IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 5 (Nov. - Dec. 2012), PP 08-11
www.iosrjournals.org

Clustering of collinear data points in lower dimensions

Terence Johnson[1], Jervin Zen Lobo[2]
[1](*Computer Engineering Department, Agnel Institute of Technology and Design, Goa University, India)*
[2](*Basic Sciences & Humanities Department, Agnel Institute of Technology and Design, Goa University, India)*

Abstract : *Clustering using the basic version of the K-Means algorithm begins by randomly selecting K cluster centers, assigning each point to the cluster whose mean is closest in a Euclidean distance sense, computing the mean vectors of the points assigned to each cluster and using these as new centers in an iterative approach. This suggests that if we identify points in the dataset which represent the final unchanging means, the task of clustering reduces to just assigning the remaining points in the dataset into clusters which are closest to these final means based on the Euclidean Distance measure. Taking a cue from the result of the K-Means algorithm this paper presents an approach for performing collinear clustering based on the idea that values in a dataset can be put into different clusters, depending on which points in the dataset lie at maximum distance from each other. The clusters are formed by finding the minimum Euclidean distance of all points in the dataset and these maximally separated data points.*

Keywords - *Collinear clustering, Maximal distance clustering, Minimum Euclidean distance, J_{min}, J_{max}.*

I. Introduction

Clustering is the process of partitioning or grouping a given set of data points into disjoint clusters [1]. Clustering organizes objects into groups whose members are alike in some way [2]. A cluster can be thought therefore as a collection of objects which are similar between them and are dissimilar to objects belonging to other clusters [3]. Clustering is the process of partitioning or grouping a given set of data points into disjoint clusters [4]. Clustering organizes objects into groups whose members are alike in some way. So basically the goal of clustering is to determine the intrinsic grouping in a set of unlabeled data [5]. It is obvious from the above arguments that clustering hinges on the notion of distance. In order to decide whether a set of points can be split into sub-clusters, with members of a cluster being closer to other members of their cluster than to members of other clusters, we need to say what we mean by "closer to" [6]. Closeness or similarity and distance or dissimilarity can be described by the Euclidean distance measure [7].

Clustering using the classical K-Means method results in obtaining final fixed points which we call the final unchanging means around which all other points in the dataset get clustered [8]. This suggests that if we identify points in the dataset which represent the final unchanging means, the task of clustering reduces to just assigning the remaining points in the dataset into clusters which are closest to these final means based on the Euclidean Distance measure [9]. This paper presents a method for performing clustering based on the idea that values in a dataset can be put into different clusters, by locating in the dataset, J points equaling the number of required clusters K which lie at maximum distance from each other. The remaining points in the dataset are assigned into K clusters formed by finding the minimum Euclidean distance of all points in the dataset and these maximally separated J data points.

II. Main Idea Of The Proposed Clustering Algorithm

The proposed algorithm is one of the simplest unsupervised learning algorithms which may be used to solve the clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters K which are fixed apriori.

Let D = [J_1, J_2, …, Jn] represent the 'n' data values in the data set. The main idea is to find initially two points in the dataset which represent the minimum and maximum values. These two points would be the ones that are the closest point to the origin representing the minimum of the dataset and the furthest point from the origin representing the maximum of the dataset.

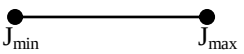


fig. 1. for one dimensional data points

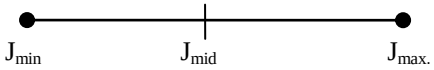
If the problem is to cluster the dataset into two clusters, then it can be done by finding all the points in the dataset which are closest to points J_{min} and J_{max} based on minimum Euclidean distance measure between all the points in the dataset and J_{min} and J_{max} respectively, giving rise to two clusters, as seen in the example below.

Cluster 1 = [J_{min}, J_2, J_3, …J_s]
Cluster 2 = [J_4, J_5, J_6,… J_{s-1}, J_{s+1}, ...J_{max}]

If the problem defined is to cluster the dataset into more than two clusters, say the required number of clusters is 3, then after performing the initial step of finding J_{min} and J_{max}, we search for the point in the dataset that would be equally spaced from J_{min} and J_{max}. If we imagine a straight line between points J_{min} and J_{max}, then the points that is equally spaced from J_{min} and J_{max} is the midpoint of J_{min} and J_{max}. Thus, we can find J_{mid} of J_{min} and J_{max} by the midpoint formula of the line. This point between J_{min} and J_{max} can be calculated as:

$$J_{mid} = \frac{d(J_{min}, J_{max})}{2}$$

where $d(J_{min}, J_{max})$ represents the distance in Euclidean measure between J_{min} and J_{max}



The Euclidean distance between 2 points is defined as the square root of the sum of the squared differences.

$$d(J_{min}, J_{max}) = \sqrt{\sum(J_{min} - J_{max})^2}$$

Since the clustering of collinear data points is being considered, the following formula can also be used to calculate the midpoint of the line joining J_{min} and J_{max} .

$$J_{mid} = \left[\left(\frac{J_{max} - J_{min}}{2}\right)\right]$$

Thus, in general, once J_{min} and J_{max} are found, for any number of clusters K = T, we can divide the interval between J_{min} and J_{max} into equal K-1 intervals.

For K = T, there will be T-2 = X_m points which divide the interval between J_{min} and J_{max} into equal lengths and spaces. X_m represents the m^{th} point which divides J_{min} and J_{max} into equal lengths or spaces and X_m ranges from 1, 2… K-2. Then the points that divide the interval between J_{min} and J_{max} into equal spaces or lengths can be found by using the formula

$$JX_m = J_{min} + X_m \left[\frac{d(J_{min}, J_{max})}{K-1}\right]$$

The points that divide the interval between J_{min} and J_{max} into equal spaces or lengths can also be calculated as given by the following formula:

$$JX_m = J_{min} + X_m \left[\frac{(J_{max} - J_{min})}{K-1}\right]$$

which can be further simplified as follows:

$$JX_m = \left(1 - \frac{X_m}{(K-1)}\right) J_{min} + \left(\frac{X_m}{(K-1)}\right) J_{max}$$

If $JX_m \notin D$, i.e., there is no value JX_m in the database D, then find a value say J_c in the dataset D which is closest to JX_m using minimum Euclidean distance measure and assign J_c to JX_m.

Assign remaining points in the data set according to the minimum Euclidean distance respectively to J_{min}, JX_m and J_{max}.

Consider a collinear clustering problem where K = T = 4. On implementing the algorithm, we will get the dataset of equally spaced points as [J_{min}, JX_1, JX_2, J_{max}]

Once these points [J_{min}, JX_1, JX_2, J_{max}] are calculated, the remaining points in the dataset are put into clusters formed by the 4 points by finding the minimum Euclidean distance between the remaining points and the 4 equally spaced points [J_{min}, JX_1, JX_2, J_{max}]. We can then have 4 clusters for example as shown below:

Cluster 1 = [J_{min}, J_2, J_3,…J_s]
Cluster 2 = [JX_1, J_4, J_5, J_6,… J_{s-1}, J_{s+1}]
Cluster 3 = [JX_2, J_7, J_8, J_9,… J_{s-2}, J_{s+2}… J_{s+10}] and
Cluster 4 = [J_{11}, J_{12}, J_{13},… J_{s-2}, J_{s+4},... J_{s+11}… J_{max}]

III. The Collinear Clustering Algorithm

Algorithm:- The collinear clustering algorithm for partitioning based on the maximum distance between objects in the cluster.
Input:- The number of clusters K and a database containing n objects.
Output:- A set of K clusters

1. Find J_{min} and $J_{max.}$ from D

2. For K = T

   $JX_m = J_{min} + X_m \left[\frac{d(J_{min}, J_{max})}{K-1} \right]$

   where $X_m = X_1, X_2, X_3 \ldots\ldots X_{k-2}$

3. If $JX_m \notin D$, then find a value say J_c in the dataset D which is closest to JX_m using minimum Euclidean distance measure and assign J_c to JX_m

4. Assign remaining points in the dataset according to the minimum Euclidean distance respectively to J_{min}, JX_m and $J_{max.}$

5. Output K clusters.

IV. Experimental Results

For a given one dimensional dataset
D = {2, 4, 10, 12, 3, 20, 30, 11, 25} and given clustering requirement as K = 2 clusters, after implementation, the two clusters were successfully found out to be:
Cluster 1 = {2, 3, 4, 10, 11, 12} and
Cluster 2 = {20, 25, 30}.

The same dataset for 3 clusters yields the results as:

Cluster 1 = {2, 3, 4}
Cluster 2 = {10, 11, 12, 20} and
Cluster 3 = {25, 30}

The same problem for 4 clusters after implementation yields the result as:

Cluster 1 = {2, 3, 4}
Cluster 2 = {10, 11, 12}
Cluster 3 = {20, 25} and
Cluster 4 = {30}

The same problem for 5 clusters after implementation yields the result as:

Cluster 1 = {2, 3, 4}
Cluster 2 = {10, 11}
Cluster 3 = {12}
Cluster 4 = {20, 25} and
Cluster 5 = {30}

The illustration of the working of this algorithm is presented below for the same dataset D = {2, 4, 10, 12, 3, 20, 30, 11, 25} and number of clusters K = 4

4.1 Working Steps

Step 1: $J_{min} = 2$ and $J_{max} = 30$
Step 2: For K = T = 4 = number of clusters required

Since required clusters K = 4, we will have:
T-2 = X_m = 4-2 = 2 equally spaced points and
K-1 = 4-1 = 3 intervals

X_m ranges from 1, 2… K-2.
i.e., having obtained the initial 2 points J_{min} and $J_{max,}$ we have K-2 equally spaced points for K clusters, which in this case are 2 equally spaced points X_1 and X_2 for 4 clusters.

X_1 = 1 = first equally spaced point in the interval between J_{min} and J_{max}

X_2 = 2 = second equally spaced point in the interval

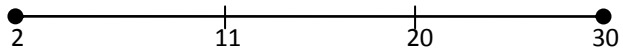
$$JX_m = J_{min} + X_m \left[\frac{d(J_{min}, J_{max})}{K-1} \right]$$

$$JX_1 = 2 + X_1 \left[\frac{28}{3} \right] = 2 + 1 \left[\frac{28}{3} \right] = 11.333$$

$$JX_2 = 2 + X_2 \left[\frac{28}{3} \right] = 2 + 2 \left[\frac{28}{3} \right] = 20.667$$

Step 3: We see that JX_1 = 11.333 and JX_2 = 20.667 do not correspond to any of the values in the dataset D.
So we assign a value J_{c1} in the data set which is closest to
JX_1 = 11.333 and a value J_{c2} in the data set which is closest to JX_2 = 20.667.
By using minimum Euclidean distance measure, we find that J_{c1} = 11 and J_{c2} = 20. We now assign these values J_{c1} and J_{c2} to JX_1 and JX_2 respectively.

So we have the interval between J_{min} and J_{max} divided as shown below:



Step 4: Assign the remaining points in D which are 4, 12, 3, 10, 25 to the clusters formed by 2, 11, 20, 30 to which the points 4, 12, 3, 10 and 25 are most similar based on the minimum Euclidean distance measure.

Step 5: On finding the minimum Euclidean distance for the above step, we get the 4 required clusters as

Cluster 1 = {2, 3, 4}
Cluster 2 = {10, 11, 12}
Cluster 3 = {20, 25} and
Cluster 4 = {30}

V. Conclusion

This paper presents an algorithm for performing collinear clustering. The experimental results demonstrated that this scheme can be used to perform collinear clustering in a fairly simplistic and efficient manner. As it is known that there are various approaches to clustering in Data mining, the purpose of this paper is to add to the existing algorithms in clustering. Improvements may be possible to the basic strategy presented in this paper. Features that could make the strategy for finding the points separated by maximal distances apart from each other more robust can be researched and implemented.

References

[1] Pang-Ning Tan, Michael Steinbachand Vipin Kumar, Introduction to data mining (Addison Wesley, 2006
[2] David Hand, Heikki Mannila and Padhraic Smyth, Principles of data mining (Cambridge, MA: MIT Press, 2001)
[3] Jiawei Han and Micheline Kamber, Data mining-concepts and techniques (San Francisco CA, USA, Morgan Kaufmann Publishers, 2001)
[4] A.K. Jain, R.C. Dubes, Algorithms for clustering data, (Englewood Cliffs, NJ: Prentice-Hall, 1998)
[5] M.R. Anderberg, Cluster analysis for application, (Academic Press, New York, 1973)
[6] J.A. Hartigan, Clustering Algorithms, (Wiley, New York, 1975)
[7] Hand, D.J., Blunt, G., Kelly, M.G. & Adams, N.M. (2000), Data mining for fun and profit, (Statistical Science) 15, 111-131.
[8] Fayyad, U., Data Mining and Knowledge Discovery, Editorial, Proc. IEEE , 1:5-10, 1997. W.J. Book, Modelling design and control of flexible manipulator arms: A tutorial review, Proc. 29th IEEE Conf. on Decision and Control, San Francisco, CA, 1990, 500-506
[9] Aggarwal, Charu C., Han,Jiawei,Wang, Jianyong, & Yu, Philip S. A framework for clustering evolving data streams, VLDB Endowment, Proceedings of the 29th international conference on very large data bases, Vldb '2003, 81–92.