

**HO CHI MINH CITY UNIVERSITY TECHNOLOGY AND EDUCATION**

🞠◊🞠◊🞠

**GVHD**

**TOPIC:HIERARCHICAL CLUSTERING**

**Project 1**

**GROUP: 02**

**SEMESTER : 1 – YEAR: 2018-2019**

**TP. HỒ CHÍ MINH –**

**Participating members list**

Group: 02

*Nguyen Tran Nguyet Minh*

*ID: 17110052*

*Tran Thuan Tuong Vy*

*ID: 17110093*

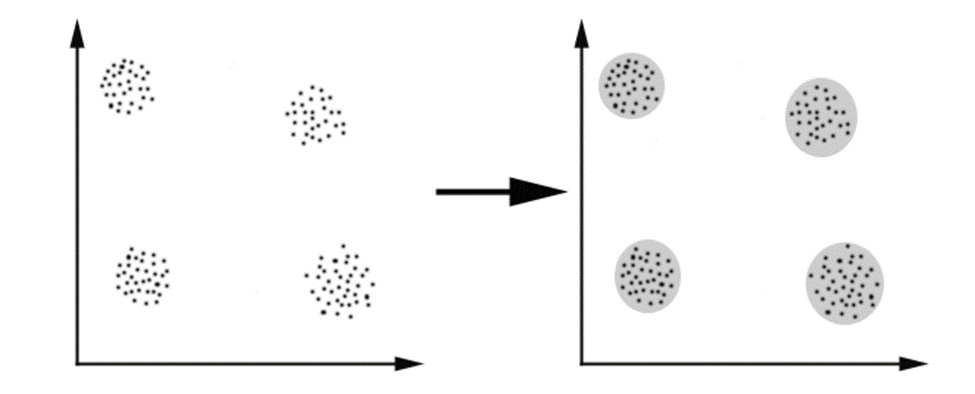
***Comment***

**CONTENT**

1. **Introduction**

***What is clustering?***

Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. We can show this with a simple graphical example:

****

In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance: two or more objects belong to the same cluster if they are “close” according to a given distance (in this case geometrical distance). This is called distance-based clustering.Another kind of clustering is conceptual clustering: two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

***The Goals of Clustering***

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.

For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding usefuland suitablegroupings (“useful” data classes) or in finding unusual data objects (outlier detection).

***Possible Applications***

Clustering algorithms can be applied in many fields, for instance:

Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records

Biology: classification of plants and animals given their features

Libraries: book ordering

Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds

City-planning: identifying groups of houses according to their house type, value and geographical location

Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones

WWW: document classification; clustering weblog data to discover groups of similar access patterns.

***Requirements***

The main requirements that a clustering algorithm should satisfy are:

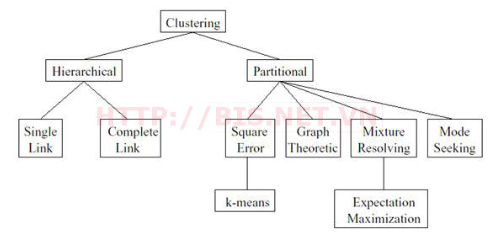
* Scalability
* dealing with different types of attributes
* discovering clusters with arbitrary shape
* minimal requirements for domain knowledge to determine input parameters
* ability to deal with noise and outliers
* insensitivity to order of input records
* high dimensionality
* interpretability and usability.

***Problems***

There are a number of problems with clustering. Among them:

* Current clustering techniques do not address all the requirements adequately (and concurrently)
* Dealing with large number of dimensions and large number of data items can be problematic because of time complexity
* The effectiveness of the method depends on the definition of “distance” (for distance-based clustering)
* If an obvious distance measure doesn’t exist we must “define” it, which is not always easy, especially in multi-dimensional spaces
* The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

Clustering techniques are classified as follows:



**2. Algorithm**

Clustering algorithms may be classified as listed below:

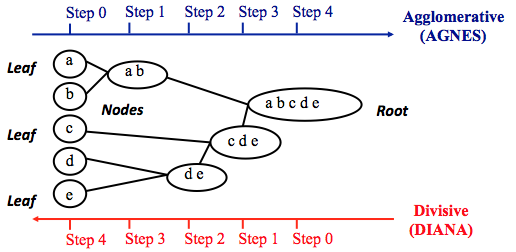
* Exclusive Clustering
* Overlapping Clustering
* Hierarchical Clustering
* Probabilistic Clustering

1. **Hierarchical clustering**
2. **What is hierarchical clustering?**

This is such a useful technique for the first stages of data analysis, helping us understand the dataset which is being surveyed by identifying patterns and the relationship between attributes. Strategies for hierarchical clustering generally fall into two types:

• Agglomerative: This is a "bottom-up" approach: each data point is a single cluster, and pairs of clusters are merged as one moves up the hierarchy. The two clusters that are the most similar are combined into a new bigger cluster (nodes) when they moves up the hierarchy. This procedure is iterated until all points are member of just one single big cluster (root).

• Divisive: This is a "top-down" approach: It begins with the root, in which all objects are included in a single cluster. When it moves down the hierarchy, the most heterogeneous cluster is divided into two. The process is iterated until all objects are in their own cluster.



We work out appropriate method based on the amount of data we have collected. Agglomerative clustering is good at identifying small clusters. Divisive clustering is good at identifying large clusters.

1. **Steps to agglomerative hierarchical clustering**

***Data structure and preparation***

The data should be a numeric matrix with:

* rows representing observations (individuals);
* columns representing variables.

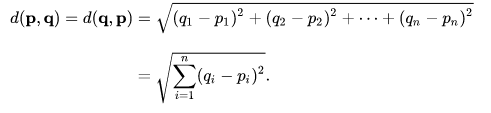
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Murder | Assault | UrbanPop | Rape |
| Alaska | 0.5079 | 1.107 | -1.212 | 2.48420 |
| Arizona | 0.0716 | 1.479 | 0.999 | 1.04288 |
| Arkansas | 0.2323 | 0.231 | -1.074 | -0.18492 |
| California | 0.2783 | 1.263 | 1.759 | 2.06782 |
| Colorado | 0.0257 | 0.399 | 0.861 | 1.86497 |

***Similarity measures***

In order to decide which objects/clusters should be combined or divided, we need methods for measuring the similarity between objects.

With "bottom-up" approach, we need to calculate the similarity information. Opposite, with "top-down" approach, we calculate dissimilarity information. There are many methods to help us do this such as Euclidean and Manhattan distances.

* Euclidean distance is the "ordinary" distance between two points that can be measured with a ruler, and is calculated using the Pythagoras formula.



* Manhattan distances is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes.

For example, we computes the Euclidean distance between objects:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Alaska | Arizona | Arkansas | California | Colorado |
| Alaska | 0 | 2.70 | 2.83 | 3.01 | 2.33 |
| Arizona | 2.70 | 0 | 2.72 | 1.31 | 1.37 |
| Arkansas | 2.83 | 2.72 | 0 | 3.76 | 2.83 |
| California | 3.0 | 1.31 | 3.76 | 0 | 1.29 |
| Colorado | 2.33 | 1.37 | 2.83 | 1.29 | 0 |

***Linkage***

The linkage function takes the distance information and groups pairs of objects into clusters based on their similarity. Next, these clusters are linked to each other to create bigger clusters. This process is iterated until all the objects in the original data set are linked together in a hierarchical tree.

There are many cluster linkage methods (the method to be used for computing distance between clusters).

* Maximum or complete linkage: The distance between two clusters is defined as the maximum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce more compact clusters.
* Minimum or single linkage: The distance between two clusters is defined as the minimum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce long, “loose” clusters.
* Mean or average linkage: The distance between two clusters is defined as the average distance between the elements in cluster 1 and the elements in cluster 2.

***Dendrogram***

Dendrograms is the graphical representation of the hierarchical tree.

In the dendrogram, each leaf corresponds to one object. As we move up the tree, objects that are similar to each other are combined into branches, which are themselves fused at a higher height.

The height of the level, provided on the vertical axis, indicates the (dis)similarity/distance between two objects/clusters. The higher the height of the fusion, the less similar the objects are. This height is known as the cophenetic distance between the two objects.

1. **Verify the cluster tree**

If the clustering is valid, the linking of objects in the cluster tree should correlate strongly with the distances between objects in the original distance matrix. The nearer the correlation coefficient value is to 1, the more correctly your information is reflected by the clustering solution.  
Using a different linkage method creates a tree that represents the different distances.

1. **Cut the dendrogram into different groups**

One of hierarchical clustering problems is that, you don’t know how many clusters there are, or where to cut the dendrogram to form clusters.

To partition your information into clusters, you can cut the hierarchical tree at a specified height.

1. **Application of hierarchical clustering**

Hierarchical clustering is used to gene expression data analysis. It helps us find out whether there are groups of genes or groups of samples that have similar gene expression patterns.

The default distance measure for most common clustering software is the Euclidean distance. The most popular methods for gene expression data are to use log2(expression + 0.25), correlation distance and complete linkage clustering agglomerative-clustering.

In principle it is possible to cluster all the genes. But with big data like this, problems might appear. So they use some preliminary analysis to select genes for clustering.

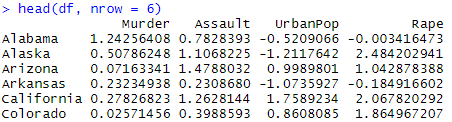
1. **Demo**

In this project, we use R language to describe the agglomerative clustering algorithms.

|  |
| --- |
| # Load the data  data("USArrests") |

We uses the dataset called “USArrests”. This is a public dataset available in R. Now, we’ll show the first 6 rows

|  |
| --- |
| head(df, nrow = 6) |

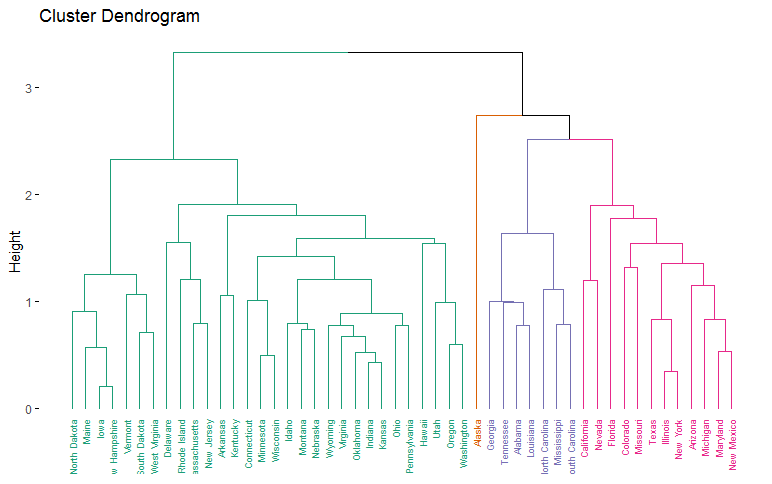


We use ‘Euclidean distance’ to calculates the distance between the data matrix rows; however, we can use the argument method to show other metrics. We try creating the hierarchical tree with average method

|  |
| --- |
| # Load Library  library("cluster")  res.agnes <- agnes(x = USArrests, # data matrix  stand = TRUE, # Standardize the data  metric = "euclidean", # metric for distance matrix  method = "average" # Linkage method  ) |

To display the hierarchical tree generated by the function hclust(), we can use the base function plot(res.hc). But to make Dendrograms observation easier, we use the function fviz\_dend() in factoextra R package.

|  |
| --- |
| fviz\_dend(res.agnes, cex = 0.5, k = 4, k\_colors = c("#1B9E77", "#D95F02", "#7570B3", "#E7298A")) |



|  |
| --- |
| # Compute cophentic distance  res.coph <- cophenetic(res.agnes)  # Correlation between cophenetic distance and the original distance  cor(res.dist, res.coph) |



To show the different between linkage methods, we try to use other method

|  |
| --- |
| res.agnes1 <- agnes(x = USArrests,  stand = TRUE,  metric = "euclidean",  method = "complete"  )  cor(res.dist, cophenetic(res.agnes1)) |



The correlation coefficient shows that using a different linkage method creates a tree that represents another original distance.

|  |
| --- |
| # Cut tree into 4 groups  grp <- cutree(res.agnes, k = 4)  head(grp, n = 4)  # Number of members in each cluster  table(grp) |





To present the result in a scatter plot, we also use function fviz\_cluster().

|  |
| --- |
| Mylist <- list(data = scale(USArrests), cluster = grp)  fviz\_cluster(mylist,  palette = c("#1B9E77", "#D95F02", "#7570B3", "#E7298A"),  ggtheme = theme\_minimal()) |

