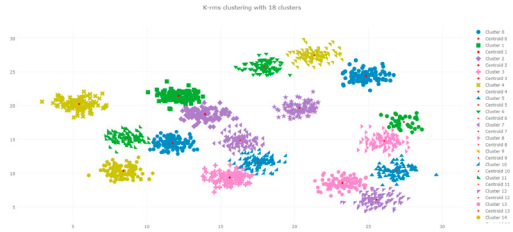

Fig. 3: D-31 Dataset Results

Fig. 4: R-15 Dataset Results

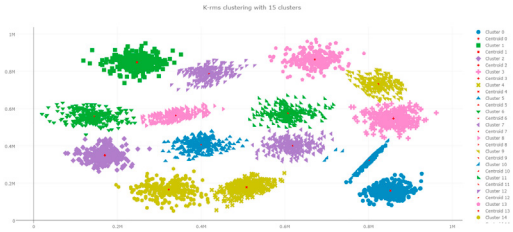

Fig. 5: S-Originals Dataset Results

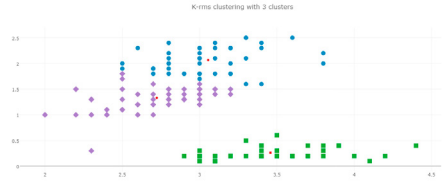

Fig. 6: Iris Dataset Results for features [1,2]

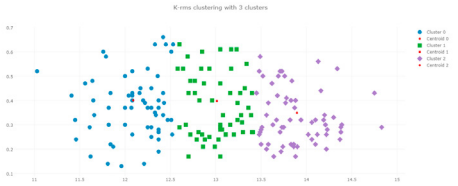

Fig. 7: Wine Dataset Results for features [0,7]

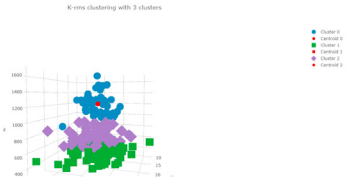

Fig. 8: Iris Dataset 3 Dimensions Results

7. Conclusion

During measuring distance, Manhattan distance was used but it took longer time and more iterations in contrast to Euclidian distance. It was observed that, Euclidean distance fits best with K-RMS as both make use of root-mean square values. Testing on Iris Dataset using three features(Sepal width, Petal length, Sepal length), as the dimensions, at once, the algorithm achieved 99.37% accuracy. While using 2 dimensions, maximum accuracy attained was 98%.The reason behind this improvement is the presence of more features at one time and the fact that the points are separated in 3-Dimensional space, makes it easier for the algorithm to differentiate between points as well as correctly specifying a cluster centroid to each of the data points. On testing a dataset of detection of various anuran calls of frogs, the accuracy was as high as 57%, by considering all of the feature dimensions. Earlier, based on 2 to 3 dimensions accuracy was as low as 3 to 4%. There can be more improvements based on this work. Manual pre-determination of clusters is a necessity of the present algorithm. However, work can be done to automate this process. A mechanism for determination of proper threshold may lead to increased efficiency of the algorithm. In future, mechanisms could be devised to systematically initialize the centroids much nearer to their corresponding clusters rather than a random initialization.

References

[1] , 2014. Copyright, in: Onajite, E. (Ed.), Seismic Data Analysis Techniques in Hydrocarbon Exploration. Elsevier, Oxford, p. iv. URL: http://www.sciencedirect.com/science/article/pii/B9780124200234099949, doi:https://doi.org/10.1016/B978-0-12-420023-4.09994-9.

[2] Atashpaz-Gargari, E., Lucas, C., 2007. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, pp. 4661–4667. doi:10.1109/CEC.2007.4425083.

[3] Bennett, K., Bradley, P., Demiriz, A., 2000. Constrained K-Means Clustering. Technical Report. URL: https://www.microsoft.com/en-us/research/publication/constrained-k-means-clustering/.

[4] He, Z., Huang, J.Z., Li, M.J., Ng, M.K., 2007. On the impact of dissimilarity measure in k-modes clustering algorithm. IEEE Transactions on Pattern Analysis Machine Intelligence 29, 503–507. URL: doi.ieeecomputersociety.org/10.1109/TPAMI.2007.53, doi:10.1109/TPAMI.2007.53.

[5] Huang, Z., 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2, 283–304. URL: https://doi.org/10.1023/A:1009769707641, doi:10.1023/A:1009769707641.

[6] Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[7] MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, University of California Press, Berkeley, Calif.. pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

[8] Niknam, T., Fard, E.T., Pourjafarian, N., Rousta, A., 2011a. An efficient hybrid algorithm based on modified imperialist competitive algorithm and k-means for data clustering. Engineering Applications of Artificial Intelligence 24, 306 – 317. URL: http://www.sciencedirect.com/science/article/pii/S0952197610001867, doi:https://doi.org/10.1016/j.engappai.2010.10.001.

[9] Niknam, T., Fard, E.T., Pourjafarian, N., Rousta, A., 2011b. An efficient hybrid algorithm based on modified imperialist competitive algorithm and k-means for data clustering. Engineering Applications of Artificial Intelligence 24, 306–317.

[10] Purcaru, D., Purcaru, I., Niculescu, E., 2006. Some methods for computing rms values and phase differences of currents and voltages, in: Proceedings of the 9th WSEAS International Conference on Applied Mathematics (MATH06), Turkey, pp. 587–591.

[11] Tzortzis, G.F., Likas, A.C., 2009. The global kernel k-means algorithm for clustering in feature space. IEEE Transactions on Neural Networks 20, 1181–1194. doi:10.1109/TNN.2009.2019722.