Comparing Unsupervised Learning Methods for High-Dimensional Data

Parag Dhere, Apurva Wankhade, Suram Saraswati Anugna, and Sonia Shah,
Department of Computer and Information Science, University of Massachusetts,Dartmouth
pdhere@umassd.edu,
awankhade@umassd.edu,
sshah4@umassd.edu,
ssuram@umassd.edu

***Abstract*—Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining software is one of many analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are more like other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. Our main aim to show the comparison of different clustering algorithms on different datasets and find out which algorithm will be most suitable for the users. Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain minute details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters corresponds to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is unsupervised learning of a hidden data concept. Data mining deals with large databases that impose on clustering analysis additional severe computational requirements.**

I. INTRODUCTION

Data mining is the use of automated data analysis techniques to uncover previously undetected relationships among data items. Data mining often involves the analysis of data stored in a data warehouse. Three of the major data mining techniques are regression, classification and clustering. In this project, we are working only with the clustering because it is most important process, if we have a very large database.

Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters.

Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bio-informatics. We are using MATLAB and R data mining tools for this purpose. It provides better interface to user than compare the other data mining tools.

II. WHAT IS CLUSTER ANALYSIS?

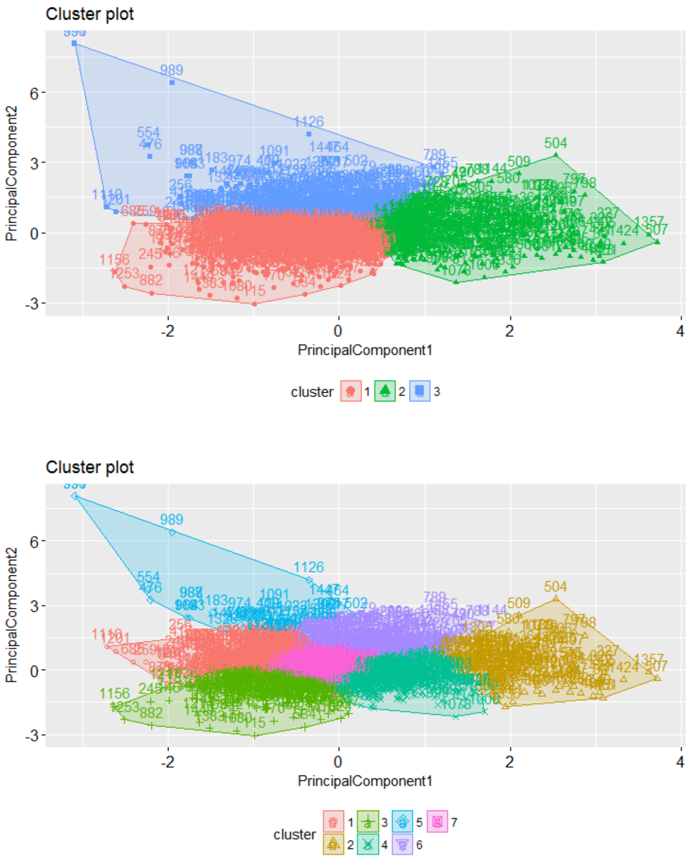
Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups. The greater the likeness (or homogeneity) within a group, and the greater the disparity between groups, the better or more distinct the clustering. The definition of what constitutes a cluster is not well defined, and, in many applications clusters are not well separated from one another.

Nonetheless, most cluster analysis seeks as a result, a crisp classification of the data into non-overlapping groups. Clustering can be divided into two sub groups: Hard Clustering: In hard clustering, each data point either belongs to a cluster completely or not. Soft Clustering: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

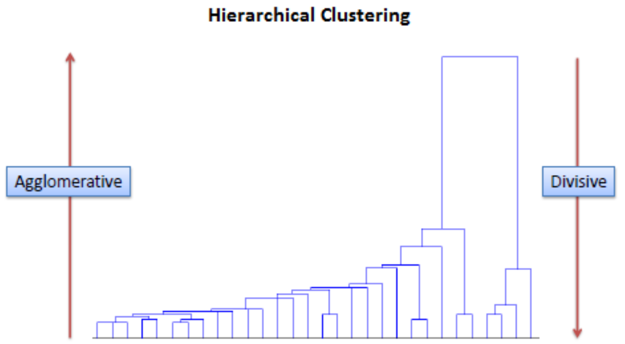
III. TYPES OF CLUSTERING TECHNIQUES

We distinguish two types of clustering techniques: Partitional and Hierarchical. Their definitions are as follows:

Partitional: Given a database of n objects, a partitional clustering algorithm constructs partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster. Partitional Clustering algorithms try to locally improve a certain criterion. First, they compute the values of the similarity or distance, they order the results, and pick the one that optimizes the criterion. Hence, many them could be considered as greedy-like algorithms.



a.Partitional



(b) Hierarchical

Hierarchical: Hierarchical algorithms create a hierarchical decomposition of the objects. They are either agglomerative (bottom-up) or divisive (top-down):

*A. **Agglomerative** algorithms start with each object being a separate cluster itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a solitary group or at any other point the user wants. These methods generally follow a greedy-like bottom-up merging.*

*B. **Divisive** algorithms follow the opposite strategy. They start with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster. This is like the approach followed by divide-and-conquer algorithms.*

Most of the times, both approaches suffer from the fact that once a merge or a split is committed, it cannot be undone or refined.

IV. DATASET

For performing the comparison analysis, we need the project datasets. In this research, we are taking data into consideration from three data repositories. IRIS, GLASS and YEAST. It provides the past project data. This should have been taken the different- different nature. These repositories are very helpful for the researchers. We can directly apply this data in the data mining tools and predict the result.

About Glass data set:

Creator: B. German Central Research Establishment Home Office Forensic Science Service Aldermaston, Reading, Berkshire RG7 4PN

Donor: Vina Spiehler, Ph.D., DABFT Diagnostic Products Corporation (213) 776-0180 (ext 3014)
Attribute Information: 1. Id number: 1 to 214 2. RI: refractive index 3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10) 4. Mg: Magnesium 5. Al: Aluminum 6. Si: Silicon 7. K: Potassium 8. Ca: Calcium 9. Ba: Barium 10. Fe: Iron 11. Type of glass: (class attribute) – 1 building$_{w}indows_{f}loat_{p}rocessed - -2building_{w}indows_{n}on_{f}loat_{p}rocessed - -3vehicle_{w}indows_{f}loat_{p}rocessed - -4vehicle_{w}indows_{n}on_{f}loat_{p}rocessed(noneinthisdatabase) - -5containers - -6tableware - -7headlamps$

About Yeast Data set:
Creator and Maintainer:
Kenta Nakai Institue of Molecular and Cellular Biology Osaka, University 1-3 Yamada-oka, Suita 565 Japan nakai@imcb.osaka-u.ac.jp http://www.imcb.osaka-u.ac.jp/nakai/psort.html
Donor: Paul Horton (paulh@cs.berkeley.edu)
Date: September, 1996
See also: ecoli database
Attribute Information: 1. Sequence Name: Accession number for the SWISS-PROT database 2. mcg: McGeoch's method for signal sequence recognition. 3. gvh: von Heijne's method for signal sequence recognition. 4. alm: Score of the ALOM membrane spanning region prediction program. 5. mit: Score

of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins. 6. erl: Presence of ”HDEL” substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute. 7. pox: Peroxisomal targeting signal in the C-terminus. 8. vac: Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins. 9. nuc: Score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins.
About Iris data set:
Creator: R.A. Fisher
Donor: MichaelMarshall (MARSHALLDated on July, 1988

Attribute Information:

1. Id number: 1 to 214 2. Sepal length in cm 3. Sepal width in cm 4. Petal length in cm 5. Petal width in cm 6. Class: – Iris Setosa – Iris Versicolour – Iris Virginica

V. METHODOLOGY

Our methodology is very simple. We are taking the past project data from the repositories and apply it on the R and MATLAB. In it, we are applying different- different clustering algorithms and predict a useful result that will be very helpful for the unaccustomed users and new researchers.

VI. CLASSIFICATION OF CLUSTERING ALGORITHM

A. K- Means methodology

The family of clustering algorithms includes the first ones that appeared in the Data Mining Community, the k-means algorithm. The k-means algorithm [Hartigan 1975; Hartigan Wong 1979] is by far the most popular clustering tool used in scientific and industrial applications. k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results into a partitioning of the data space into Verona cells. The goal in k-means is to produce k clusters from a set of n objects, so that the squared-error objective function:

$$E = \sum_{i=1}^{k}\sum_{p\epsilon c_i}|p-m_i|^2 \tag{1}$$

is minimized. In the above expression, Ci are the clusters, p is a point in a cluster Ci and mi the mean of cluster Ci. The mean of a cluster is give n by a vector, which contains, for each attribute, the mean values of the data objects in this cluster and, input parameter is the number of clusters, k and as an output the algorithm returns the centers, or means, of every cluster Ci, most of the times excluding the cluster identities of individual points. The distance measure usually employed is the Euclidean distance. Both for the optimization criterion and the proximity index, there are no restrictions, and they can be specified according to the application or the users preference.

Euclidean distance, defined as:

$$d(\hat{x},\hat{y}) = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{2}$$

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, calculate the positions of the K centroids again.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

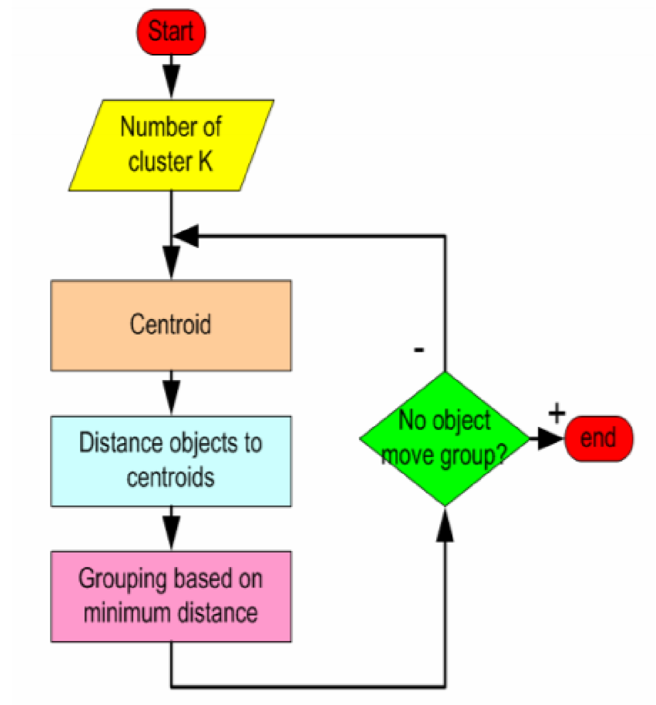

Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

Fig:k-means

Advantages to Using this Technique:

- With many variables, K-Means may be computationally faster than hierarchical clustering (if K is small).
- K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

Disadvantages to Using this Technique

- Difficulty in comparing quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome).
- Fixed number of clusters can make it difficult to predict what K should be.
- Does not work well with non-globular clusters.
- Different initial partitions can result in different final clusters.

- It is helpful to rerun the program using the same as well as different K values, to compare the results achieved.

Iris dataset takes 3 iteratioon

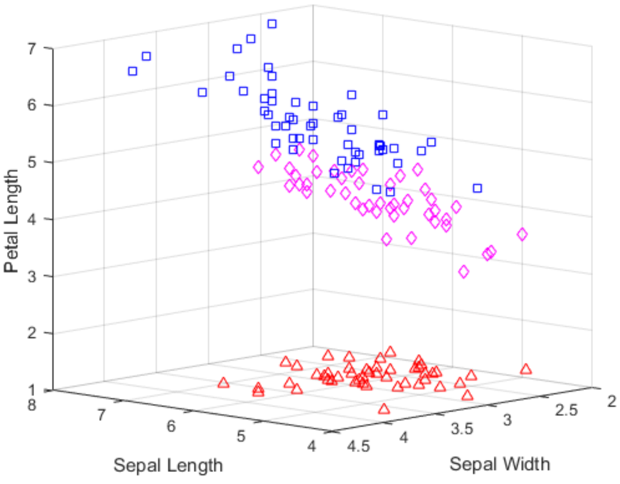


Fig:Iris dataset takes 3 iterations

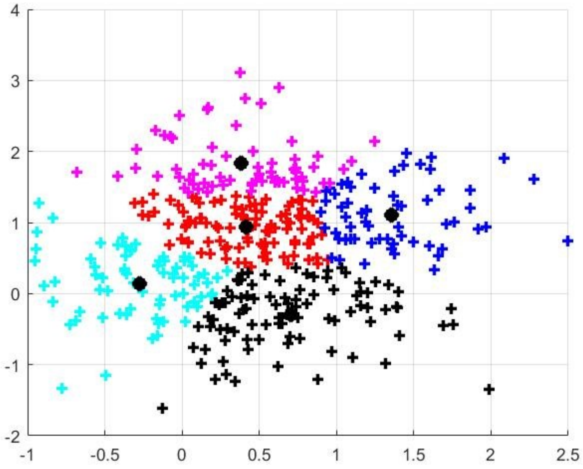


Fig: Glass Dataset takes 5 iterations for 5 cluster

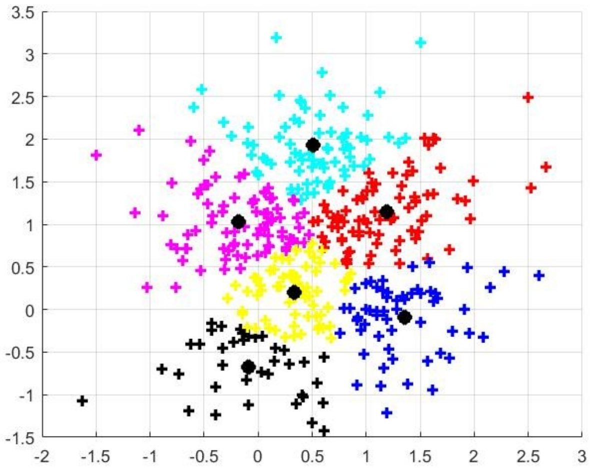


Fig: Yeast dataset takes 8 iterations for 6 cluster

For interpretation and validation of consistency within cluster analysis the Elbow Method is used. It is designed to help finding the appropriate number of clusters in a dataset.

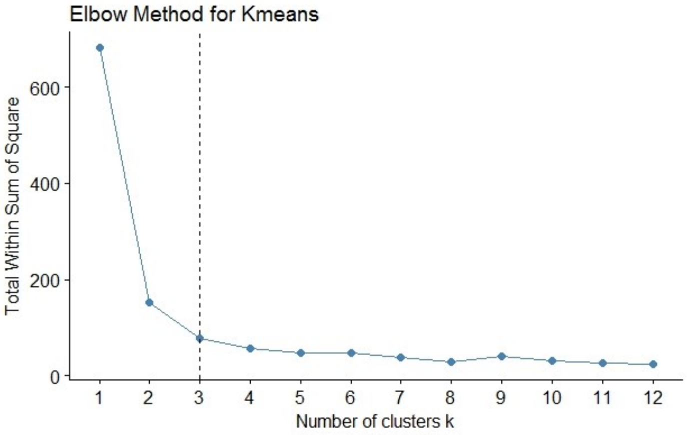


Fig: Elbow method of Iris datasets

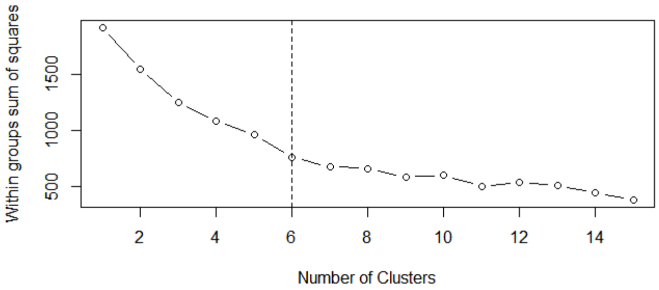


Fig: Elbow method of Glass datasets

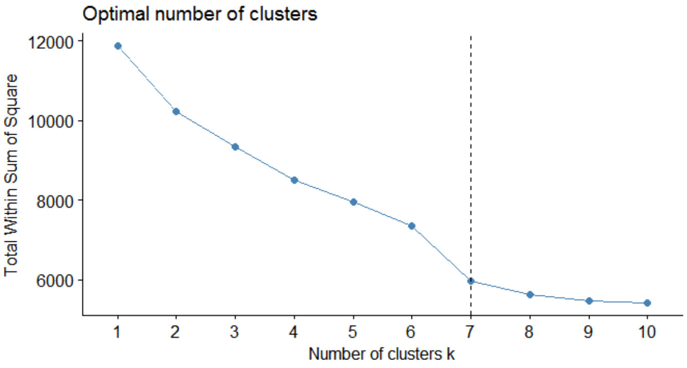


Fig: Elbow method of Yeast datasets

B. *Hierarchical methodology*

Hierarchical clustering builds a cluster hierarchy or, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down).

Agglomerative: This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

Basic algorithm is as:
1. Compute the distance matrix between the input data points.
2. Let each data point be a cluster.
3. Repeat
4. Merge the two closest clusters.
5. Update the distance matrix.
6. Until only a single cluster remains.

The decision of merging two clusters is taken based on closeness of these clusters. There are multiple metrics for deciding the closeness of two clusters, but we consider Euclidean:

$$EuclideanDistance : ||a-b||_2 = \sqrt{(\sum(a_i-b_i))} \quad (3)$$

Key operation is the computation of the distance between two clusters.

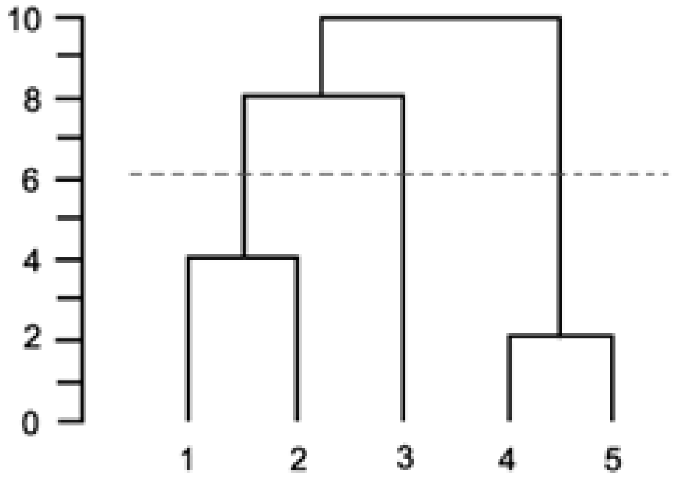


Fig: Agglomerative Example

Dissimilarity between two clusters

Single-link clustering:
Single-link distance between clusters Ci and Cj is the minimum distance between any object in Ci and any object in Cj .
The method is also known as nearest neighbor clustering

$$D_{sl}(C_i, C_j) = min_{x,y}\{d(x,y)|X\epsilon\ C_{i,y}\epsilon\ C_j\}$$

(4)

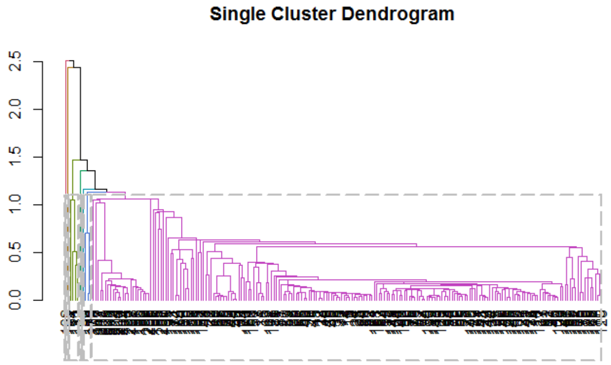


Fig: Iris Dataset

Complete-linkage clustering:
Complete-link distance between clusters Ci and Cj is the maximum distance between any object in Ci and any object in Cj
The method is also known as farthest neighbor clustering

$$D_{cl}(C_i, C_j) = max_{x,y}\{d(x,y)|X\epsilon\ C_{i,y}\epsilon\ C_j\}$$

(5)

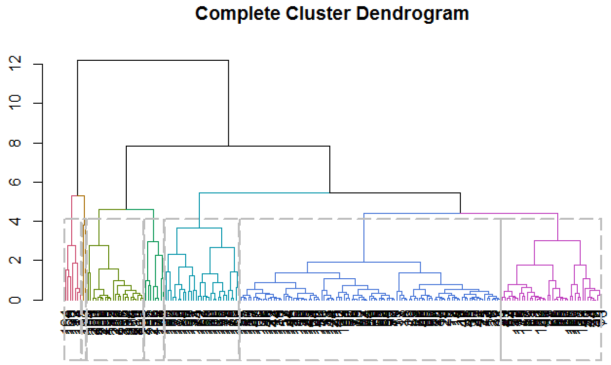


Fig: Iris Dataset

Average-linkage Clustering:
Group average distance between clusters Ci and Cj is the average distance between any object in Ci and any object in Cj

D$_{avg}(C_{i,j}) = \frac{1}{|C_i|\times|C_j|}\sum_{xi,yi} d(x,y)$(6)

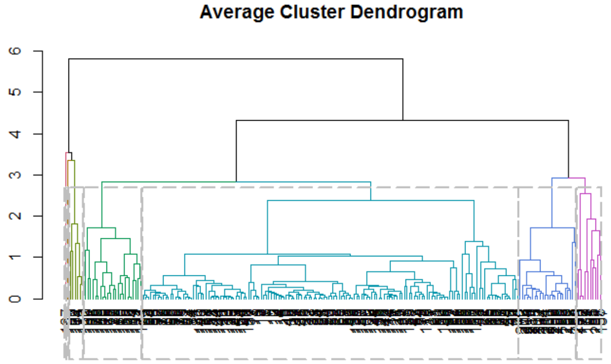


Fig: Iris Dataset

Advantages of hierarchical clustering:
1.Embedded flexibility regarding the level of granularity.
2.Ease of handling of any forms of similarity or distance.
3.Consequently, applicability to any attribute types.

Disadvantages of hierarchical clustering:
1.Vagueness of termination criteria
2.The fact that most hierarchical algorithms do not revisit once constructed (intermediate) clusters with the purpose of their improvement.

Difference between K-Means and Hierarchical clustering:

- Hierarchical clustering cant handle big data well but K-Means clustering can. This is because the time complexity of K-Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. O(n2).
- In K-Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K-Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K-Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram.
- K-Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram.

C. Spectral Methodology

In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm. Nevertheless, on the first glance spectral clustering looks a bit mysterious, and it is not obvious to see why it works at all and what it really does. Though spectral clustering algorithms are simple and efficient, their performance is highly dependent on the construction of a similarity matrix. We assume that we are given data points x1, . . ., xn which can be arbitrary objects, and their comparisons sij = s (xi, xj), measured according to some similarity function which is symmetric and non-negative. We denote the corresponding similarity matrix by S = (sij)i,j=1...n.

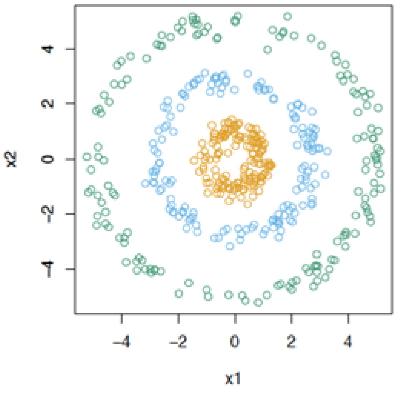


Fig: Spectral Clustering

Given an enumerated set of data points, the similarity matrix may be defined as a symmetric matrix A, where

$$A_{ij} >= 0 \tag{7}$$

represents a measure of the similarity between data points with indices i and j. The general approach to spectral clustering is to use a standard clustering method(there are many such methods, k-means is discussed below) on relevant eigenvectors of a Laplacian matrix of A. There are many ways to define a Laplacian which have different mathematical interpretations, and so the clustering will also have different interpretations. The eigenvectors that are relevant are the ones that correspond to smallest several eigenvalues of the Laplacian except for the smallest eigenvalue which will have a value of 0. For computational efficiency, these eigenvectors are often computed as the eigenvectors corresponding to the largest several eigenvalues of a function of the Laplacian. It partitions points into two sets (B1,B2) based on the eigenvector v corresponding to the second-smallest eigenvalue of the symmetric normalized Laplacian defined as

$$L^{norm} = 1 - D^{1/2}AD^{-1/2}$$

(8)

where D is the diagonal matrix

$$D_{ii} = \sum_j A_{ij} \tag{9}$$

A mathematically equivalent algorithm takes the eigenvector corresponding to the largest eigenvalue of the random walk normalized Laplacian matrix .

P=D^{-1}A(10)

Another possibility is to use the Laplacian matrix defined as,

L= D - A

Rather than the symmetric normalized Laplacian matrix. Partitioning may be done in various ways, such as by computing the median **m** of the components of the second smallest eigenvector **v**, and placing all points whose component v in is greater than **m** in **B1**, and the rest in **B2** . The algorithm can be used for hierarchical clustering by repeatedly partitioning the subsets in this fashion. If the similarity matrix **A** has not already been explicitly constructed, the efficiency of spectral clustering may be improved if the solution to the corresponding eigenvalue problem is performed in a matrix-free fashion (without explicitly manipulating or even computing the similarity matrix), as in the Lanczos algorithm. For large-sized graphs, the second eigenvalue of the (normalized) graph Laplacian matrix is often ill-conditioned, leading to slow convergence of iterative eigenvalue solvers. Preconditioning is a key technology accelerating the convergence, e.g., in the matrix-free LOBPCG method. Spectral clustering has been successfully applied on large graphs by first identifying their community structure, and then clustering communities. Spectral clustering is closely related to nonlinear dimensionality reduction, and dimension reduction techniques such as locally-linear embedding can be used to reduce errors from noise or outliers.

Spectral clustering algorithm:
Input: Similarity matrix

$$S\epsilon\ R_{n\times n} \tag{11}$$

, number k of clusters to construct.

- Construct a similarity graph. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- Compute the first k eigenvectors v1, . . . , vk of L.
- Let

  $$V\epsilon\ R_{n\times k} \tag{12}$$

  be the matrix containing the vectors v1, . . . , vk as columns.

- For i = 1, . . . , n, let yi $\epsilon\ R^k bethevectorcorrespondingtothei - throwofV. Clusterthepoints(\mathbf{y}_i)_{i=1,\ldots,n}inR_kwiththek - meansalgorithmintocl$

Output: Clusters A1, . . . ,Ak with

$$Ai = j|yj\epsilon\ Ci \tag{13}$$

.

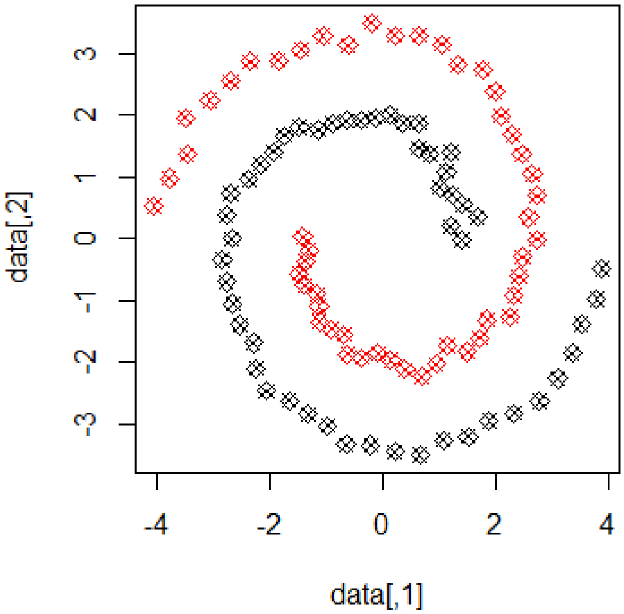


Fig: Iris Dataset

Difference between Spectral and K-means:

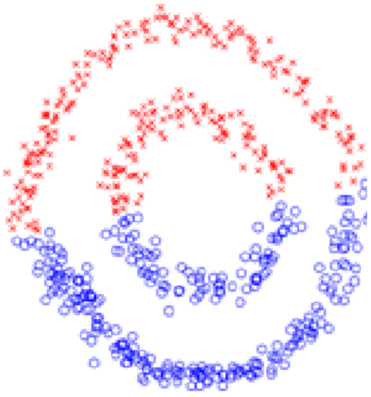


Fig:K-Means with 2 clusters

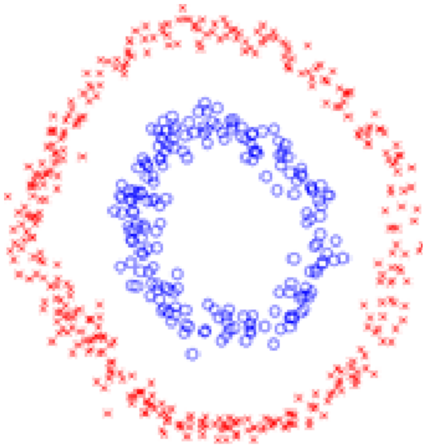


Fig:Spectral with 2 clusters

K-means clustering, divide the objects into k clusters such that some metric relative to the centroids of the clusters are minimized. While, for spectral clustering data points as nodes of connected graph and clusters are found by partitioning this graph, based on its spectral decomposition, into subgraphs.
K-means is a clustering method that aims to find the positions i, i=1...k of the clusters that minimize the distance from the data points to the cluster. The K-means clustering uses the square of the Euclidean distance formula.

$$d(X, \mu_i) = ||X - \mu_i||_2^2 \tag{14}$$

In Spectral Clustering, the starting point is a NxN matrix of pairwise similarities between all observation pairs. We represent the observations in a similarity graph as:

1.The N vertices/nodes vi represent the observations, and pairs of vertices are connected by an edge (link) if their similarity is positive (or exceeds some threshold).
2.The edges are weighted by the sii.
K-means is extremely sensitive to cluster center initialization. Bad initialization can lead to Poor convergence speed and Bad overall clustering.
For Spectral Clustering, there are number of issues like:

- We must choose the type of similarity graphe.g. nearest neighbors, and associated parameters such as the number of nearest of neighbors.
- We must also choose the number of eigenvectors to extract from L and finally, as with all clustering methods, the number of clusters.

The k-means algorithm works reasonably well when the data fits the cluster model:
1. The clusters are spherical: the data points in a cluster are centered around that cluster.
2. The spread/variance/density/size of the clusters is similar: Each data point belongs to the closest cluster.

For Spectral clustering: Effective in tasks like image processing. Scalability challenge: Computing eigenvectors on a large matrix is costly. Can be combined with other clustering methods.

VII. Result

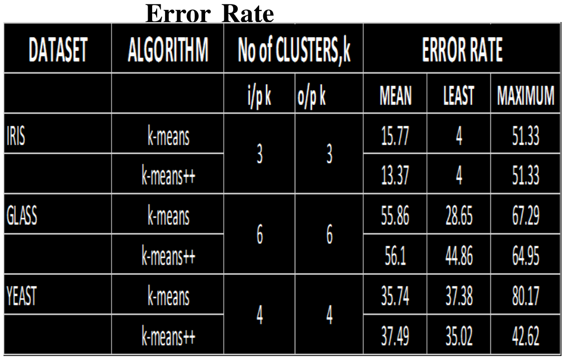
How Algorithms are Compared ?:

The three clustering algorithms are compared according to the following factors:
1. The size of the dataset
2. Number of the clusters
3. Types of dataset
4. Types of software.

Accuracy

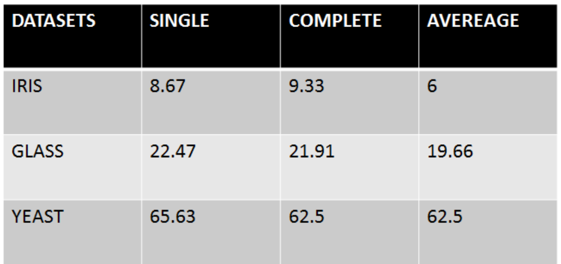
Dataset	K-Means	Hierarchical Clustering	Spectral Clustering
Iris	87.987	84	51.93
Glass	76.8	81.91	59.76
Yeast	67.231	47	43.83

Error Rate



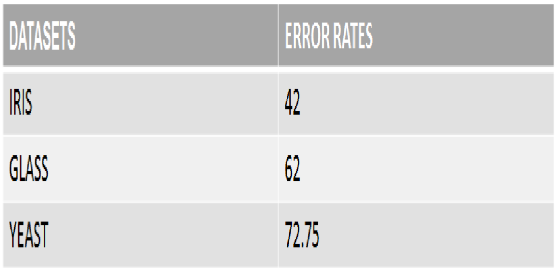
DATASET	ALGORITHM	No of CLUSTERS,k		ERROR RATE		
		i/p k	o/p k	MEAN	LEAST	MAXIMUM
IRIS	k-means	3	3	15.77	4	51.33
	k-means++			13.37	4	51.33
GLASS	k-means	6	6	55.86	28.65	67.29
	k-means++			56.1	44.86	64.95
YEAST	k-means	4	4	35.74	37.38	80.17
	k-means++			37.49	35.02	42.62

Fig: KMean Clustering



DATASETS	SINGLE	COMPLETE	AVEREAGE
IRIS	8.67	9.33	6
GLASS	22.47	21.91	19.66
YEAST	65.63	62.5	62.5

Fig: Hierarchical Clustering



DATASETS	ERROR RATES
IRIS	42
GLASS	62
YEAST	72.75

Fig: Spectral Clustering

VIII. CONCLUSION

After analyzing the results of testing the clustering algorithms and running them under several factors and situation, the following conclusions are obtained.

- As the number of clusters, k becomes greater; the performance of Spectral Algorithm becomes lower.
- The performance of k-means is better than hierarchical clustering algorithm.
- As the value of k becomes greater, the accuracy of hierarchical clustering becomes better until it reaches the accuracy of Spectral algorithm.
- K-means has less quality(accuracy) than the others.
- All the algorithms have some ambiguity in some (noisy) data when clustered.
- The quality of k-means become very good when using huge dataset.
- Hierarchical clustering show satisfactory results when using small dataset.
- Hierarchical and Spectral clustering algorithms give best result when using random dataset and vice versa.
- Running the clustering algorithms using any software gives almost the same results even when changing any of the factors because most software use the same procedures and ideas in any algorithm implemented by them.

IX. FUTURE WORK

This paper was intended to compare between some data clustering algorithms. Through our extensive search, we were unable to find any study that attempts to compare between the three clustering algorithms under investigation.
As a future work, comparisons between these three algorithms can be attempted according to distinct factors other than those considered in this paper. One crucial factor is normalization. Comparing between the results of algorithms using normalized data or non-normalized data will give different results.

X. APPLICATION

From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others. Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. Data mining adds to clustering the complications of very large datasets with very many attributes of several types. This imposes unique computational requirements on relevant clustering algorithms. Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

XI. REFERENCES

[ABKS96] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and J org Sander. OPTICS: Ordering Points To Identify the Clustering Structure. In Proceedings of the International Conference on Management of Data, (SIGMOD), volume 28(2) of SIGMOD Record, pages 4960, Philadelphia, PA, USA, 13 June 1996. ACM Press.

[And73] Michael R. Anderberg. Cluster analysis for applications. Academic Press, 1973.

AGGR98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In Proceedings of the International Conference on Management of Data, (SIGMOD), volume 27(2) of SIGMOD Record, pages 94105, Seattle,WA, USA, 14 June 1998. ACM Press.

ANDERBERG, M. 1973. Cluster Analysis and Applications. Academic Press, New York.

www.wikipedia.com

XII. APPENDIX

We had meetings every week for at least an hour.
Feb7th: Group was formed and we had an informal meeting for introduction.
Feb 12th: Started selecting the Technique by researching through different papers about selecting the technique.
Feb 18th: All finalized with Unsupervised Learning technique and started with the research of methods and datasets to be chosen.
Feb 19th: All came on a decision of choosing 3 methods and 3 datasets.
Started working for Project Proposal and submitted by Feb 22nd Feb 23rd: Started working for Project Proposal Presentation.
Feb 28th: Project Proposal was presented by Anugna and Sonia.
March 5th: Successful plotting of data points and clusters for all datasets in R.
March 12th: Successful implementation of k-means in R and MATLAB for all datasets.
March 19th: Hierarchical clustering study and choose Agglomerative algorithm. Implemented Single-Link dendrogram.
March 20th: Colorful dendrograms.
March 21st: Calculating accuracy and error rates for methods implemented till dates.
March 30th: Started working on Elbow methods for all datasets and Preparation of Mid-term Project presentation.
April 6th: Mid-term Project Presentation presented by all Group Members.
April 9th: Complete-Link and Average- Link Hierarchical Clustering.
April 16th: Spectral Clustering- study and research started.
April 20th: Final Report and Presentation started.
April 27th: Presentation Submitted.

CONTRIBUTION:

k-means Clustering:	Apurva Wankhade
Hierarchical Clustering:	Parag Dhere
Elbow method:	Sonia Shah
Spectral Clustering	Anugna
Project Proposal Presentation:	Parag Dhere
Mid-term Presentation:	Parag Dhere
Project Report:	Apurva Wankhade
Project Presentation:	Parag Dhere
LaTeX Report Preparation:	Parag Dhere
LaTeX Presentation Preparation:	Sonia Shah Parag Dhere