The Regress Averaging Clustering Algorithm

Terence Johnson[1], Santosh Kumar Singh[2], Valerie Menezes[3]

[1]PhD Scholar, AMET University, Chennai, India
[1]Asst. Prof, Dept. of Computer Engineering, Agnel Institute of Technology and Design, Goa, India

[2]Head, Dept. of Information Technology, Thakur College of Science and Commerce, Mumbai, India

[3]Asst. Prof, Dept. of Computer Engineering, Agnel Institute of Technology and Design, Goa, India
{ykterence@rediffmail.com, sksingh14@gmail.com, vmm@aitdgoa.edu.in}

***Abstract*-- The output of the K Means clustering process shows that the K-Means which are obtained finally, yield the mean of the dataset under consideration when averaged. In this novel approach to clustering, we use this revelation to find the average of the entire dataset and then depending on the number of required clusters we find the K unchanging means using Regress Averaging where-in we use the average of a dataset to find the points which yield the average.**

***Index Term*-- Regress Averaging, clustering, Euclidean distance measure**

1 INTRODUCTION

Clustering is the process of grouping together similar data into categories or groups called clusters. The main objective of clustering is to maximize intra-cluster similarity and minimize the inter-cluster similarity [1]. Clustering is one of the most important task in data mining and is applied for multiple purpose like scientific data investigation, information storage and recovery, text mining, spatial database and IT applications, Internet analysis, bio-medical analysis, market segmentation and much more [2].

1.1 Literature Review

Clustering is a conventional problem in the database, machine learning, artificial intelligence and theoretical literature and is defined as follows: Given a set of points in space, find a categorization of the points into groups called clusters so that the points within each cluster are similar to one another and dissimilar to points in other clusters [3]. The resemblance of points in a cluster is based on several score functions [4]. Broad-spectrum classes of methods in clustering include partitioning methods in which a set of representative points are used in order to segment the points implicitly [5]. Numerous variants of this technique exist such as the K-Means and K-Medoids algorithms [6]. The K-Means algorithm is a method of cluster analysis which targets to categorize n data observations into K clusters in which each observation is allocated to the cluster with the nearest mean. Given a set of preliminary means, each point is allocated to one of them, and then each previous cluster average is substituted by the mean of the respective cluster. These two steps are reiterated until convergence i.e. until the cluster means reappear in the succeeding stage or until the clusters resurface in the succeeding step. As the K-Means method in several instances takes an exponential time to converge, the clustering is forced to halt after a specified number of iterations. A point is allocated to the cluster which is closest in the Euclidean distance sense to the point. In medoid based methods, the points from the data repository are used as representative points as the algorithm searches for the best set of k representative points which result in optimum clustering. A pragmatic technique in this class called CLARANS [7], improves the efficiency of the algorithm by restricting the search space. In density-based clustering methods [8], the ϵ-neighbourhood of a point is used in order to find dense regions in which clusters exist. The BIRCH method [9] which uses a CF-Tree, in order to incrementally construct clusters is one of the most efficient techniques for low-dimensional data and it requires only one scan over the database. A method, called CURE [10] which stops the formation of a cluster hierarchy when that level contains a predefined number of clusters provides extremely high quality because it uses robust methods to measure distances between clusters and so can adjust well to different shapes of clusters. High-dimensional data poses a challenge to clustering algorithms as most of them do not work efficiently in higher dimensional spaces because of the inherent sparsity of the data [11]. This problem has been referred to as the curse of dimensionality [12]. In high dimensional space, the distance between every pair of points is almost the same for a variety of data distributions and distance functions. In such situations, even the meaning of closeness in high-dimensional data may be suspect. This issue can be resolved by using feature selection in order to reduce the dimensionality of the space [13] but it may not always be feasible to prune off too many dimensions without losing a substantial amount of information. There are various methodologies to dimension reduction like principal component analysis [14] and random projections [15] in clustering high dimensional data. In these cases, dimension reduction is performed as a pre-processing step and is alienated from the clustering process: once the new features are selected forming the subspace, they remain unchanged during the clustering process. Another approach is called subspace clustering [16] where the primary focus is on selecting a small number of dimensions from the original set of dimensions in an unsupervised way so that clusters form in this subspace. However, if we curtail the subspace to become linear combinations of the original dimensions [17], then, the subspace that is obtained using Fisher's linear discriminant

ratio [18] is perhaps the best subspace to perform clustering, because with this subspace, clusters are well separated.

1.2 Problem Area in K-Means Clustering

The motivation for the Regress Averaging Clustering Algorithm which is proposed in this paper comes from the output of the K-means clustering algorithm. The traditional K-Means algorithm works by arbitrarily choosing K cluster averages, putting each datum to the cluster whose average is nearest according to the Euclidean distance measure, then calculating the average of the set given to each cluster and using them as new averages in a loop until gathering of all points into the respective clusters they belong. The algorithmic running time of the K-Means technique is O(nktd) [19] with n being the total number of input set values, k being presented by the user as the total clusters into which the input is to be grouped, t representing the exact amount of repetitions the algorithm has to undergo in the event that the cluster averages are not the same in the next iteration or if the clusters are not the same in the next iteration and d involving the dimensions associated with each datum. If the cluster averages are not the same in the succeeding iteration or if the clusters are not the same in the succeeding step then the K-Means algorithm ends the clustering process tersely after the iterations t given by the user. This is likely to lead to the formation of inaccurate clusters. On numerous occasions, the K-Means method requires exponential time to find the averages and so, abruptly ending the clustering process after a number of specified repetitions is not likely to return precise clusters.

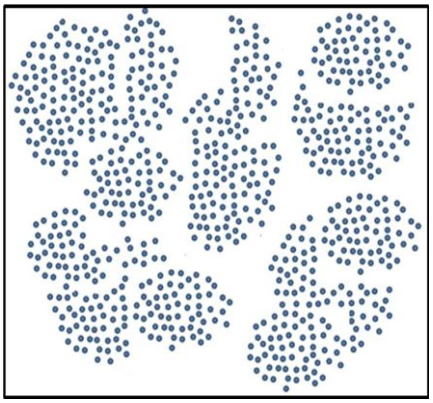

Fig. 1. Dataset to find 5 clusters

1.3 Resolution of the problem and significance of the proposed algorithm

This issue can be resolved if it is possible to pinpoint the eventual fixed K-Means for the K required clusters by skimming the input set at the commencement itself as no sooner these K means get traced, the job of clustering reduces to only allocating the left over data from the input into clusters, which are nearest to these eventual K fixed means based on the regular distance measures. In this paper we propose to solve this problem of the clustering having to end abruptly, by first finding the average of the entire dataset and then finding the K unchangeable means subject to the number of stated clusters using the average of the dataset to find the points which yield the average. It should be noted that the idea proposed in this paper is not based on the assumption that the mean of all data equals to the mean of the K mean points. On the contrary, the algorithm tries to locate K points equal to the number of required clusters which when averaged yields the same result as that when the entire dataset is averaged. The significance of the proposed algorithm is that it aims to eliminate from the K Means algorithm, the need for iteratively finding the final means around which all clusters form. This in turn eliminates the need for closing the clustering process abruptly using t which would have led to formation of inaccurate clusters.

2 Proposed Work

The K-Means method results in clusters with means that do not change and around which all other points in the dataset get clustered. This suggests that, if points

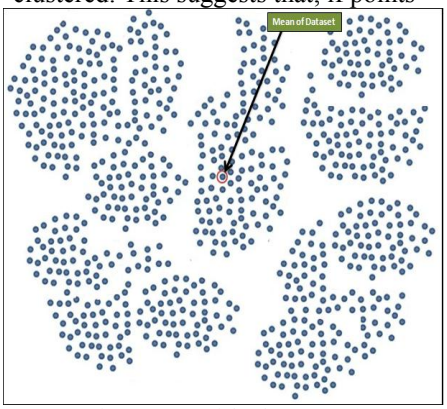

Fig. 2. Mean of the dataset (M_D).

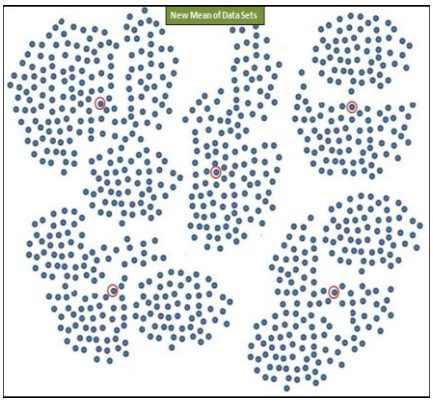

Fig. 3. New points which when averaged yields M_D.

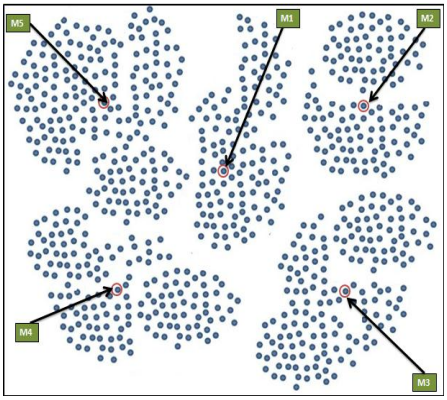

Fig. 4. New Means.

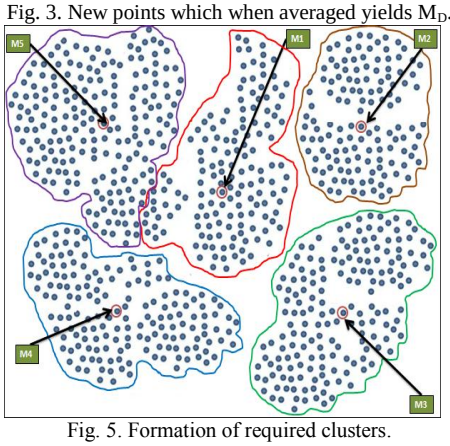

Fig. 5. Formation of required clusters.

representing these unchanging means in a dataset are identified, then the task of clustering reduces to only assigning the remaining points in the dataset into clusters, which are closest to these fixed means based on the Euclidean distance measures. This revelation from the result of the K-Means which indicates that the unchanging K-Means actually yields the mean of the entire dataset when averaged is used in this paper. The algorithm then progresses by first finding the average of the entire dataset (Fig. 1) and then depending on the number of required clusters the K unchanging means are computed using the formulations as shown below

Mean of dataset, M_D = Average of all points in the dataset (Fig. 2)

Mean of dataset, M_D = Average of n points in the dataset for final K means

Mean of dataset, M_D = (Sum of k points representing final K means) / n

For number of required clusters = K, we have

$M_D = (x_1 + x_2 + ...+ x_n) / K$

$M_D * K = (x_1 + x_2 + ...+ x_n)$

Mean of dataset * K = Sum of n points in the dataset

Thus it is seen that we need to find K points (Fig. 3) from the dataset which when added yields the sum equal to (M*k). This is known as Regress Averaging whereby we use the average of a dataset to find the points which yield the average. Each of the new means (Fig. 4) represents the final unchanging means for a clustering problem requiring five clusters. These new means when averaged will give the mean of the entire dataset. This method provides a faster and efficient way of finding the final unchanging means as it eliminates the need for having to stop the clustering process if the means have not repeated or if the clusters have not repeated even after the specified number of iterations have reached. Once the K points yielding the sum equal to (M_D *k) are found, then the remaining points in the dataset are assigned into clusters (Fig. 5), based on closeness to these final K unchanging means.

2.1 The Regress Averaging Clustering Algorithm

Input:

i) A database containing n objects. D = {D_1, D_2, D_3, D_4,……,D_n}.

ii) The number of required clusters K = T.

Output:- A set of K clusters.

Step 1: Find the average of all points in the dataset.

$M_D = (D_1 + D_2 + ...+ D_n) / n$

Step 2: Multiply the average by the number of required clusters K.

$M_D * K = (D_1 + D_2 + ...+ D_n)$

Step 3: Find K points from the dataset which when added yields the sum equal to

(M_D *k). We solve this by the sum of subsets [20].

Step 4: Assign the remaining points in the dataset into clusters formed by these K

points using the Euclidean distance measure shown below.

$$d(i,j)=\sqrt{(w_1-w_2)^2+(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$

Step 5: Output K clusters.

It is important to mention at this stage that the search for K unchanging means by solving the subset sum problem may not seem to be a sensible approach to take, as it is a well-known fact that the mean of a cluster may not always coincide with a point in that cluster. For example, the mean of {1, 2, 6} is 3, which is not in the set. However, it needs to be pointed out that the objective of using subset sum for finding the points which yield the average from the average of the dataset itself is not to find the exact means but an approximation of the same. For example, the mean of {2, 3, 4, 10, 11, 12} is 7 and if it required to partition this set into 2 clusters, then the two points(means) which yield 7*2=14 when added, can be {2, 12}, {3, 11} or {4, 10}. Although the exact means for the two eventual clusters {2, 3, 4} and {10, 11, 12} are 3 and 11 respectively, the same clusters are also obtained when the subset sum yields the two means as {2, 12} or {4, 10}.

In high dimensional or multi-dimensional datasets, the selection of the means for clustering by Regress Averaging can be done by first reducing the dimensions of the data using principal component analysis and then by projecting the principal components onto one of the principal dimensions. This projected dataset can then be apportioned into clusters of data as per the requirement [21].

2.2 Implementation of the proposed algorithm

For the given dataset given below and the clustering requirement of 3 clusters

D = {12, 99, 23, 15, 52, 60, 43, 87, 92}

Mean of the dataset = Sum of n points / n

Mean of the dataset = (12+99+23+15+52+60+43+87+92) / 9 = 483/9

Mean of the dataset = 53 = M_D

For a clustering requirement of K = 3 clusters, we have

$M_D = (12+99+23+15+52+60+43+87+92) / K$

$M_D = 483/3 = 159$

$M_D *K= 159$

$53 * 3 = (D_x + D_y + D_z)$

This suggests that we need to find K=3 points D_x , D_y and D_z from the dataset which when added yields the sum equal to M*K=159. On solving this by sum of subsets we could get the 3 points D_x , D_y and D_z to be 12, 60 and 87. We see that 12+60+87 yields a sum equal to 159. Each of these three points 12, 60 and 87 represent new means which is the final unchanging means for a clustering problem requiring three clusters. These new means when averaged will give the mean of the entire dataset which equals 53. Now the task of clustering eases to just allocating the left over points in the dataset into clusters, which are closest to these final means 12, 60 and 87 based on the Euclidean distance measure.

Thus the clusters so formed are as shown below

Cluster 1 = { 12, 15, 23} , Cluster 2 = {43, 52, 60} and Cluster 3 = {87, 92, 99}

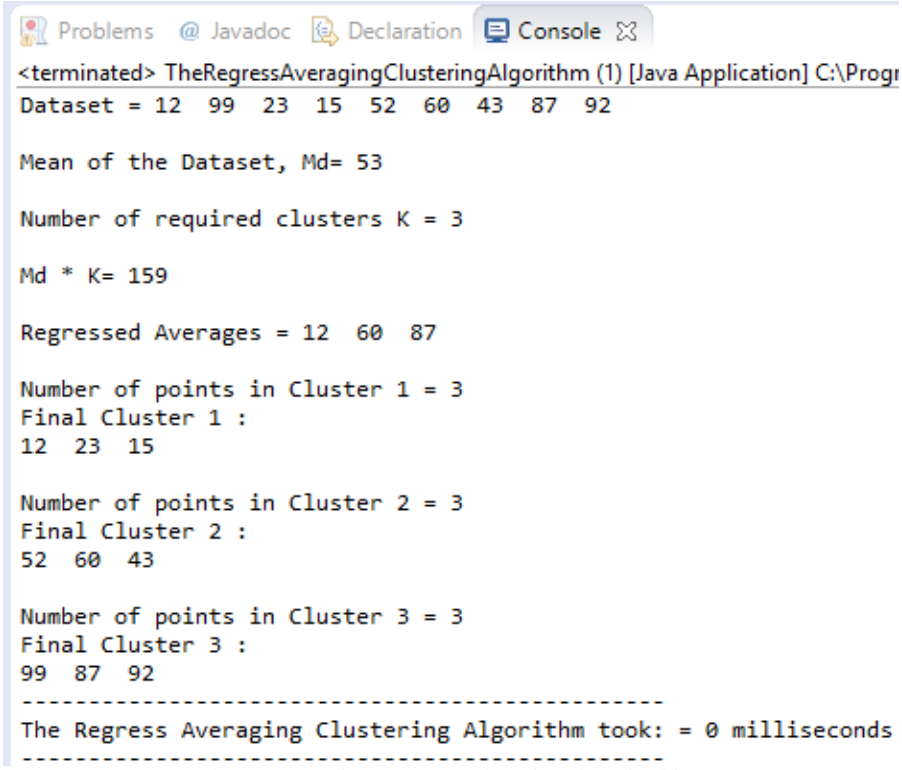

Fig. 6. Output of Regress Averaging Clustering Algorithm

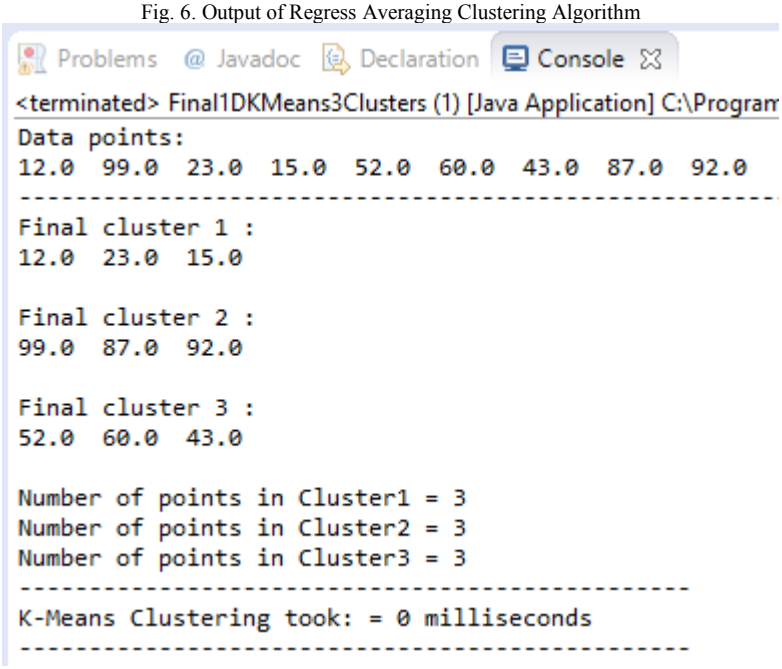

Fig. 7. Output of K Means Clustering Algorithm.

The algorithm is implemented in software for the dataset in D as shown in Fig.6 and also cross verified with the results of the K Means algorithm shown in Fig.7.

3 Conclusion

The K-Means clustering algorithm yields accurate results when either the cluster means repeat in the succeeding iteration or the clusters repeat in the succeeding iteration. The K-Means algorithm in several instances takes a very long time to compute the means and so, abruptly halting the clustering process after a certain number of specified iterations will not

yield accurate clusters. This paper tries to look into this problem by using regress averaging and thereby eliminating the abrupt terminations associated with the number of iterations t.

REFERENCES

[1] A. Baraldi and E. Alpaydin , Constructive feedforward ART clustering networks – Part I and II, IEEE Trans. Neural Netw, vol. 13, no. 3, pp. 645–677, May 2002.

[2] Sunita Jahirabadkar and Parag Kulkarni.: SCAF-An efficient approach to classify subspace clustering. International Journal of Data Mining and Knowledge Management Process, vol. 3 no. 2 (March 2013).

[3] M. Ester, H.P Kriegel, J.Sander, M.Wimmer and X. Xu Incremental Clustering for Mining in a Data Warehousing Environment, Proc. Very Large Databases Conf., 1998.

[4] Charu C. Agarwal and Philip S. Yu, Redefining clustering for high dimensional applications, IEEE transactions on Knowledge and Data Engineering, Vol. 14, No. 2, March/April 2002.

[5] A. Jain and R. Dubes, Algorithms for Clustering Data, New Jersey, Prentice Hall, 1998.

[6] L. Kaufman and P. Rousseuw, Finding Groups in Data-An Introduction to Cluster Analysis. Wiley,1990.

[7] 7. R. Ng and J.Han, Efficient and Effective Clustering methods for Spatial Data Mining, Proc. Very Large Databases Conf., 1994.

[8] M. Ester, H.P Kriegel, J.Sander, M.Wimmer and X. Xu A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proc. Knowledge Discovery in Databases and Data Mining Conf., 1996.

[9] T. Zhang, R.Ramakrishnan, and M.Livny, BIRCH: An Efficient Data Clustering Methods for Very Large Databases, Proc. ACM SIGMOD Conf., 1996.

[10] S. Guha, R. Rastogi, and K.Shim, CURE: An Efficient Clustering Algorithm for Large Databases, Proc. ACM SIGMOD Conf., 1998.

[11] J. Kieinberg, Two Algorithms for Nearest Neighbour Search in High Dimensional space, Proc. ACM Symp. Theory of Computing, 1997.

[12] P. Indyk, and R.Motwani, Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality, Proc. ACM Symp. Theory of Computing, pp. 604-613, 1998.

[13] R. Kohavi and D. Sommerfield, Feature Subset Selection Using the Wrapper Method: Overfitting and and Dynamic Search Space Topology, Proc. Int'l Conf. Knowledge Discovery and Data Mining, 1995.

[14] Jolliffe, I. (2002). Principal component analysis. Springer. 2nd edition.

[15] Dasgupta, S. (2000). Experiments with random projection. Proc.16th Conf. Uncertainty in Artificial Intelligence (UAI 2000).

[16] Rene Vidal, Subspace Clustering: Dimensionality Reduction Methods-Applications in Motion Segmentation and face clustering, IEEE Signal processing Magazine, pp 52-68, March 2011.

[17] Chris Ding, Tao Li, Adaptive Dimension Reduction Using Discriminant Analysis and K-means Clustering. ICML '07, Proceedings of the 24th international conference on Machine learning, Pages 521-528, ACM New York, NY, USA.

[18] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern classification, 2nd ed. Wiley.

[19] Tan, Steinbach, Kumar, An Introduction to Data Mining, Addision-Wesley, 2005.

[20] Adarsh Kumar Verma, 'Ads' Algorithm for Subset Sum Problem, International Journal of Computer Applications (0975-8887), Volume 66 – No. 13, pp 32-34, March 2013.

[21] Zhao Yanchang, Song Junde, A general framework for clustering high dimensional datasets, CCECE 2003 - CCGEI 2003, Montrhal, Mayhi 2003, 0-7803-7781-8/03, pp. 1091-1094, 2003 IEEE