HIERARCHICAL AGGLOMERATIVE CLUSTERING USING PCA

Manohar Akula
Ira A. Fulton Schools of Engineering
Arizona State University
Tempe, USA
makula@asu.edu

Abstract*— Engineering, Biology, Psychology, Economics, and Recommendation systems all use clustering. Because the number of clusters or model parameters is seldom known and must be decided before clustering, the number of clusters or model parameters is rarely known and must be calculated before clustering. A method for clustering with multiple clusters has been provided. Hierarchical clustering is a type of cluster analysis used in data mining that aims to create a hierarchy of clusters. There are two sorts of strategies for hierarchical clustering: Agglomerative: This is a "bottom-up" strategy in which each observation is placed in its own cluster, with pairs of clusters merging as one progresses up the hierarchy. Divisive: This is a "top down" technique in which all observations begin in one cluster and are divided iteratively as one progresses down the hierarchy.

I. INTRODUCTION

"Clustering is defined as "the process of organizing datasets into groups of comparable data points." Clustering is based on the following information: Points belonging to the same group are as similar as feasible, whereas points belonging to separate groups are as distinct as possible. Clustering may be seen in items placed in a mall, books grouped in a library, and so forth.

Types of Clustering:

i. Exclusive Clustering.
ii. Overlapping Clustering.
iii. Hierarchical Clustering.

Hierarchical Agglomerative *Clustering:*

It employs a "bottom-up" technique, in which each observation is assigned to its own cluster, with pairs of clusters merging as one progresses up the hierarchy.

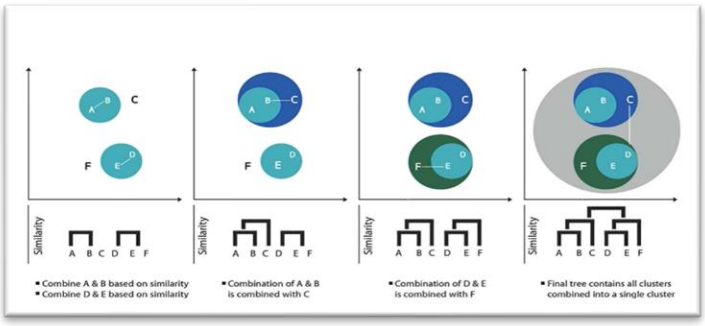

Fig.1

Fig.1: *The figure shows the clustering into a dendrogram using Hierarchical Agglomerative Clustering.[1]*

The capacity to examine dendrograms, which gives insight into the degree of similarity between any two data points, is one of the most appealing aspects of employing HAC.

There are a variety of ways to connect the points. "Ward Linkage" was utilized in this presentation to reduce the variance of the clusters being merged. This is quite close to K-Means' minimization of Within Cluster Sum of Squares (WCSS).

The two disadvantages of the HAC are

1. Very dependent on proper startup
2. It's possible that coincidental clusters will emerge.

K-Means Clustering:

The objective of the K-Means Clustering algorithm is to group similar elements or data points into a cluster. It is a simple clustering algorithm & "K'' in K-Means represent the number of clusters.

Dimensionality Reduction:

What if a dataset has more features than there are features? It gets increasingly difficult to visualize and then work on the practice set. Most of these features are likely to be linked and so redundant. Dimensionality reduction strategies are employed in these cases. Dimensionality reduction is the process of establishing a collection of primary variables in order to reduce the number of random variables under consideration.

The objective is to keep as much information as feasible while reducing the dataset's dimensionality. This is accomplished by feature selection and feature extraction, in which the feature space is reduced by removing features, then creating new independent features based on old independent variables, and finally removing the least important variables.

Principal Components Analysis (PCA):

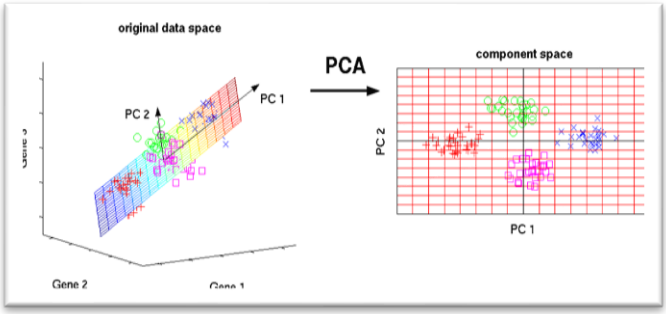

Fig.2

Fig.2: *The figure shows the transformation of a high dimensional data (3 dimension) to low dimensional data (2 dimension) using PCA [2]*

PCA is a method for transforming a dataset's columns into a new collection of characteristics known as Principal Components. This successfully compresses a huge portion of the data throughout the whole dataset into fewer feature columns. This allows for dimensionality reduction and the visualization of any class or cluster separation.

II. PROCEDURE

Hierarchical Agglomerative Clustering or bottom-up clustering began with an individual cluster, i.e., each data point is regarded an individual cluster, also known as a leaf. Then, for each cluster, the distance between them is calculated. The two clusters with the smallest distance between them would combine to form what we called a node. Newly created clusters calculate the member of their cluster distance with another cluster outside of their cluster once again. The process is continued until all of the data points are allocated to a single cluster termed root. The ultimate result is a tree-based representation of the items known as a dendrogram. Hierarchical Agglomerative Clustering does not provide a precise number of how our data should be grouped. It is up to us to choose the cut-off point.

The important stages in Hierarchical Agglomerative Clustering:

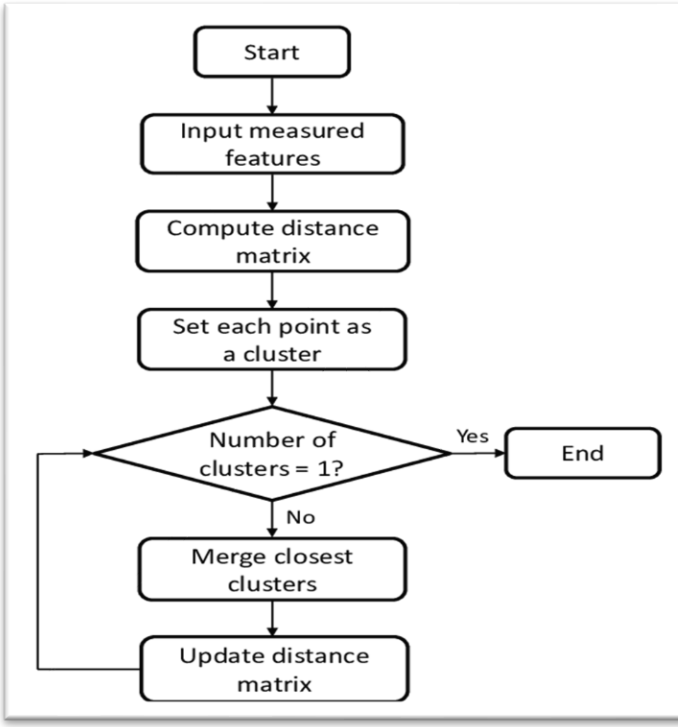

Fig. 3

***Fig 3**: The figure shows the flow chart representation of the multiple stages in Hierarchical Agglomerative Clustering.[3]*

1. Preparing the data: To initiate the operation we need to process the data in the form of a numeric matrix, with rows representing columns representing observations (individuals); and columns representing variables.

2. Measuring the distance between the data points: To measure the distance between the data points I used one of the most common calculation called Euclidean distance calculation.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Fig. 4

***Fig 4**: The figure shows Euclidean distance calculation [4].*

In basic terms, Euclidean distance is the length of a straight line between points x and y.

3 Linkage creation: The linking criterion determines where the distance is measured. It is a rule that we devise in order to determine the distance between clusters.

$$D(X, Y) = \min_{x \in X, y \in Y} d_{x,y}$$

Fig. 5

***Fig 5**: The figure shows Minimum distance calculation [5].*

- Where the distance between clusters X and Y is determined as the shortest distance between x and y, which are members of the X and Y clusters, respectively.

4. Determining the number of clusters: The easiest technique to determine the cluster number is to look at our dendrogram and choose a certain value as our cut-off point (manual way). Typically, we select the cut-off point those results in the highest vertical line.

The steps involved in PCA:

1. Standardizing each column: Standardization is mandatory before performing PCA as it is very sensitive to the variances. Thus, the goal of standardization is to make all the feature spaces have mean = 0 and variance = 1.

2. Covariance Matrix Computation: Covariance is used figure out how two variables are related to each other i.e., to determine if the variables are moving in the same direction with respect to each other or not. If covariance is positive, it means that if one variable increases, then the other increases as well. The opposite becomes true when covariance is negative. The covariance matrix is used to calculate the covariance of all possible combinations of columns resulting in the formation of a square matrix.

3. Computing Eigen values and Eigen Vectors: The amount of variance is explained using Eigen values and Eigen vectors. It is also used to understand the relation between the columns. The magnitude of Eigenvectors is one.

4. Deriving Principal Component Features: The Principal Component features are obtained by performing dot product of the standardized columns and eigen vector.

III. RESULTS

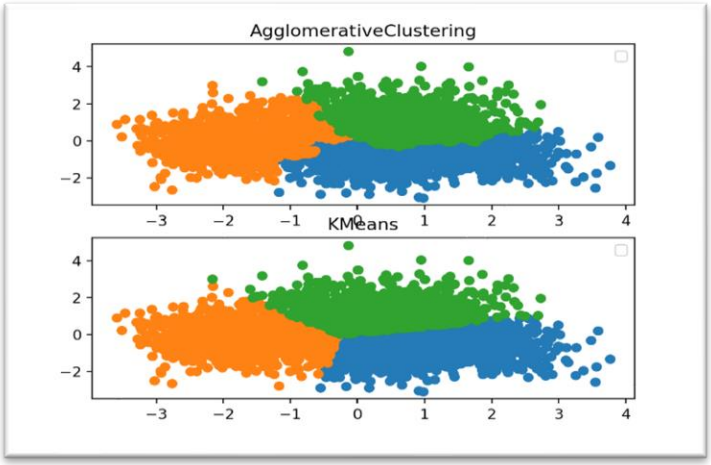

Fig 6

Fig 6: *The figure shows the plot of Agglomerative clustering vs K means clustering using PCA for K =3.*

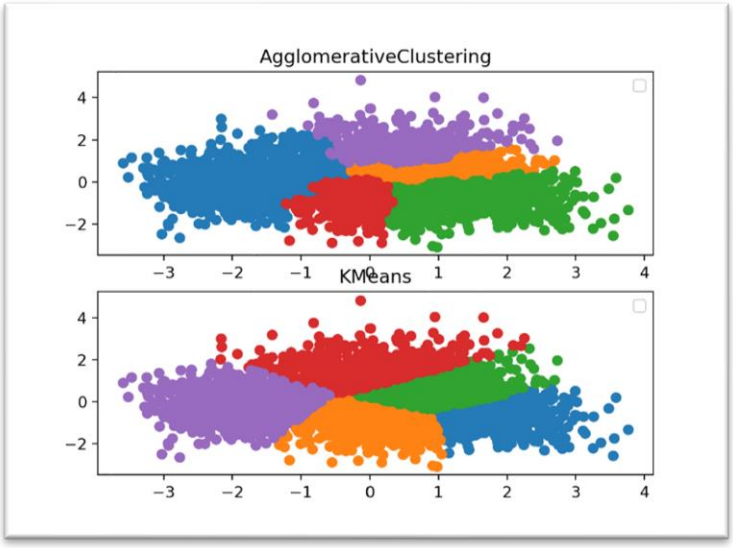

Fig 7

Fig 7: *The figure shows the plot of Agglomerative clustering vs K means clustering using PCA for K =5..*

IV. CONCLUSION

In comparison to K-Means clustering, Hierarchical Agglomerative Clustering provides insight on the closeness of all data points via a dendrogram. This strategy becomes much more significant when we extend clustering to a multi-dimensional space and use more than two characteristics.

V. REFERENCES

[1].https://link.springer.com/content/pdf/10.1007/s00357-014-9161-z.pdf

[2].https://www.researchgate.net/figure/Flow-chart-of-agglomerative-hierarchical-clustering_fig1_313238175

`

Appendix:

```
#Manohar Akula
#ASU ID: 1223335191
#EEE 511- Extra Credit Assignment

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn import metrics


data = np.array(pd.read_csv(r'C:\Users\akula\OneDrive\Desktop\yourdata.csv'))
scaler = StandardScaler()
data = scaler.fit_transform(data)
pca = PCA(2)
data = pca.fit_transform(data)

for no_of_clusters in [3,5]:
    X = data
    model = AgglomerativeClustering(n_clusters= no_of_clusters,
affinity='euclidean', linkage='ward')
    model.fit(X)
    labels = model.labels_
    ax1 = plt.subplot(2, 1, 1)
    for k in range(0,no_of_clusters):
        ax1.scatter(X[labels== k, 0], X[labels==k, 1])
    ax1.legend()
    ax1.set_title("AgglomerativeClustering")

    X = data
    kmeans = KMeans(n_clusters=no_of_clusters)
    kmeans.fit(X)
    predictions = kmeans.predict(X)
    frame = pd.DataFrame(X)
    frame['C'] = predictions
    frame.columns = ['X1', 'X2', 'C']
    ax2 = plt.subplot(2, 1, 2)
    for k in range(0,no_of_clusters):
        X = frame[frame["C"]==k]
        ax2.scatter(X["X1"],X["X2"])
    ax2.legend()
    ax2.set_title("KMeans")
    plt.show()
```