Error Evaluation on K- Means and Hierarchical Clustering with Effect of Distance Functions for Iris Dataset

Harish kumar Sagar
School of IT, RGPV
Bhopal India

Varsha Sharma
School of IT, RGPV
Bhopal India

ABSTRACT

In Data clustering (a sub field of Data mining), k-means and hierarchical based clustering algorithms are popular due to its excellent performance in clustering of large data sets. This paper presents two different comparative studies which includes various Data Clustering algorithms for analyzing best one with minimum clustering error. The foremost objective of this paper is to divide the data objects into k number of different clusters with homogeneity and the each cluster should be heterogeneous to each other. However, these both algorithms (K-Mean and Hierarchical) are not free with the errors. In this paper, firstly various distance has been considered for these two algorithms for comparing and analyzing the best distance methods to solve the existing problems..

Keywords

K-Means, Hierarchical, Euclidean Distance, Manhattan Distance, Filtering cluster, Density Based clustering algorithms on clustering error.

1. INTRODUCTION

Data clustering is a data exploration method that allows objects with same characteristics to be grouped together in order to facilitate their further processing. Data clustering has various engineering application such as the recognition of part families for cellular producer. The k-means clustering algorithm is one of the most accepted data clustering algorithms. It requires the number of cluster in the data to be pre-specified. Searching suitable number of clusters for a given data set is normally a trial-and-error process made more difficult by the subjective nature of deciding what constitutes correct clustering [1].

2. CLUSTERING ALGORITHM

2.1 K-Means algorithm

K-means is probably the most widely used clustering technique [5]. It belongs to the class of iterative centroid-based divisive clustering algorithm. It is different from hierarchical clustering in that it requires the number of clusters, k, be determined in advance.

Algorithm Description

K-Means is an algorithm for partition (or cluster) N data points into K disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion:

$$j = \sum_{j=1}^{n} \sum_{n \varepsilon s_j} \left| x_n - \mu_j \right|^2$$

Where X_n is a vector representing the nth data point and μ_j is the geometric centroid of the data points in S_j.

The procedure of K-Means is:

- Arbitrarily make any partition and clustering the data points into K clusters.
- Compute the centroid of each cluster based on all the data points within that cluster.
- If a data point is not in the cluster with the closest centroid, switch that data point to that cluster.
- Repeat step 2 and 3 until convergence is achieved. By then each cluster is stable and no switch of data point arises

2.1.1 Distance Functions In K-Means ALGORITHM

2.1.1.1 Euclidean distance function

In mathematics, the Euclidean distance or Euclidean metric is the "general" distance between two points that one would measure with a dimension, and is given by the Pythagoras formula. By using this formula as distance, Euclidean space becomes a metric space [2].

Euclidean distance is

D (i , j) =

$$\sqrt{(a_{i1} + a_{j1})^2 + (a_{i2} + a_{j2})^2 + \cdots \ldots + (a_{in} + a_{jn})^2}$$

Where i = ($a_{i1}, a_{i2}, \ldots \ldots \ldots a_{in}$) and j =($a_{j1}, a_{j2}, \ldots \ldots . . a_{jn}$) two n dimension object.

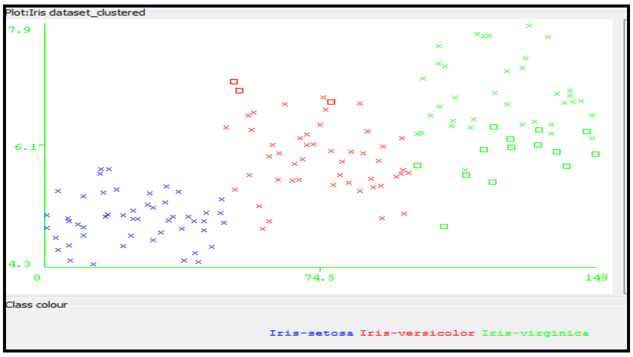

Fig 1: Cluster Diagram of Euclidean Distance Function

In this fig 1 shows cluster generation on by Euclidean function three different type of cluster which is specify by three different colors red, green, and blue. Cross represent selected parameter and squares represent unselected parameter.

2.1.1.2 Manhattan distance function

The function of the Manhattan distance enumerate the distance that can be traveled to get from one data point to another if a network path as follows. The Manhattan distance between two elements is the sum of the differences of the corresponding components[2].

$$D\,(\,i,\,j) = \left|a_{i1} + a_{j1}\right| + \left|a_{i2} + a_{j2}\right| \ldots\ldots \left|a_{in} + a_{jn}\right|.$$

Where i = $(a_{i1}, a_{i2}, \ldots\ldots\ldots a_{in})$ and j = $(a_{j1}, a_{j2}, \ldots\ldots a_{jn})$ two n dimension object.

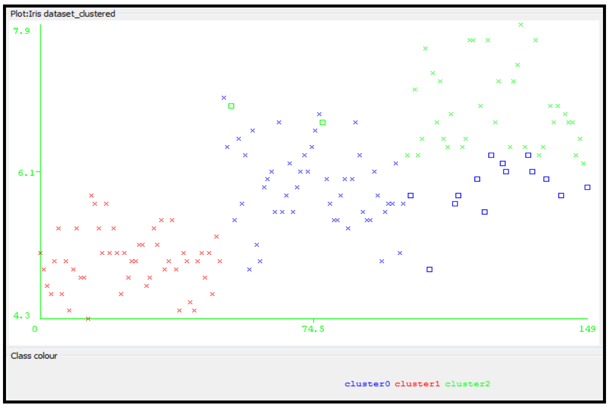

Fig 2: Cluster Diagram by Manhattan Distance Function

In this fig 2 shows cluster generated by specific algorithm three different type of cluster which is specify by three different colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

2.1.1.3 Density based function

Density-based methods suppose that the points that belong to each cluster are drained from a specific probability distribution. The overall distribution of the data is assumed to be a mixture of multiple distributions. The goal of these methods is to identify the groups and their distribution parameters. These methods are designed for discovering clusters of arbitrary shape which are not necessarily convex, namely:

$$x_i, x_j \in c_k$$

This does not necessarily imply that:

$$\alpha * x_i + (1 - \alpha) * x_j \in C_k$$

The idea is to continue growing the given cluster as long as the density (number of objects or data points) in the neighbourhood exceeds some threshold. Namely, the neighbourhood of a given radius has to contain at least a minimum number of objects. When each cluster is characterized by local mode or maxima of the density function, these methods are called mode-seeking. Much work in this field has been based on the underlying assumption that the component densities are multivariate Gaussian (in case of numeric data) or multinomial (in case of nominal data) [3].

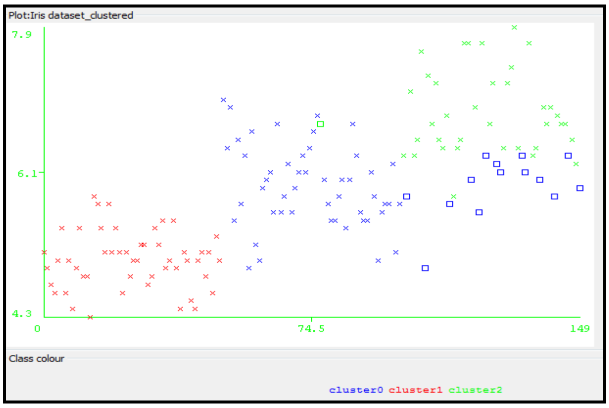

Fig 3: Cluster Diagram of Density Based Algorithm

In this fig 3 shows cluster generated by Density based algorithm three different type of cluster which is specify by three different colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

2.1.1.4 Filter cluster Function

In this segment, the filtering algorithm is illustrated. The algorithm is based on the storing of multidimensional data points a kd tree. A kd-tree is a binary tree, which represents hierarchical subdivision of the point set's bounding box using axis aligned splitting hyper planes. Each node of the kd-tree is linked with a closed box, called a cell. The root's cell is the bounding box of the point set. If the cell contains in general a point (or more generally, less than a small constant), then it was confirmed that a leaf. If not, it is split into two hyper rectangles by an axis-orthogonal hyper plane. We start by computing a kd-tree in favour of the known data points. For each interior node p in the tree, we compute the number of concerned data point’s p: count and weighted centroid p: wgtCent, which is defined to be the vector sum of all the involved points. The real centroid is just p: wgt Cent=p: count. It is easy to convert the kd-tree construction to compute this additional information in the same space and time bounds specified above. The initial centers can be chosen by any technique desired. Remember that, for every one stage of Lloyd's algorithm, for each of the k centers, we need to compute the centroid of the set of data points for which this center is closest. We then move this centre to the computed centroid and proceed to the next stage [4].

```
The Filtering algorithm
Filtering(kdNode p,CandidateSet Q)
{
        C ← p.cell;
        If ( p is a leaf )
        {
                $Q^*$← the closet point in Q to p.point;
                $Q^*$← wgtCent ← $Q^*$.wgtCent +p.point;
        $Q^*C$ .count ← $Q^*$.count+1;
}
Else {
        $Q^*$ ← the closest point in Qto C`s midpoint;
        for each (Q ∈ Q \ { $Q^*$})
if (Q.is Father (Q,$^*$ ,C)) Q ← Q \ {Q};
 if ( |Q|= i)
        {
                $Q^*$ . wgtCent ← $Q^*$.wgtCent +p.wgtCent;
        $Q^*$ C .count ← $Q^*$.count+p.count;
        }
Else {
                        Filter (p.left,Q);
                        Filter(p.right,Q);
                        }
        }
}
```

It remains to describe how to determine whether there is any part of cell C that is closer to candidate Q than to Q^*. Let H be the hyper plane bisecting the line segment QQ. H defines two halfspaces; one that is closer to z and the other to Q^*. If C lies entirely to one side of H, then it must lie on the side that is closer to Q_ (since C's midpoint is closer to Q_) and so Q may be pruned. To determine which is the case, consider the vector ~p . Q ÿ Q_, directed from $Q^*$to Q.

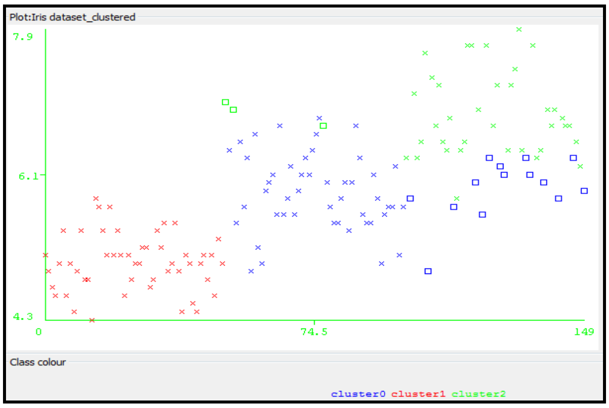

Fig 4: Cluster Diagram Generated by Filtered Algorithm

Figure 4 shows cluster generation on by Filtered algorithm, three different types of clusters which is specify by three different colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

2.2 Hierarchical clustering

These methods build clusters by recursively partitioning the cases, either in a top-down or bottom-up. These methods can be divided as follows. [6].

- Agglomerative hierarchical clustering— Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.
- Divisive hierarchical clustering — All objects initially belong to one cluster. Then the group is divided into subgroups, divided successively into their own subgroups. This process continues until the desired cluster structure is obtained.

2.2.1 Manhattan distance function

Compute the distance that probable travelled to get from one data point to the further if a grid-like path is followedThe Manhattan distance between two elements is the sum of the differences of the corresponding components. The formula for this distance between a point X= (X_1, X_2, etc.) and a point Y= (Y_1, Y_2 etc.) is:

$$d = \sum_{i=1}^{n} |X_i + y_i|$$

Where n is the number of variables, and Xi and Yi are the values of the i^{th} variable, a points X and Y respectively.

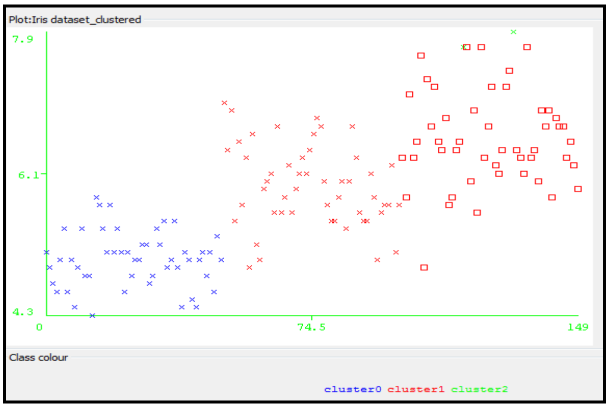

Fig 5: Cluster Diagram Generated by Manhattan Distance Function

Figure 5 shows cluster generated by Manhattan distance function three different type of cluster which is specify by three different types colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

2.2.2 Euclidean distance function:

This is the most commonly selected kind of distance. It simply is the geometric distance in the multidimensional space [9, 10]. The formula for this distance between a point X (X_1, X_2, etc.) and a point Y (Y_1, Y_2, etc.) is:

$$d = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}$$

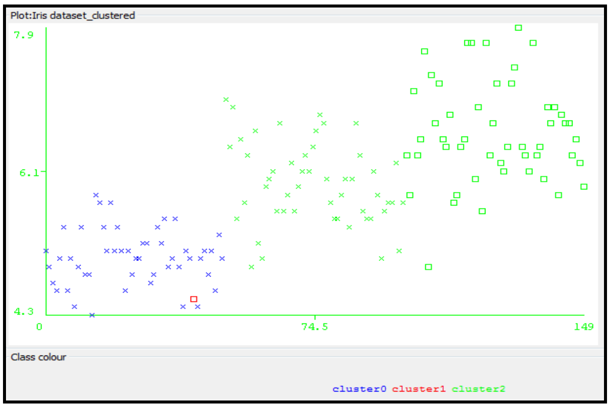

Fig .6: Cluster Diagram Generated by Euclidean Distance

Figure 6: shows cluster generation on by Euclidean distance function three different type of cluster which is specify by three different colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

Derived from the Euclidean distance between two points of data involves calculating the square root of the sum of the squared differences between the values.. The following figure

K-means Clustering algorithms	**Cluster error (in %)**
Euclidean distance clustering algorithm	11.3330
Manhattan distance clustering algorithm	10.6777
Density based clustering algorithm	10.0000
Filtered cluster algorithm	11.3330

illustrates the difference between Manhattan distance and Euclidean distance

2.2.3 Chebyshev Distance Function

In mathematics, Maximum Chebyshev distance metric is a metric defined in a vector space where the distance between two vectors is the largest differences along any coordinate dimension [8]. It is named after Pafnuty Chebyshev.

It is also recognized as chessboard distance, since in the game of chess the minimum number of moves needed by a king to go from one square on a chessboard to another equals the Chebyshev distance between the centers of the squares, if the squares have side length one, as represented in 2-D spatial coordinates with axes aligned to the edges of the board [7].

The Chebyshev distance between two vectors or points p and q, with standard coordinates p_i andq_i, respectively, is

$$D_{Chebyshev}\ (p\ ,q) = max(|p_i - q_i|)$$

$$\lim_{k\to\infty} (\sum_{k\to\infty}^{n} |p_i - q_i|)^{1/k}$$

Mathematically, the Chebyshev distance is a metric induced by the supremum norm or standard uniform. It is an example of an injective metric. In two dimensions, i.e. plane geometry, if the point's p and q have Cartesian coordinates (x_1, y_1) and(x_2, y_2), their Chebyshev distance is:

$$D_{chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$$

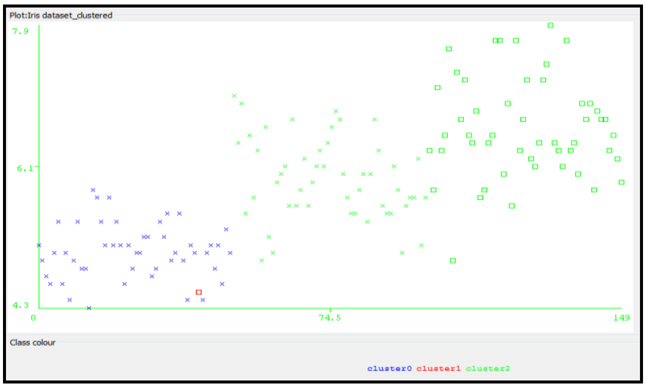

Fig.7: Cluster Diagram Generated by Chebyshev Distance Algorithm

Figure 7: shows cluster generated by Chebyshev Distance algorithm three different type of cluster which is specifying by three different colors red, green, and blue. Cross represent selected parameter and squares unselected parameter.

3. EXPERIMENTS RESULTS

This paper, represent a comparative study of various clustering algorithm to find out a method for robust clustering generation.

Table 1: Result table of k-means clustering algorithm

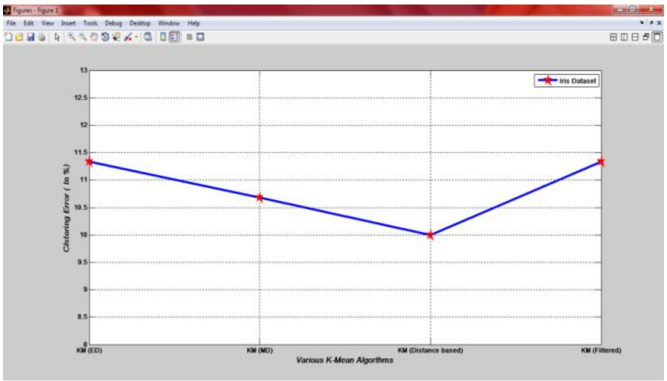

Fig 8: Simulation Chart Result of k means Algorithm

This simulation result chart show as two dimension xy plane in this figure 8: x- axis as a k-means clustering algorithm and y-axis show as a clustering error (in %) , the bold line show as a iris data set in this graph the bold line show error between clustering algorithm and clustering error (in %), we are representing k-mean "KM", Euclidean Distance is represented "ED" , Manhattan Distance "MD" and Distance Based. From figure 8 it is clear that the best k-means clustering algorithms is a Density based clustering algorithm.

Table 2 : Result Table of Hierarchical clustering algorithm

Hierarchical clustering algorithm	**Clustering error (in %)**
Chebyshev distance clustering algorithm	**34**
Euclidean distance clustering algorithm	**34**
Manhattan distance clustering algorithm	**32**

Figure 9: shows that simulation result on by hierarchical clustering algorithms, x-axis as a hierarchical clustering algorithm and y-axis show as a clustering Error (in %) and vertical bold line show as a Iris data set, we are representing Hierarchical Clustering "HC", Chebyshev Distance algorithm "CD", Euclidean Distance "ED" and Manhattan Distance "MD". We have achieved minimum error to generate cluster by Manhattan Distance clustering algorithm, this algorithm best hierarchical clustering algorithm.

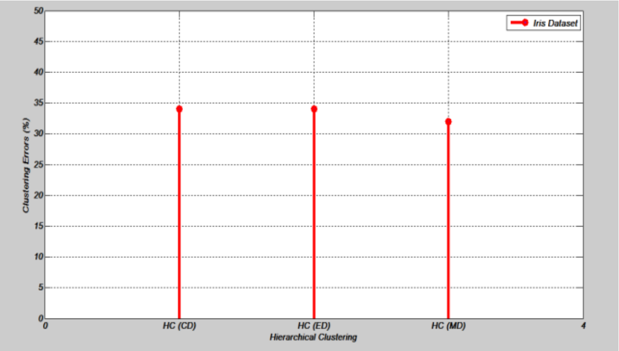

Fig 9: Simulation chart Result of Hierarchical Algorithms

4. CONCLUSION

In this paper, the problem was to predict best clustering algorithms by comparing various clustering techniques. Performances of these clustering methods are measured by the percentage of the incorrectly classified data instances. If the percentage of the incorrectly classified instances will be low then the performance of the clustering is to be considered well. Our paper has presented comparative studies which divided in to two parts where first we study k-means clustering algorithms second in we study hierarchical clustering. In k-means clustering algorithm, minimum clustering error has given by Density based algorithm (10.0000 %). Thereafter, in case of hierarchical clustering algorithms the minimum clustering error has given by Manhattan Distance clustering algorithm is (32%).

5. REFERENCES

[1] Pham, D. T., S. S. Dimov, and C. D. Nguyen. "Selection of K in K-means clustering." Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 219.1 (2005): 103-119

[2] Awasthi, Rekha, Anil K. Tiwari, and Seema Pathak. "Empirical Evaluation on K Means Clustering with Effect of Distance Functions for Bank Dataset." IJITR1.3 (2013): 233-235.

[3] Density-based clustering algorithms DBSCAN and SNN by Adriano Moreira, Maribel Y. Santos and Sofia Carneiro.

[4] Kanungo, Tapas, et al. "An efficient k-means clustering algorithm: Analysis and implementation." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7 (2002): 881-892.

[5] J.L. Bentley, Multidimensional Binary Search Trees Used for Associative Searching.

[6] Esteves, Rui Maximo, Rui Pais, and Chunming Rong. "K-means clustering in the cloud--a mahout test." In Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on, pp. 514-519. IEEE, 2011.

[7] F. Caoa et. al., "An initialization method for the k-Means algorithm using neighborhood model", Computers and Mathematics with Applications, vol. 58, pp. 474 – 483, 2009.

[8] Han, Jiawei, Kamber, Micheline. (2000) Data Mining: Concepts and Techniques. Morgan Kaufmann.

[9] Euclidean Distance in http://people.revoledu.com /kardi/tutorial/Similarity/EuclideanDistance.html.

[10] Euclidean distance in http://en.wikipedia.org/wiki/ Euclidean_distance#One-dimensional_distance