Global Journal of human-social science: G
Linguistics & Education
Volume 16 Issue 3 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-460X & Print ISSN: 0975-587X

Agglomerative Hierarchical Clustering: An Introduction to Essentials. (1) Proximity Coefficients and Creation of a Vector-Distance Matrix and (2) Construction of the Hierarchical Tree and a Selection of Methods

By Refat Aljumily
University of Newcastle, United Kingdom

Abstract- The article is on a particular type of cluster analysis, agglomerative hierarchical analysis, and is a series of four main parts. The first part deals with proximity coefficients and the creation of a vector-distance matrix. The second part deals with the construction of the hierarchical tree and introduces a selection of clustering methods. The third deals with a variety of ways to transform data prior to agglomerative cluster analysis. The fourth deals with deals with measures and methods of cluster validity. The fifth and final part deals with hypothesis generation. The present article covers the first and second partsonly. It explains how agglomerative cluster analysis works by implementing it in a data matrix step by step.

Keywords: proximity, metric space, vector space, (non) euclidean space, symmetric matrix, agglomeration, centroid, sum of squares, median.

GJHSS-G Classification : FOR Code: 139999

Strictly as per the compliance and regulations of:

Agglomerative Hierarchical Clustering: An Introduction to Essentials. (1) Proximity Coefficients and Creation of a Vector-Distance Matrix and (2) Construction of the Hierarchical Tree and a Selection of Methods

Refat Aljumily

Abstract- The article is on a particular type of cluster analysis, agglomerative hierarchical analysis, and is a series of four main parts. The first part deals with proximity coefficients and the creation of a vector-distance matrix. The second part deals with the construction of the hierarchical tree and introduces a selection of clustering methods. The third deals with a variety of ways to transform data prior to agglomerative cluster analysis. The fourth deals with deals with measures and methods of cluster validity. The fifth and final part deals with hypothesis generation. The present article covers the first and second partsonly. It explains how agglomerative cluster analysis works by implementing it in a data matrix step by step. Different types of agglomerative hierarchical clustering methods are applied on purposely-made data matrix so different types of cluster structures are made from that same dataset. The last three parts will be covered in the next publication(s).There are many articles, tutorials, and books on this subject. The article has two main objectives: (1) to keep the discussion short and easy to understand by (hopefully) any reader and (2) to develop the motivation for using agglomerative hierarchical clustering to analyse any high-dimensional data of interest with respect to some research question.

Keywords: proximity, metric space, vector space, (non) euclidean space, symmetric matrix, agglomeration, centroid, sum of squares, median.

I. Introduction

Agglomerative Hierarchical Cluster Analysis, abbreviated (AHCA), is a particular type of cluster analysis and is a useful multivariate exploratory technique that has found application in different research fields such as data mining, social sciences, biology, information retrieval, statistics, pattern recognition, ecology and psychology. Agglomerative Hierarchical Cluster Analysis is not a single method but rather a family of different but related computational methods that make no a priori assumptions about the structure of data. Agglomerative Hierarchical Analysis methods try to discover structured interrelationships among data vectors that might be interesting in relation to a research purpose. More specifically, all the methods of the family try to identify and graphical display of structure in data when data is too large either in terms of the number of variables or of the number of objects described, or both, for it to be readily interpretable by direct inspection. Agglomerative Hierarchical Analysis methods generate hierarchically ordered clusters and represent proximity structure among objects in high-dimensional space not as a spatial cluster but as a constituency tree or dendrogram. All the methods work by grouping a set of objects in the domain of interest into distinct clusters according to how relatively similar/dissimilar those objects are in terms of the variables that describe them. Each object is described by a set of variables. Any two objects will be more or less similar/dissimilar on the basis of some definition of proximity between them.

This article is in four main parts. The first part gives a general description of agglomerative hierarchical cluster analysis and proposes an interpretation of the result related to it. The second part first provides some relevant mathematical concepts that will be used in agglomerative hierarchical clustering: cluster, metric space, vector space, and proximity matrix, and then goes into the detail of how proximity among pairs of vectors is measured and how a cluster tree is built. The third part shows twelve different varieties of agglomerative hierarchical analysis and applies them to a data matrix M. The final part concludes the discussion.

a) *Agglomerative Hierarchical Cluster Analysis (AHCA) and interpretation*

AHCA is known as a bottom-up or alternatively left to right approach. This approach is the more often used and also better covered in the relevant textbooks, e.g., [1], [2], and [3]. This is probably because AHCA provides more information than the other methods in that they not only identify the main clusters, but also their constituency relations relative to one another as

Author: School of English Literature, Language and Linguistics, University of Newcastle, Newcastle Upon Tyne, Tyne and Wear NE1 7RU, UK. e-mail: refat.aljumily@newcastle.ac.uk

well as their internal structures. The result of the utilization of AHCA is shown by a diagram called a 'constituency tree' or 'dendrogram', which groups together related data vectors based on the relativities of proximity among all pairs of data vectors. Figure/1shows the result from the application of AHCA to eight data vectors.

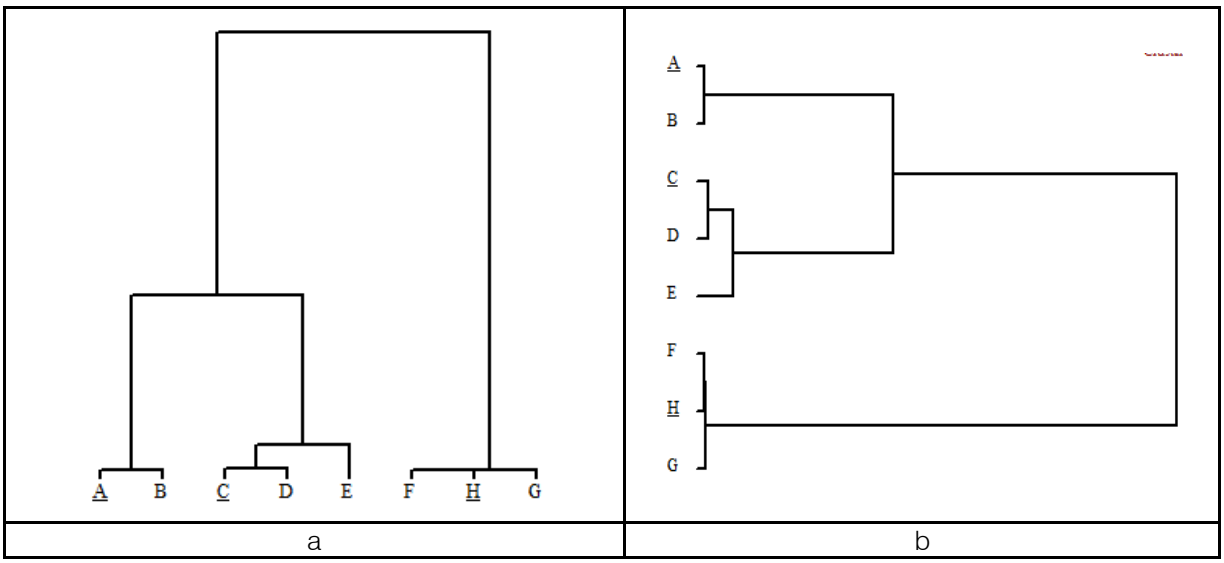


Figure 1 : (a) Vertical view of bottom-up tree and (b) horizontal view of left to right tree of five data items

Figure/1shows the cluster structure of eight data vectors as a hierarchical dendrogram. To interpret the dendrogram correctly one has to understand how it is constructed, so a short intuitive account is given here; technical details are given later in the course of the discussion. The dendrogram in this figure can be viewed in different ways, that is, either vertically (a) or horizontally (b). In it the letters at the leaves are labels for the vectors in the dataset: "A" is the first vector, "B" the second, and so on. These labels are agglomerated into clusters in a sequence of steps. AHCA treats each data vector as a single cluster on its own and then sequentially agglomerate pairs of clusters until all clusters have been agglomerated into a single larger cluster that contain all data vectors. The links included in the hierarchy represent the constituency structure for the entire dataset: vector "A" and vector "B" constitute a cluster (A B), vector "C" and vector "D" constitute a cluster (C D), which itself combines with vector "E" so constitutes a cluster ((C D) E) that are combined together with (A B) to form an even higher-level cluster ((A B) ((C D) E)), and so on. The lengths or heights of the links represent degrees of closeness: the shorter the link, the more similar the clusters. This is reflected in the cluster tree by the relative lengths of these links by the constituency structure of the proximity relations among, for example, vectors (A B)and vectors (F H) or vector (G). The longest (vertical/horizontal) lines at the top or right of the dendrogram separate the vectors into three main groups. The dendrogram represents vector proximity in n-dimensional space. For example, vector "F" and vector "H" are very close in the data space, and this pair is close to vector "G".

II. Space Concepts

a) Cluster Definitions

From cluster analysis viewpoint, the power of human eye or brain can recognize structures that are contained in data by perceiving any clusters in it, despite the fact that the clusters may vary somewhat in different viewpoints, in many different sizes and shapes or even when they are interpreted or understood. To accept such a view we have to understand what a cluster is. Indeed, humans can detect patterns or connections in any surrounding environment and can distinguish between them, and clusters are a kind of pattern. In a countryside position, for example, we can see clusters of trees, or farm buildings, of sheep. In any clear night we can see in the sky clusters of stars. And, closer to current interests, anyone looking at a data plot immediately sees any clusters that might be present. Looking at the data plotted in the two-dimensional space below, on the basis of our innate pattern recognition capability and without recourse to any obvious definition of the cluster, we can see that in figure/2a there is a random cloud of points with no clear structure emerging behind the data, and that in figure/2b there are some local areas of concentrations of points, but these are not explicitly defined. By contrast, we can clearly see that in figures/2c and 2d there is a clear structure: figure/2c shows three clusters of equal size, whereas figure/2d shows two clusters of unequal size, the smaller of which is in the upper-left part of the plot and the larger one in the lower part.

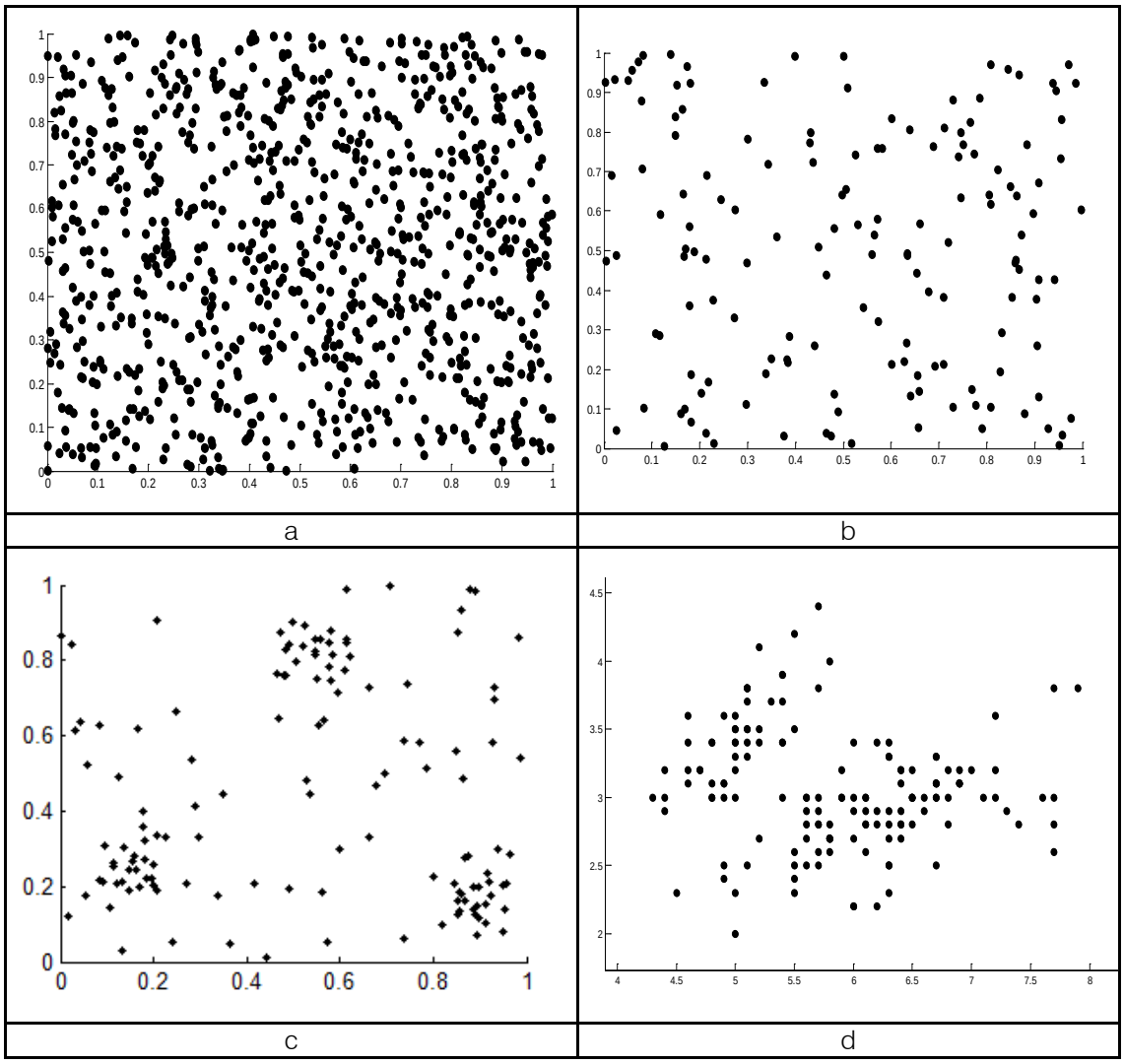


Figure 2 : A random scatter of points in two dimensions

The term cluster, however, does not have a precise definition, but there are some working definitions of what a cluster is that are commonly used. Three of them are given by [4] and [5]. They are:

- "A cluster is a set of entities which are alike, and entities from different clusters are not alike";
- "A cluster is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it";
- "Clusters may be described as connected regions of multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points".

The first definition of a cluster is a very general one and is best described as a similarity-based cluster definition. It assumes that objects are similar to each other within the same cluster and dissimilar to objects in different clusters. The second introduces the distance view of similarity and is best described as a distance-based cluster definition. It assumes that the similarity or dissimilarity between data vectors can be measured on the basis of the distance between them. The third definition of a cluster introduces density view of similarity and is best described as a density-based cluster definition. It assumes that each cluster is representing a given region that has its own demand distribution which symbolizes the data vectors enclosed by that region. This definition is more often used when the clusters are irregular or intertwined, and when noise and outliers are present [6].

Considering these three working definitions, we can see that even if the clusters consist of entities, points, or regions, the data vectors within each cluster are more similar in some respects than are other data vectors outside the clusters. A cluster is therefore a collection of data vectors which are similar between

them and are dissimilar to the data vectors belonging to other clusters. Figure/3 shows a sample data vectors plotted (left) with its corresponding clusters (right) on a two-dimensional scatter plot.

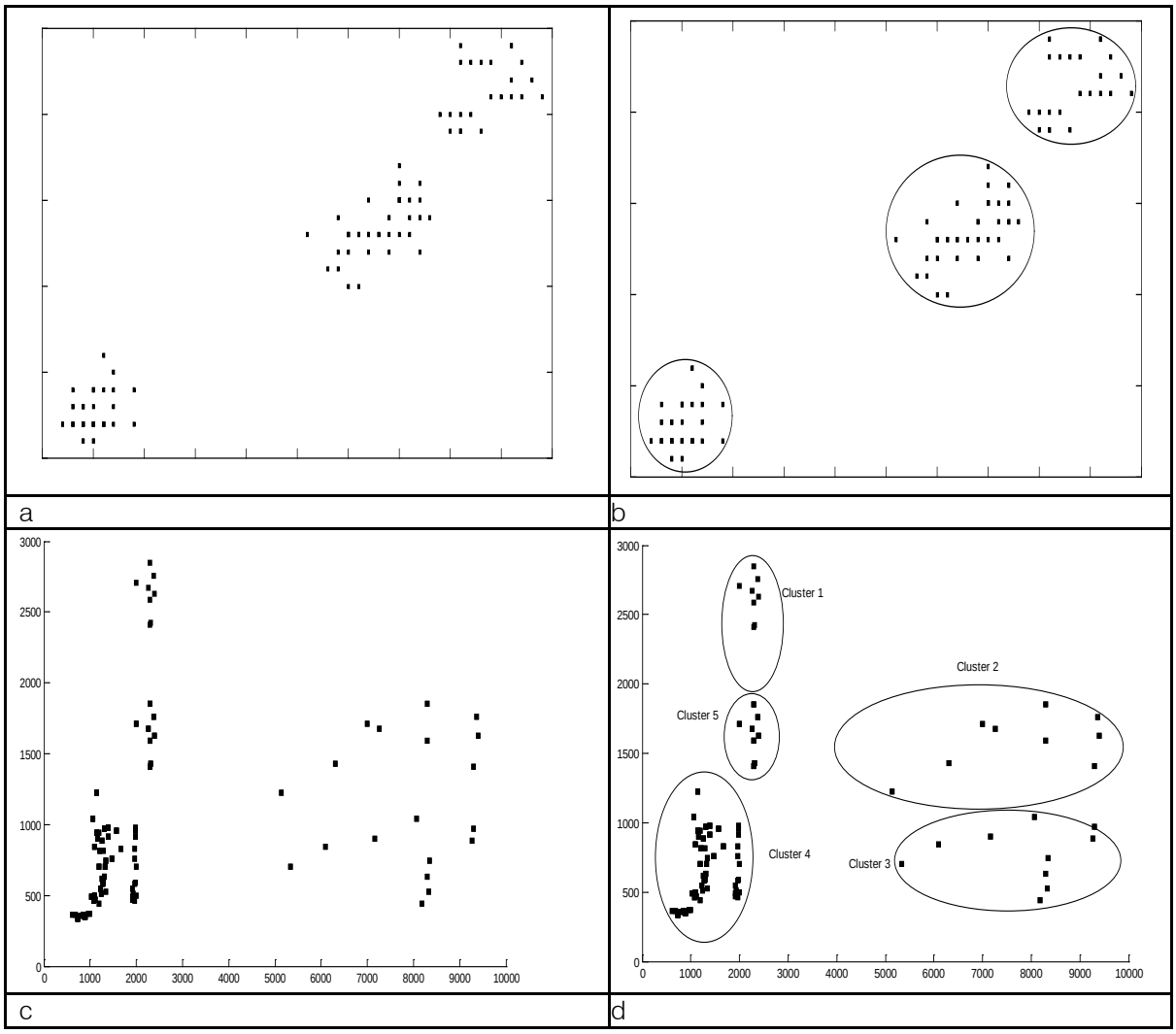


Figure 3 : A scatter of points (left) and its clusters (right) in two dimensions

In Figure/3a-b, the data vectors are clustered into three clear clusters labeled (cluster1, cluster2, cluster3) and in Figure (3c-d), the data vectors are clustered into five clear clusters labeled (cluster1, cluster2, cluster3, cluster4, cluster5) on the basis of some definition of proximity. If anyone is going to attempt an AHCA on data, then he/she should address the issue of what proximity coefficient to use at an early stage.

b) Proximity coefficient

Cluster analysis, by definition, is a process of identifying those data vectors that are similar and of establishing a hierarchical classification relationship among them on the basis of some index of proximity. What we mean by "on the basis of some index of proximity" is to calculate how data vectors plotted as points in multidimensional space are "close to" or "far away from" each other. To do so, we need to know the relative proximity between any two data vectors in different clusters.

A proximity coefficient is either a similarity or distance coefficient between every pair of data vectors in the space. The term proximity is more commonly used to refer to either one of these two coefficients. The term of proximity always suggest the question: proximity with respect to what? Most clustering procedures use pairwise measures of proximity. Two data vectors are close when their distance is small or their similarity is large. The choice of proximity coefficient is a crucial problem in cluster analysis [4]. The choice of which proximity measure to use in the first place is largely a matter of the type of data collected. All clustering information must be built up from the basic data types in the space. The type(s) of data collected in a given study determine the type of clustering analysis used. Most of the clustering algorithms can be applied to only certain kinds of data and some particular measures of

distance/similarity. As Everitt et al. [2] points out different proximity coefficients can and do lead to different cluster solutions, and as such it would be extremely useful to be able to select a proximity coefficient that is in some sense optimal. No reliable selection procedure exists, however. The choice of coefficient in any given application is governed by the nature of the data and by the clustering algorithms that will use it. There are many different types of data that one can collect. The following is a diagram showing some types of data that can be expressed either in terms of numbers or a natural language description.

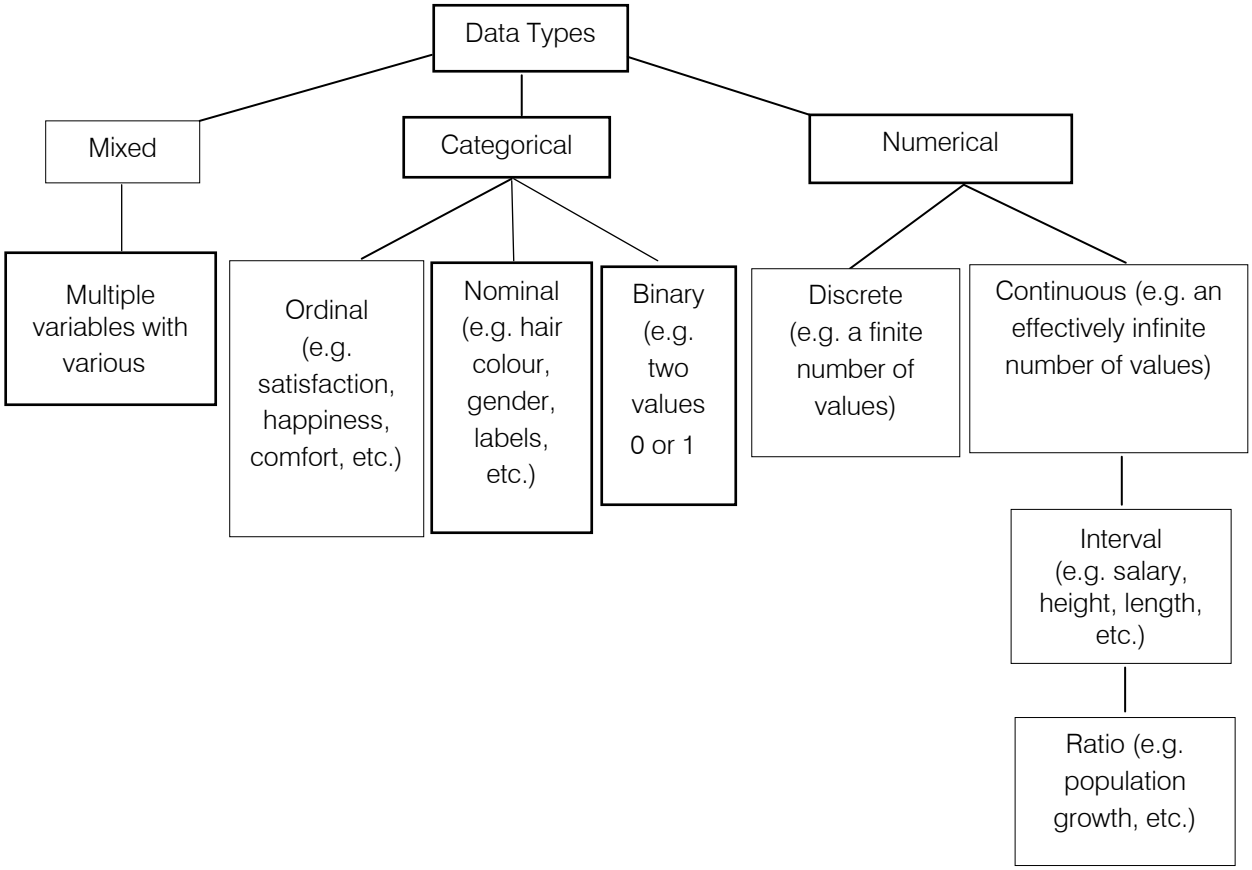


Figure 4 : Types of data

For details of each of the data types see, for example, [7], [8], and [9].

However, proximity between pairs of data vectors can be measured in terms of their correlation, of their similarity coefficient, of the angle between them, or of distance in Euclidean space [2]. With data in which all the variables are categorical, measures of similarity are most generally used. The most commonly used similarity coefficient, at least for binary data is the Jaccard similarity coefficient and is calculated as: $Sij=a/(a+b+c)$. To illustrate, Table/1 gives a matrix of binary variables of dimension 6 x 8.

Table 1 : A 6 x 8 data matrix

	Variables							
	*x*1	*x*2	*x*3	*x*4	*x*5	*x*6	*x*7	*x*8
Vectors A	1	0	0	1	0	0	1	0
B	0	1	0	1	0	0	1	0
C	1	1	1	1	0	0	1	1
D	0	1	1	1	0	0	0	1
E	0	1	1	1	0	0	0	1
F	0	1	0	0	0	0	1	1

Where each row vector is a student and the column vectors are binary tags or states of some student response, e.g. answer to test questions. The state (1) means a variable is present indicating a correct answer in the data vectors and (0) means it is absent indicating an incorrect answer. This data can be summed and placed in a contingency table in the form of the count of the number of the variables in each vector. The first two column data vectors (A) and (B) are worked out and the coefficient of matches among them are shown in Table/2.

Table 2 : A 2 x2 contingency table for the first two vectors in Table (1)

		Vector B	
		1	0
Vector A	1	*a*=2	*b*=1
	0	*c*=1	*d*=4

In this table, the rows represent the presence or absence of a set of *X* variables for a single student {*x*1, *x*2,……*x*8} for the first two row data vectors in Table/1. Cell *a* includes the count of the number of the *X* variables for which the two vectors both have the variable present. Cell *b* represents the number of variables the number of variables for which the first has the variables present and the second does no, and cell *c* includes the number of variables for which the second student has the variable present and the first student does not. Finally, cell *d* includes the count of the number of the *X* variables for which neither student has the variable present.

Now we compute the similarity coefficients between the students based on the coefficient of matches by using Jaccard similarity coefficient. The equation used to calculate the similarity between data vector A and B is the following:

$S_{AB}=a/(a+b+c)= 2/4= 0.500.$

Applying this equation to the other row data vectors:

$S_{AC}=a/(a+b+c)= 3/6= 0.500$

$S_{AD}=a/(a+b+c)= 1/6=0.167$

$S_{EF}=a/(a+b+c)= 2/5=0.400$

we obtain the following similarity matrix:

	A	B	C	D	E	F
A	0.000					
B	0.500	0.000				
C	0.500	0.500	0.000			
D	0.167	0.400	0.667	0.000		
E	0.167	0.400	0.667	1.000	0.0000	
F	0.200	0.500	0.500	0.400	0.400	0.000

Figure 5 : Jaccard Similarity Coefficient matrix for the six data vectors in the data matrix shown in Table /1

Jaccard Similarity Coefficient equates similarity with the three types of matches (a, b, c) only, excluding the coefficient of match 'd'. It, however, indicates maximum similarity when the two data vectors have identical values, in which case b=c=0 and $S_{AB=}$1.0. This coefficient also indicates maximum dissimilarity when there are no 1-1 matches, in which case a=0 and $S_{AB=}$0.0. The basic idea of similarity coefficient is to give relative similarity between data vectors. Two data vectors are similar, relative to the cluster membership, if their profiles across variables are "close" or they share "many" characteristics in common, relative to those which other pairs share in common. For the Jaccard similarity coefficient matrix, we obtain the following hierarchical tree:

Figure 6 : AHCA using Jaccard Similarity Coefficient

It can be seen that the data vectors (D) and (E) are mathematically most similar (or closest) to each other since they have identical matching coefficients (b and c = 0) and the similarity coefficient between them has the value of 1.0. Data vectors (A) and (B) are also similar since they share similar coefficient of matches. It can also be seen that the data vector (F) is very different from the others. (Note that because only simple data matrix have been used there are only two data vectors representing the two most similar cases that are closer to each other than any other pair in the data matrix).

The similarity coefficients depend on the selected agglomerative clustering method for constructing the hierarchical tree and thus may differ for different methods or different similarity coefficients. Look at the following dendrograms generated by different hierarchical clustering methods using the Jaccard similarity coefficient:

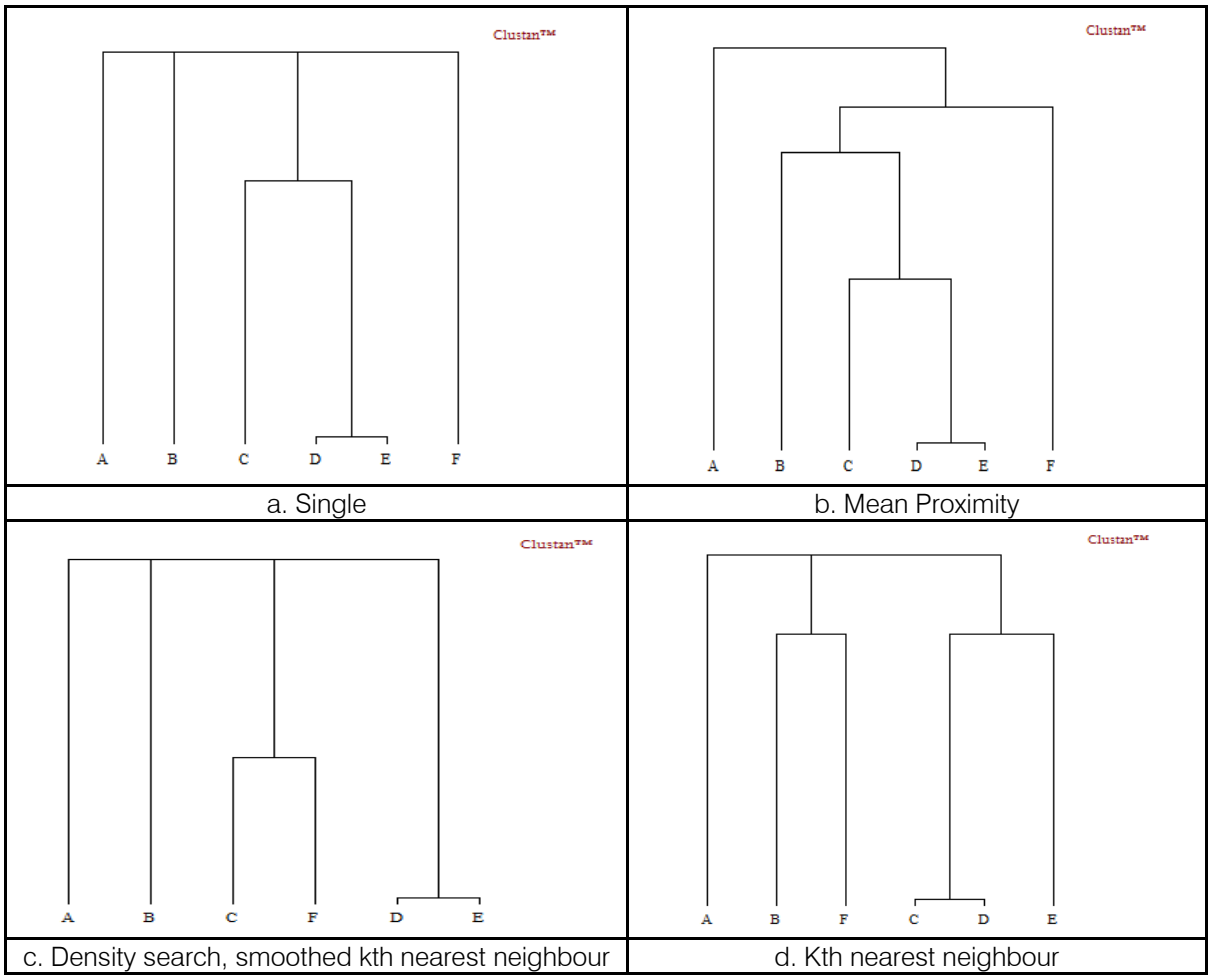


Figure 7 : AHCA four agglomerative hierarchical methods using Jaccard Similarity Coefficient applied to the matrix in Figure 5

More is said about all of these methods in due course; the important thing to realize at this stage is that Jaccard Similarity Coefficient was tried with Ward, Median, Centroid, and Sum of Squares, but the application showed that these methods are not defined for similarity coefficients. To work on it, however, similarity coefficient would have to be converted to dissimilarity by subtracting every value from the maximum similarity by using one of the standard conversion methods:

$$dissim_{ij} = \sqrt{sim_{ii} + sim_{jj} - 2sim_{ij}}$$

(Note when you subtract from the maximum we invert the scale so that previously small values are large. Another way to invert the scale is to multiply the similarity values by minus one, creating dissimilarity values).

The possible similarity coefficients of pairwise similarity are many, and these, together with their equations and properties, are available in, for example, [2], [3], and [10].

Table 3 : various similarity coefficients

Similarity Coefficient	Equation
Matching coefficient	$Sij=(a+d)/(a+b+c+d)$
Jaccard coefficient (Jaccard 1908)	$Sij=a/(a+b+c)$
Rogers and Tanimoto (1960)	$Sij=(a+d)/[a+2(b+c)+d]$
Sneath and Sokal (1973)	$Sij=a/[a+2(b+c)]$
Gower and Legendre (1986 A)	$Sij=(a+d)/[a+1/2(b+c)+d]$
Gower and Legendre (1986 B)	$Sij=a/[a+1/2(b+c)]$
Yule coefficient	$Sij=ad-bc/ad+bc$

Hamann coefficient	$Sij=(a+d)-(b+c)/(a+d)+(b+c)$
Sorenson coefficient	$Sij=2a/2a+b+c$
Rusell and Rao coefficient	$Sij=a/a+b+c+d$

However, when all the selected variables are numerical (continuous or discrete), distance between all pairs of data vectors is commonly computed by using a suitable distance coefficient. A distance coefficient is a measure which defines a distance between vectors of a set of data and it is typically termed metric space if it achieves the metric (triangular) inequality. Ideally, every distance measure should be a metric if the following conditions are satisfied:

$d(x,y) \geq 0$: this condition defines a positive-definite function, saying that distance can't be negative.

$d(x,y)=0$ if $x=y$: this condition says, as above, that distances are always positive except where the data vectors are identical in which case the distance is necessarily 0.

$d(x,y)=d(y,x)$: this condition says that the distance from x to y is the same as the distance from y to x, i.e. the distance is symmetric.

$d(x,z) \leq d(x,y)+d(y,z)$: this condition is called the triangle inequality which says that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side. The triangle inequality can only be an equality if the remaining side lies exactly on the line connecting the two sides.

In mathematics, a metric space is a set for which distances between all data vectors in the set are defined. These distances, taken together, are called a metric on the set. A distance coefficient is said to have the Euclidean property or to be Euclidean if it always produces distance matrices that are fully embedded in a Euclidean space (i.e. points in space). If a distance matrix is Euclidean then it is also metric but the converse does not follow. Non-Euclidean distances are of different kinds: some still satisfy the metric inequality but have no Euclidean representation (e.g. City block distance), while others are not (e.g. Bray-Curtis distance). The application of these distance measures in agglomerative clustering still makes very good sense as a distance measure between different objects. Discussions on non-Euclidean distances and their applications can be found in, e.g. [11] and [12].

However, choices for some of these distance coefficients are given in the following table that summarizes their equations and properties:

Table 4 : Distance coefficients

Distance coefficient	Description
Squared Euclidean Distance	This measures the distance d between two data vectors i and j, and is expressed as: $$d_{ij}^2 = S_k \frac{W_{ijk}\ (x_{ik}-x_{jk})^2}{S_k\ w_{ijk}}$$ where: X_{ik} is the value of variable k in data vector i, and W_{ijk} is a weight of 1 or 0 depending upon whether or not the comparison is valid for the kth; if differential variable weights are specified. It is the weight of the kth variable, or 0 if the comparison is not valid.
Euclidean Distance	This measures the distance d_{ij} which is obtained by taking the Square root of Squared Euclidean Distance d_{ij}^2 as calculated above.
Euclidean Sum of Squares	The Euclidean Sum of Squares (ESS) EP for cluster P is expressed by: $$E_p = S_{iep} c_i\ S_j \frac{W_j (x_{ij}-mp_j)^2}{S_j\ w_j}$$ Where: X_{ij} is the value of variable j in data vector i within cluster P Ci is an optional differential weight for data vector i W_j is an optional differential weight for variable j m_{pj} is the mean of variable j for cluster P The total ESS for all clusters P is thus $E=S_pE_p$ and the increase in the Euclidean Sum of Squares $I_p E_q$ at the union of two clusters p and q is: IpEq = Ep Eq-Ep-Eq
City Block Distance	City Block Distance, or the Manhatten metric distance, is the Sum of the distances on each variable and is expressed as:

	$d_{ij} = S_K \frac{W_{ijk} \left\| x_{ik} - x_{jk} \right\|}{S_k \; w_{ijk}}$
Product-Moment Correlation	Pearson's correlation coefficient gives the correlation coefficient distance between vectors A and B, and is expressed as: $S_{i,j} = \sum_{K-1}^{N} (C_{k,i} - \overline{C}_i)(C_{k,j} - \overline{C}_j) \Big/ \sqrt{\sum_{K-1}^{N} (C_{k,i} - \overline{C}_i)^2 \sum_{K-1}^{N} (C_{k,j} - \overline{C}_j)^2}$

These distances are closely related, and if all the variables are measured on the same scale or have been transformed or standardized, there is no particular reason to prefer one over another. But if all the variables are measured on the different scale or if the data comprise different variables, then it is important to select the most appropriate proximity coefficient prior to clustering. Detailed discussion on distances in vector space can be found in, e.g., [13] and [14].

c) Vector space

The central concept in agglomerative hierarchical clustering is data vectors in n-dimensional vector space. To understand how hierarchical clustering works, it is necessary to have a firm grasp of this concept. For the present purpose, the distance measure that is most commonly used, most straightforward to apply, and practically simple to understand, will be sufficient. This is the Euclidean distance, or straight-line distance, and almost everyone is familiar with, i.e. can be measured with a ruler.

A Euclidean vector space is a geometrical interpretation of a vector in which the dimensionality *n* of the vector defines an *n*-dimensional space, the sequence of numerical values comprising the vector specifies coordinates in the space, and the vector itself is a point at the specified Cartesian coordinates [1], [15], [16], and [17]. For example, a vector **v** = (2, 4) defines a two-dimensional space and its two components are coordinates in that space; a vector **v** = (2,4,6) defines a 3-dimensional space, and its values in the specified coordinate system place it at the corresponding position in the space; and so on to any dimensionality. This is shown graphically in Figure/8:

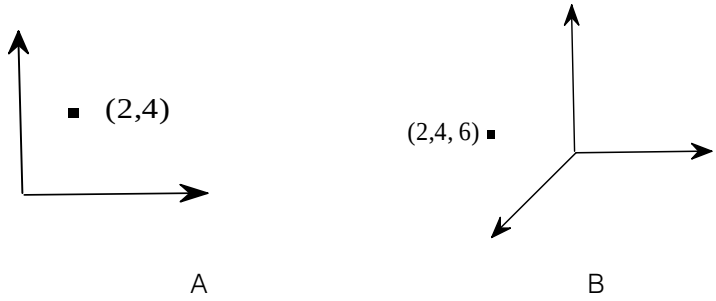


Figure 8 : 2 and 3-dimensional vector spaces

Any number *m* of vectors can exist in an *n*-dimensional vector space, where *m* corresponds to the number of rows in any given matrix M, and *n* corresponds to the number of columns.

d) Distance in vector space

In what follows, the generic term "proximity" is used to refer to the distance relations between and among pairs of vectors. This may be understood in the following ways.

To speak of a vector as a straight line, we see that if we draw a straight line from the origin (0,0) to the position of any point in the space of the axes (X,Y), the distance between the origin to that point is known as the length of a vector and can be measured as in Figure/9.

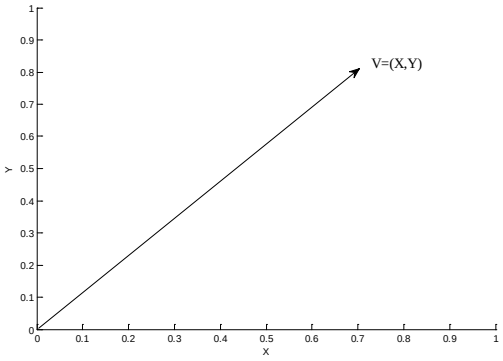


Figure 9 : A Vector in space

If we draw two straight lines from the origin (0,0) to the position of point A and B then we know that there are two vectors in the space and their lengths can be measured and compared. Two straight lines (vectors) are called equivalent (equal) if they have the same length, and unequal if they have different length. Thus the figure/10 shows that the length of vector A is greater than the length of B.

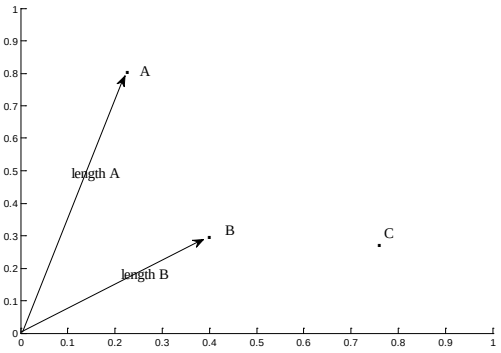


Figure 10 : Vector length

Because each vector is understood as a straight line determined by 2 points in the coordinate system, we may find the position of any vector if its coordinates are known (i.e. the position of vectors with reference to those two lines is known when we know their distances from the axes). Thus, in the figure/10 the position of the vector A is (0.2, 0.8) and vector (B) is (0.4,0.3).

Based on geometrical notions, we may state that the basic elements of vector space are length and angle. These can be used to determine the distance relations between and among vectors, and thus their cluster structure. To illustrate this, when two straight lines (or vectors) meet at a point in a space, there is an angle θ between them, as shown in the Figure/11 below.

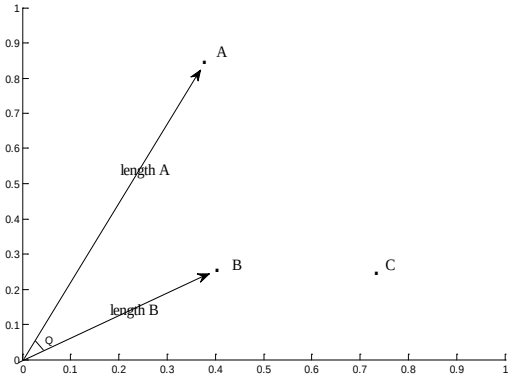


Figure 11 : The angle between vectors

After the length and angle are identified, the distance between two vectors can be measured and relative distances between pairs of vectors compared, so that distance (AC) in figure/12 is greater than distance (AB); this is the basis for several types of clustering method.

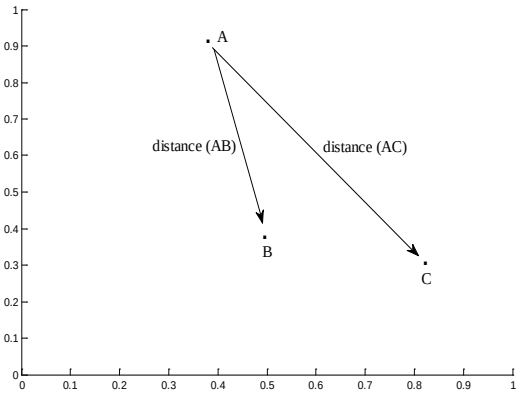


Figure 12 : Vector distances

The distance between any two vectors in a space is determined by the size of the angle between the straight lines meeting at the main point or origin of the space's coordinate system, and on the lengths of those lines. Suppose A and B to be any two vectors having identical lengths and separated by an angle θ (figure/13):

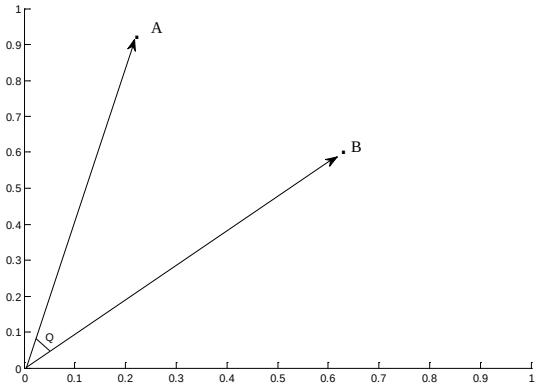


Figure 13

If the angle is fixed and the lengths of the vectors are not the same, then the distance between the two vectors A and B increases (figures/14a and 14b).

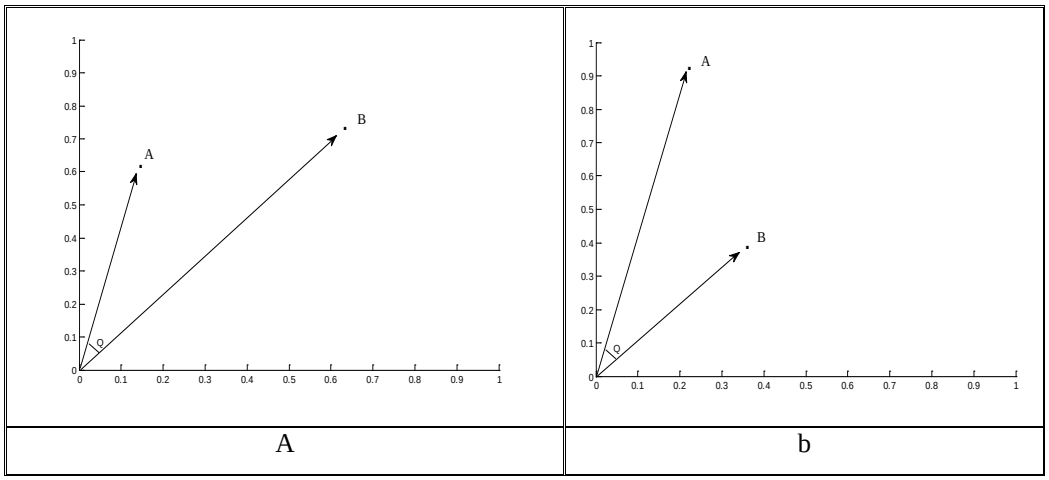


Figure 14

If the lengths of the vectors are the same but the degree of the angle is increased, the distance between the vectors increases (figure/15a), and if the degree of the angel is decreased, the distance is also decreased (figure/15b).

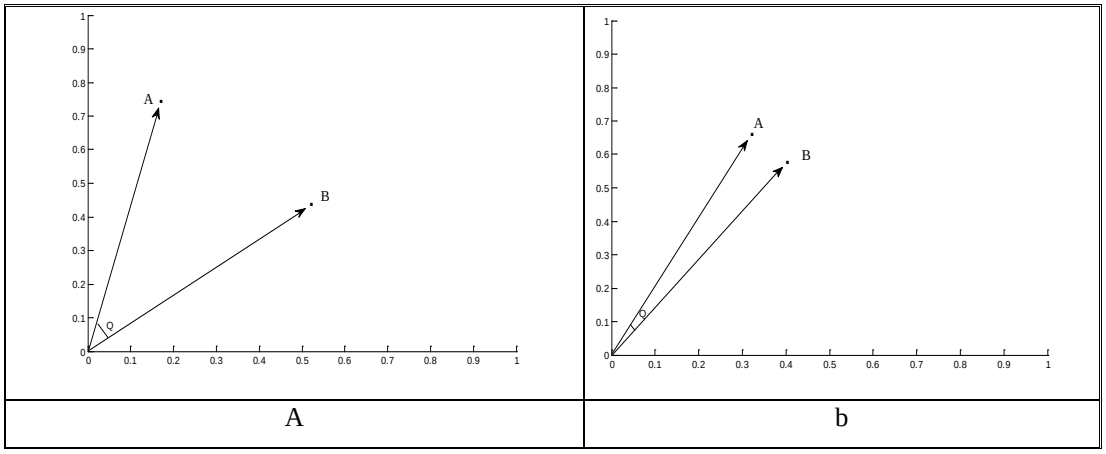


Figure 15

e) Distance in vector space

Most agglomerative hierarchical clustering methods however rely on the concept of distance among data vectors in n-dimensional space (data is represented in the form vectors of real numbers). Data vectors are grouped into similar or dissimilar clusters based on the information found in them: data vectors are considered similar if they are closer together and dissimilar if they are further apart in n-dimensional space. An intuition for how the measure of the distance between vectors in a vector space is best gained by working through a simple numerical example. Very often we use the equation for the Euclidean distance to quantify the distance in vector space. Consider the following triangle:

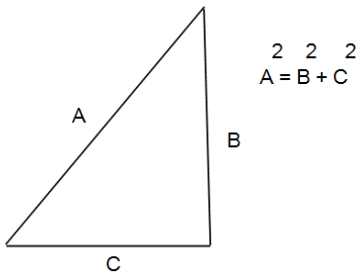


Figure 16 : Intuitive example

Here, the distance between the two points at the vertices of the triangle is:

$$\mathbf{length(A)} = \sqrt{(\boldsymbol{length(B)^2}} + \sqrt{(\boldsymbol{length\ (C)^2}}$$

The origin of this equation is in the Pythagorean Theorem. Pythagoras' theorem says that if we square the two shorter sides in a right-angled triangle and add them together, we get the same as when you square the longest side (the hypotenuse). In the triangle in Figure/16, (B) and (C) are the two shorter sides and (A) is the hypotenuse, so if we square (B) and (C) and add them together B^2+C^2 we get the same as if we square A (A^2). Therefore, $B^2 + C^2 = A^2$. Consider two points in 2- dimensional space:

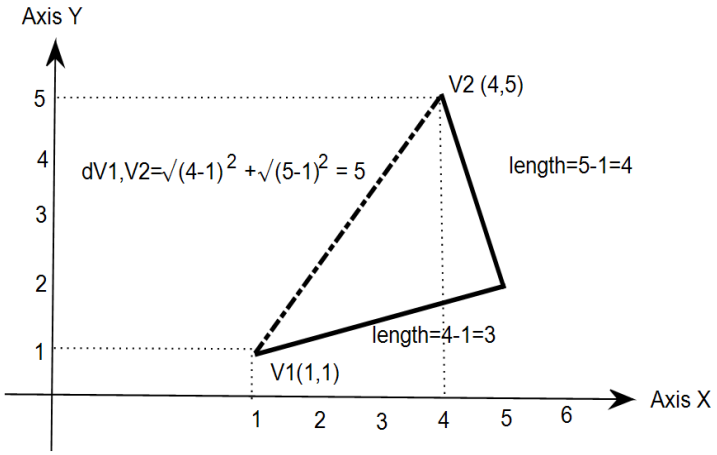


Figure 17 : Pythagoras' theorem applied to distances in two-dimensional space

The horizontal line (i.e. distance) goes from V1 at (1, 1) to V2 at (4,5), so it is obvious that its length $|X1-X2|$ is (4-1)=3 units. The vertical line or distance goes from V2 at (4,5) to (1,1), so again its length $|Y1-Y2|$ is obvious = 4 units. With this in mind, we get a right-angled triangle with lengths 3 and 4. By the Pythagorean theorem, the square of the hypotenuse is $(\text{hypotenuse})^2 = 3^2+4^2 = 25$, which gives the length of the hypotenuse as 25, same as the distance between the two vectors V1 and V2 according to the distance equation above. Thus the Euclidean distance between them is $dV1,V2 = \sqrt{(4-1)^2} + \sqrt{(5-1)^2} = 5$.

Various other distance measures are also possible as discussed above, but they needn't concern us here. Euclidean distance is the simplest and most widely used of the various distance measures. Euclidean distance is also best provided for in software implementations, and so is used here.

However, this quantification applies to any dimensionality n. That is, Euclidean distance applying Pythagoras' theorem can also be generalized or extended to measure the distance between any number of data vectors in any number of dimensions.

$$\boldsymbol{di,j} = \sqrt{(\boldsymbol{i1-j1})^2} + \sqrt{(\boldsymbol{i2-j2})^2} + \sqrt{(\boldsymbol{i3-j3})^2}$$

Look at the figure/18which shows 9 data vectors forming four triangles in 3-dimensional space, where each triangle is in its own space.

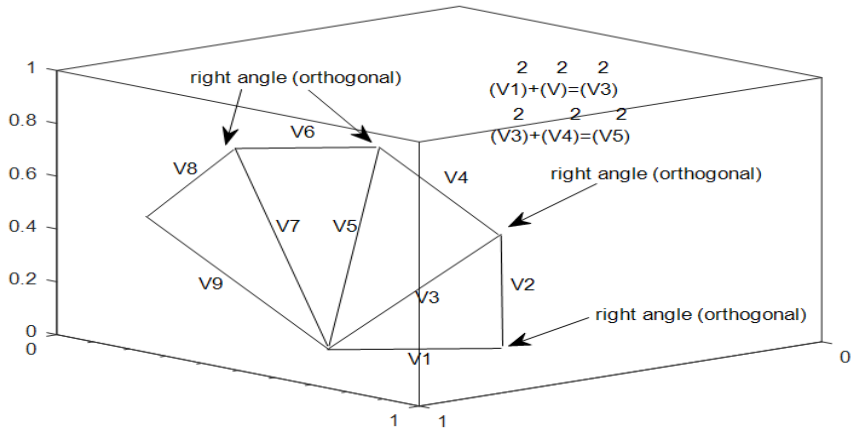


Figure 18 : 4 triangles in 3-dimensional space based on Pythagoras' theorem

More triangles can be found based on the distance measurements among the 9 data vectors but in this figure we limit the calculation to four triangles and the dimensionalities to three.

f) Distance matrix and agglomerative clustering

Because the above quantification of distance in vector space applies to any dimensionality, and not just to the 2 and 3-dimensional spaces that can be visualized, it can be used to define clusters in data of any dimensionality. This is what agglomerative hierarchical clustering does, and it does so in two steps:

i. *Construction of a distance matrix*

When all the distances between all possible pairs of data vectors are measured, they are gathered and entered in a distance matrix which looks like the Table 5:

Table 5 : Distance matrix based on Euclidean distance between 4 data vectors

	V1	V2	V3	V4
V1	0	2.828	3.162	5.99
V2	2.828	0	1.414	3.162
V3	3.162	1.414	0	2
V4	5.099	3.162	2	0

Looking at the distance matrix shows that all of the entries on the main diagonal are zero because the distance from a data vector to itself is zero and that the stored values in the triangle below the diagonal are mirror-images of the stored ones in the triangle above. The distance matrix is an n x n symmetrical, with rows and columns, on either side since the distance between V1 and V2 is identical to the distance between V2 and V1: the distance between any pair of vectors is the same in either direction.

ii. *Construction of a hierarchical tree based on the distance matrix*

Agglomerative hierarchal cluster analysis uses the quantified notion of distance described above, and the distance table more particularly, to find clusters in data. Numerous ways of doing this has been developed, most of them are variations on a theme; for present purposes the theme goes like this.

- For a data set containing m vectors, we start by defining m clusters, one for each vector.
- Using as many steps as necessary, at each step we combine the two clusters with the smallest distance between them into a new, composite (sub) cluster.

To understand this, consider the following data that consists of 14 two-dimensional points shown in Table 6.

Table 6 : a 14 x 8 data matrix

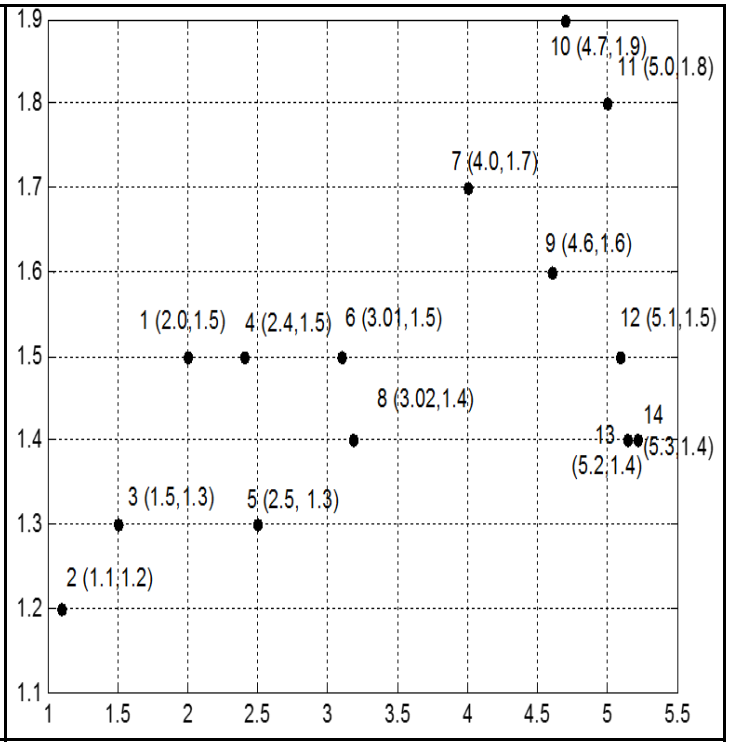
1	4	1.10	1.09	1.79	0.99	1.14	3.25
2	4	1.20	1.08	1.61	0.99	1.15	3.24
3	4	1.19	1.07	1.62	1.15	1.23	3.27
4	4	1.18	1.06	1.61	1.98	1.16	3.22
5	4	1.16	1.04	1.64	0.96	1.17	1.21
6	0.94	0.43	0.38	2.00	0.97	1.06	0.80
7	0.96	0.47	0.43	1.44	0.97	1.10	0.87
8	0.94	0.47	0.43	1.79	0.95	1.10	0.88
9	0.94	0.92	0.84	1.77	0.98	1.14	0.93

10	0.98	0.79	0.76	1.47	0.96	1.12	0.13
11	0.99	0.49	0.47	0.01	0.99	1.13	0.08
12	2.00	3.50	3.49	3.02	0.83	1.13	4.14
13	2.02	3.40	3.72	3.16	0.97	1.19	4.18
14	2.04	3.52	3.52	3.24	0.93	1.12	4.25

The x y coordinates of the points and the plots are shown in Figure/19:

Case	X	Y
1	2.0	1.5
2	1.1	1.2
3	1.5	1.3
4	2.4	1.5
5	2.5	1.3
6	3.0	1.5
7	4.0	1.7
8	3.02	1.4
9	4.6	1.6
10	4.7	1.9
11	5.0	1.8
12	5.1	1.5
13	5.2	1.4
14	5.3	1.4

a



B

Figure 19 : The xy coordinates of the 14 data vectors (right) of data matrix in Table/6 (left)

We calculate the Euclidean distance between all pairs of vectors as shown in Figure/17 above and construct the distance matrix for the 14 vectors. Figure/20 below is the one that we looked at above in Table/6 and it is repeated here for clarity of the indicated area between point 3 and 1.

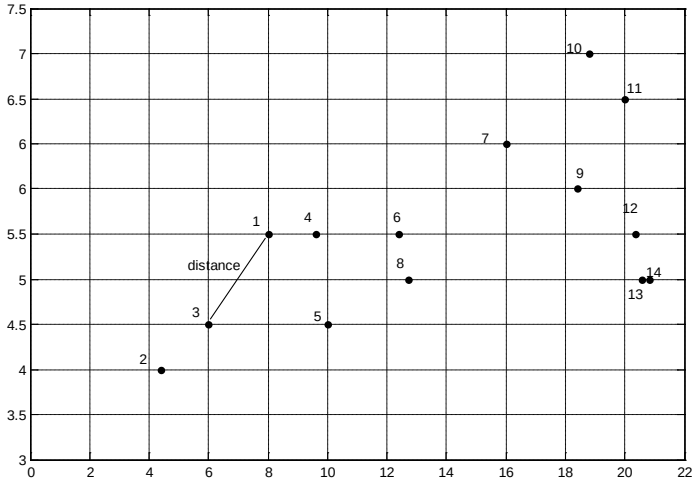


Figure 20 : The plot of the 14 data points

For this data matrix, we abstract the following distance matrix:

Table 7 : A distance matrix for the 14 data vectors

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.000	0.006	0.010	0.146	0.599	2.339	2.266	2.259	2.120	2.738	3.291	2.550	2.696	2.673
2	0.006	0.000	0.005	0.140	0.589	2.366	2.264	2.275	2.123	2.726	3.212	2.560	2.717	2.694
3	0.010	0.005	0.000	0.100	0.612	2.389	2.287	2.299	2.146	2.758	3.245	2.575	2.723	2.703
4	0.146	0.140	0.100	0.000	0.726	2.490	2.388	2.405	2.250	2.853	3.326	2.778	2.896	2.884
5	0.599	0.589	0.612	0.726	0.000	1.520	1.464	1.478	1.356	1.505	1.967	3.712	3.893	3.910
6	2.339	2.366	2.389	2.490	1.520	0.000	1.464	1.478	1.356	1.505	1.967	3.712	3.893	3.910
7	2.266	2.264	2.287	2.388	1.464	0.046	0.000	0.018	0.069	0.109	0.382	4.691	4.922	4.955
8	2.259	2.275	2.299	2.405	1.478	0.008	0.018	0.000	0.054	0.125	0.545	4.546	4.764	4.789
9	2.120	2.123	2.146	2.250	1.356	0.075	0.069	0.054	0.000	0.108	0.592	3.813	4.015	4.048
10	2.738	2.726	2.758	2.853	1.505	0.144	0.109	0.125	0.108	0.000	0.330	4.905	5.131	5.186
11	3.291	3.212	3.245	3.326	1.967	0.643	0.382	0.545	0.592	0.330	0.000	6.396	6.690	6.773
12	2.550	2.560	2.575	2.778	3.712	4.634	4.691	4.546	3.813	4.905	6.396	0.000	0.015	0.011
13	2.696	2.717	2.723	2.896	3.893	4.847	4.922	4.764	4.015	5.131	6.690	0.015	0.000	0.010
14	2.673	2.694	2.703	2.884	3.910	4.866	4.955	4.789	4.048	5.186	6.773	0.011	0.010	0.000

In what follows a 6 x 6 subset of the original 14 x14 distance matrix constructed in Table/7will be used. This makes it possible to show the whole process of constructing a hierarchical tree step by step rather than just a fragment, thereby baking the discussion clearer. The procedure is based on the principal that a set of data vectors has a cluster structure if it can be divided into two or more groups in which the members of any given group are close to one another in the data space, and far from members of other cluster in the space. At each step in tree construction, therefore, one looks for the clusters that are closest to one another and amalgamates them into a super ordinate cluster, and this continues until all the data vectors have been assigned to one of the clusters. The following discussion will demonstrate this.

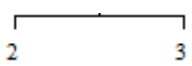
Initially, each row vector of the data matrix is taken to be a cluster on its own; i.e., clusters here and henceforth are shown in brackets. The distance matrix is now searched to find the smallest distance between these data vectors. This is the distance between vector 3 and vector 2 in Table 8: 0.005, shown shaded in Figure/21a. These are combined into a first agglomerated cluster (2, 3) by drawing the tree, as below, and then transforming the distance matrix to incorporate the first cluster.

	1	2	3	4	5	6
1	0.000					
2	0.006	0.000				
3	0.010	0.005	0.000			
4	0.146	0.140	0.100	0.000		
5	0.599	0.589	0.612	0.726	0.000	
6	2.339	2.366	2.389	2.490	1.520	0.000

a. Initial distance matrix reproduced from Table (8) with smallest distance highlighted

Cluster 1	Cluster 2	Agglomerated distance
(2)	(3)	0.005

b. Table of agglomeration



c. Graphical representation of (c)

Figure 21

Transformation of the distance matrix takes a bit of understanding, so it is described in detail.

- The table in figure 21 a is transformed into the one in figure 21b.
- Row vectors and column vectors are removed from the distance matrix and replaced them with a single blank row and column to represent the (2,3) cluster; 0 is inserted as the distance from (2,3) to itself.
- The minimum distances from (2,3) to the remaining data vectors (1), (4), (5), and (6) are inserted into the blank cells of the (2,3) row and column. Confused?

In the original distance matrix, the distance between (2) and (1) is 0.006 and between (3) and (1) is 0.010, shown shaded in figure/22a below. The minimum distance here is 0.006, and is inserted into the relevant cell representing the minimum distance between (2,3) and (1). The distance between (2) and (4) in the original distance matrix is 0.140 and between (3) and (4) it is 0.100. The minimum distance here is 0.100 and it is inserted into the relevant cell representing the distance between (2,3) and (4). The distance between (2) and (5) in the original distance matrix is 0.589 and between (3) and (5) it is 0.612.The minimum distance here is 0.589 and it is inserted into the relevant cell representing the distance between (2,3) and (5). The distance between (2) and (6) in the original distance matrix is 2.366 and between (3) and (6) it is 2.389.The minimum distance here is 2.366 and it is inserted into the relevant cell representing the distance between (2,3) and (6). Emendation of the distance table is now complete, and the resulting table is the basis for the next step in the construction of tree. Now the distance table is searched to find the smallest distance between vectors. This is the distance between vectors (2,3) and (1): 0.006. Vectors (2, 3) and (1) are now combined into a new subordinate cluster ((2,3),1) by drawing the tree as below, and then emending the distance table to incorporate the new cluster.

	1	2	3	4	5	6
1	0.000					
2	0.006	0.000				
3	0.010	0.005	0.000			
4	0.146	0.140	0.100	0.000		
5	0.599	0.589	0.612	0.726	0.000	
6	2.339	2.366	2.389	2.490	1.520	0.000

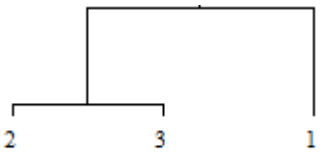
a. Distance matrix from Table (8)

	1	(2,3)	4	5	6
1	0.00				
(2,3)	0.006	0.00			
4	0.146	0.100	0.000		
5	0.599	0.589	0.726	0.000	
6	2.339	2.366	2.490	1.520	0.000

b. Transformed version of a.

Cluster 1	Cluster 2	Agglomerated distance
(2)	(3)	0.005
((2),(3))	(1)	0.006

c. Table of agglomeration



d. Graphical representation of (c)

Figure 22

We must note that the distance matrix has shrunk by one row and column. In any process of agglomerating clusters, this shrinkage will continue as we proceed.

Emendation of the distance table proceeds as step (1) explained above by removing the rows and columns and replacing them with single blank row and column to represent the new ((2,3)1) sub-cluster. Then the minimum distance from ((2,3),1) to the remaining data vectors (4), (5), and (6) is inserted into the blank cells. From Figure/ 22, the distance between (2,3) and (1) is 0.006 and between (4) and (1) is 0.146; the minimum distance is 0.006, and it is inserted into the relevant cell. The distance (2,3) and (5) 0.589 and between (1) and (5) is 0.599; the minimum distance here is 0.589, and it is inserted into the relevant cell. The distance between (2,3) and (6) is 2.366 and between (1) and (6) is 2.339; the minimum here is 2.339, and it is inserted into the relevant cell.

	1	2	3	4	5	6
1	0.000					
2	0.006	0.000				
3	0.010	0.005	0.000			
4	0.146	0.140	0.100	0.000		
5	0.599	0.589	0.612	0.726	0.000	
6	2.339	2.366	2.389	2.490	1.520	0.000

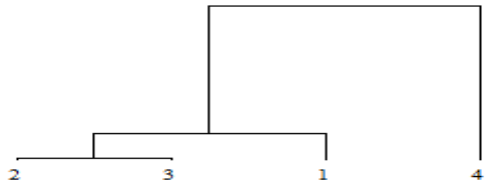
a. Distance matrix from Table (8)

	((2,3),1)	4	5	6
((2,3),1)	0.000			
4	0.006	0.000		
5	0.589	0.726	0.000	0.000
6	2.339	2.490	1.520	0.000

b. Transformed version of a.

Cluster 1	Cluster 2	Agglomerating distance
(2)	(3)	0.005
((2),(3))	(1)	0.006
(((2),(3)),(1))	(4)	0.006

c. Table of agglomeration



d. Graphical representation of (c)

Figure 23

The distance table is searched to find the smallest distance between vectors. This is the distance between vectors ((2,3),1) and (4): 0.006. Clusters ((2,3),1) and (4) are now agglomerated into a subordinate cluster (((2,3),1),4) as shown in the tree above, and then emending the distance matrix to incorporate the new cluster. Emendation of the distance matrix proceeds as in step 1 and 2. The rows and columns (2,3) and (4) are removed from the table and replaced them with a single blank row and column to represent the new (((2,3,4),1) cluster. The next step is to insert into the blank cells the ((2,3),1),4) to the remaining clusters (5) and (6). The distance between ((2,3),1) and (5) is 0.589 and between (4) and (5) is 2.726; the minimum is 0.589 and it is inserted into the relevant cell. The distance between ((2,3),1) and (6) is 2.339 and between (4) and (6) is 2.490; the minimum is 2.339 and it is inserted into the relevant cell. Here the smallest distance is 0.589 and thus clusters ((2,3),1),4) and (5) are now agglomerated into a subordinate cluster (((2,3),1),4),5) as shown in the tree below. The distance matrix is emended to incorporate the new cluster. Emendation of the distance table is now complete and the resulting matrix is the basis for the final step.

	1	2	3	4	5	6
1	0.000					
2	0.006	0.000				
3	0.010	0.005	0.000			
4	0.146	0.140	0.100	0.000		
5	0.599	0.589	0.612	0.726	0.000	
6	2.339	2.366	2.389	2.490	1.520	0.000

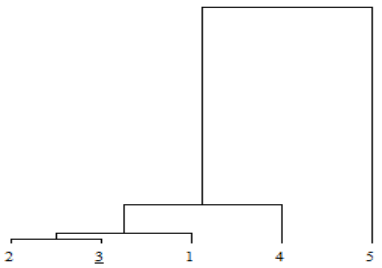
a. Distance matrix from Table (8)

	(((2,3),1),4)	5	6
(((2,3),1),4)	0.000		
5	0.589	0.000	
6	2.339	1.520	0.000

b. Transformed version of a.

Cluster 1	Cluster 2	Agglomerated distance
(2)	(3)	0.005
((2),(3))	(1)	0.006
(((2),(3)),(1))	(4)	0.006
(((2),(3)),(1),(4))	(5)	0.589

c. Table of agglomeration



d. Graphical representation of (c)

Figure 24

The minimum distance from (((2,3),1),4),5) to the remaining vector (6) is inserted into the blank cell of the (((2,3),1),4),5) column. The distance table generated in Figure/21 above is searched to find the smallest distance between vectors. There is only one remaining vector value. Clusters (((2,3)1,4),5) and (6) are now combined into a subordinate cluster (((((2,3),1),4),5),6) by drawing the tree and then emending the distance table to incorporate the new cluster.

	1	2	3	4	5	6
1	0.000					
2	0.006	0.000				
3	0.010	0.005	0.000			
4	0.146	0.140	0.100	0.000		
5	0.599	0.589	0.612	0.726	0.000	
6	2.339	2.366	2.389	2.490	1.520	0.000

a. Distance matrix from Table (8)

	(((2,3),1),4),5)	6
(((2,3),1),4),5)	0.000	
6	2.339	0.000

b. Transformed version of a.

Cluster 1	Cluster 2	Agglomerating distance
(2)	(3)	0.005
((2),(3))	(1)	0.006
(((2),(3)),(1))	(4)	0.006
(((2),(3)),(1),(4))	(5)	0.589
(((2),(3)),(1),(4),(5))	(6)	0.2339

c. Table of agglomeration

d. Graphical representation of (c)

Figure 25

All 6 data vectors have now been incorporated into the cluster tree and tree construction stops.

	(((2,3),1),4),5),6)
(((2,3),1),4),5),6)	0.000

Figure 26

In this example, we only obtained distance measurements and cluster agglomerations for only 6 data vectors from the original 14 x 8 data matrix of Table/7, because the calculation can become extremely long, it is important to emphasize that for a set of 14 data vectors there would be a total of 91 steps including the main diagonal zero-values. This can be given in relationship of the number of possible successive agglomerations: n (n-1)/2 where n is the number of data vectors. However, the steps explained above are repeated on the whole data matrix, and the result is shown in Figure/27:

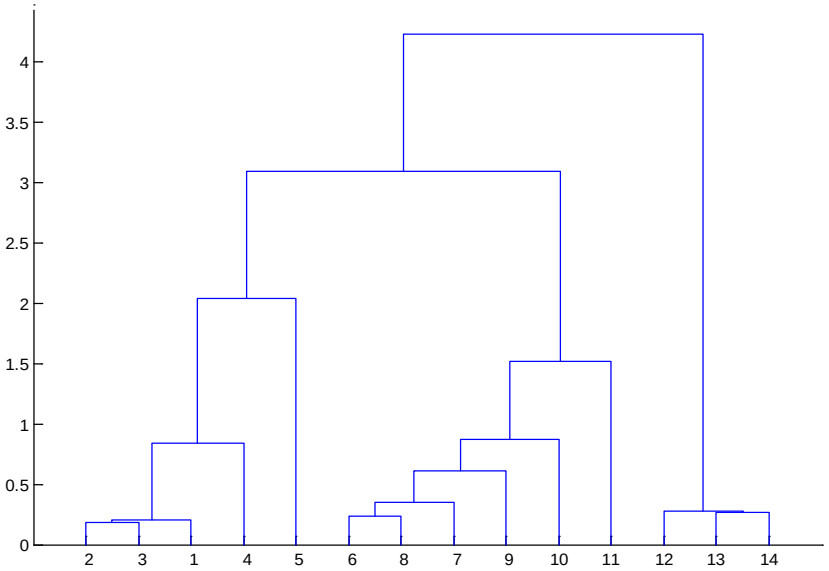


Figure 27 : Agglomerative hierarchical analysis for the 14 data vectors of the data matrix in Table 7

In this figure, the 14 vectors are represented as clusters and agglomerated together on the basis of the relativities of distance among them and the structure presented in a tree-like diagram. In this figure, all the 14 vectors are agglomerated into three main clusters which represent the relativities of distance among them as a dendrogram in figure/ 28a and their corresp.

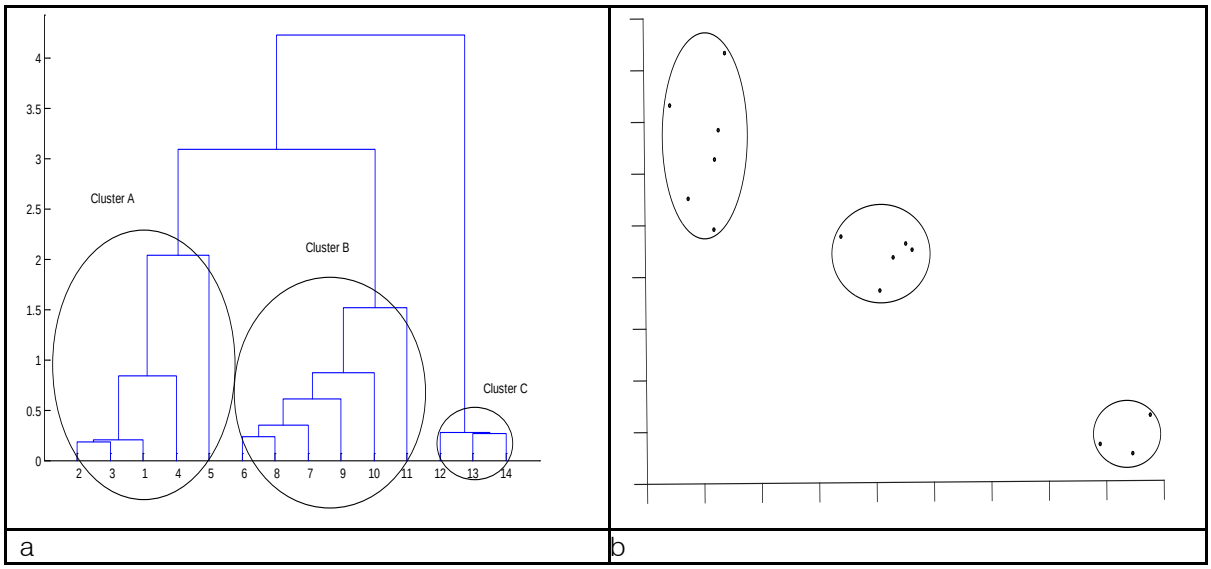


Figure 28 : Three main clusters for the 14 data vectors of the data matrix in Table 7

Given that the hierarchical clustering tree tells us nothing more than what the two-dimensional plot tells us, what is gained? In the current case nothing. The real power of agglomerative hierarchical cluster analysis consists in its independence of vector space dimensionality. Put it another way, direct plotting is limited to two, three, or fewer dimensions but there is no dimensionality limit on agglomerative hierarchical cluster analysis. It can determine relative distances in vector spaces of any clustering and represent those distance relativities as a dendrogram like the one above.

g) Agglomerative Hierarchical Clustering Methods

Many agglomerative clustering methods are treated as variations on a single major approach; they require the data to be in the form of vectors of real numbers and follow the same standard framework: Initially, before clustering has begun, each data vector is treated as a cluster or group, clustering begins by a successive agglomeration of the two closest or nearest pair of clusters (i.e. the two data vectors that are separated by the smallest distance) to form first cluster. The process of agglomerating two data vectors and fusing their characteristics is repeated until only one cluster remains. Extensive range of agglomerative clustering methods exists; though most of them operate in a similar way, their calculation is different. Eleven of these methods are introduced. They are:

- Single linkage (or nearest neighbor) method

In this method, the distance between two clusters A and B is based on the membership (i.e. data vectors) in each cluster that are nearest together (shortest distance).

$$D_s(A,B) = \min\{d(a \in A, b \in B)\}$$

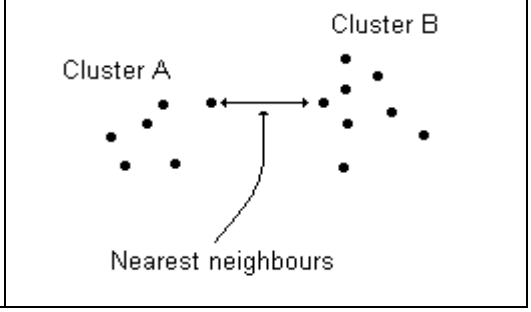


Figure 29 : Single clustering

On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest single linkage distance.

- Complete clustering (furthest neighbor) method

In this method, the distance between two clusters A and B is based on the data vectors in each cluster that are furthest apart or furthest neighbors (longest distance).

$$D_c(A,B) = \max\{d(a \in A, b \in B)\}$$

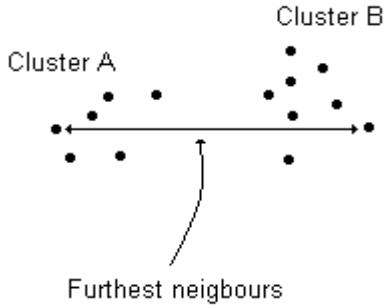


Figure 30 : Complete clustering

On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest complete linkage distance.

- Average clustering method

In this method, also known as the unweighted pair-group using average approach conventionally abbreviated (UPGMA), the distances between all possible data vectors embedded in the two clusters A and B are calculated and summed, and the distance between cluster A and cluster B is the average of that sum.

$$D_{avg}(A,B) = \frac{\sum_{i=1..m, j=1..n} d(a_i \in A, b_j \in B)}{m \times n}$$

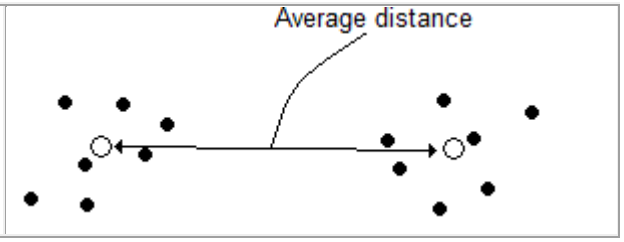


Figure 30 : (Group) Average clustering method

Where $D_{avg}(A,B)$ is the average link distance between A and B, d is the distance between a single pair of data vectors, m is the cardinality of cluster A, and n is the cardinality of cluster B. On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest average linkage distance.

- Weighted Average clustering method

This method has also been referred to as the weighted pair-group using average approach conventionally abbreviated (WPGMA). In this method, when two clusters A and B are agglomerated, the distance D between some other cluster, say, C and the newly formed cluster AB is the simple average of D_{CA} and D_{CB}, thus:

$$(D)\ C, AB = \tfrac{1}{2}\{D_{CA} + D_{CB}\}$$

On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest weighted average linkage distance.

- Ward's method or an increase in sum of squares clustering method

This method involves the concept of sum-of-squares error, abbreviated SSE. Given a set D of *n* values, the SSE of D is the sum of the squared differences between each value in D and the mean of all values in D:

$$SSE_D = \sum_{i=1..n} \left| d_i \in D - \frac{\sum_{j=1..n} d_j \in D}{n} \right|^2$$

Ward's method calculates the distance between clusters A and B as

$$D_{Ward} = SSE(A,B) - \left(SSE(A) + SSE(B)\right)$$

On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest increase in the sum of squares.

- Sum of squares clustering method

The distance between two clusters A and B is calculated as the sum of the squared distances between

the data vectors of clusters A and B and the centroid of the agglomerated cluster. The sum of squares method is only calculated for squared distances. For a given set of n data vectors, this method seeks to minimize the sum of the squared distances between the data vectors and the centers (or means) of the clusters to which they belong. In this respect, it is very similar to Increase in Sum of Squares (Ward's method) above.

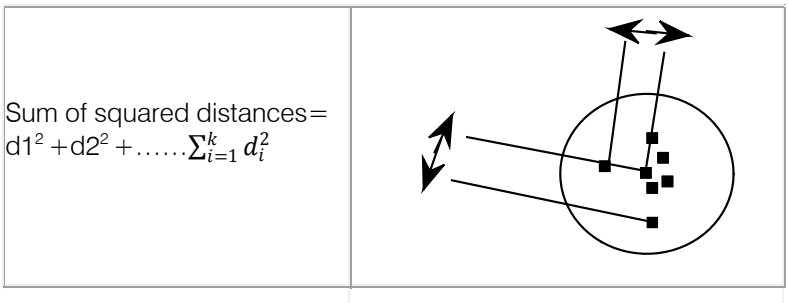


Figure 31 : Sum of Squares clustering

- Centroid clustering method

This method is also known as the unweighted pair-group method using the centroid approach (UPGMC). The Centroid method is only calculated in terms of squared distances. The squared distance between two clusters A and B is calculated as the squared distance between the cluster means, or centroids. The size or weight of a cluster is not relevant, although its spatial distribution is used in the calculation of the centroid. This method should, strictly speaking, only be used with a matrix of squared distances.

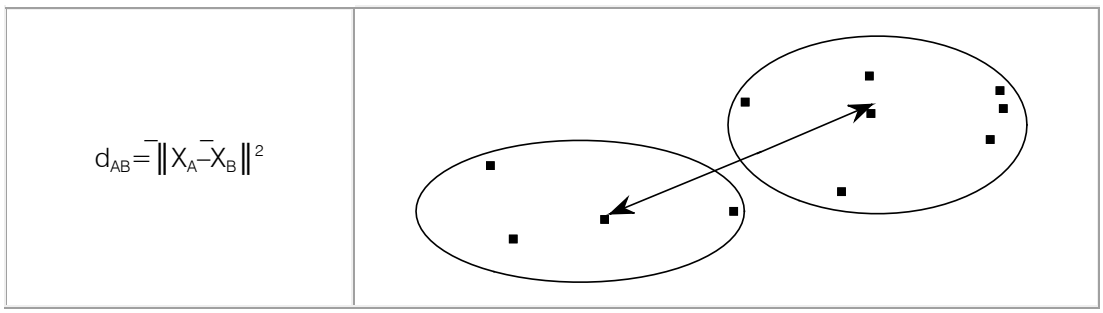


Figure 32 : Centroid clustering

Where X_A and X_B are the mean vectors for the data vectors in A and the data vectors in B respectively. On this basis, at each step of the clustering process, we combine the two data vectors that have the smallest centroid distance.

- Median clustering method

Also known as the weighted pair-group method using centroid approach (WPGMC). The Median method is only calculated in terms of squared distance. In this method, the distance between two clusters A and B is represented by the squared Euclidean distance between the median (mid-point) for the data vectors in cluster A and the median for the data vectors in cluster B. This gives equal weight to clusters of different sizes, unlike the centroid, which is weighted by the number of data vectors in each cluster. However, the two data vectors with the smallest distance between medians are agglomerated at each step.

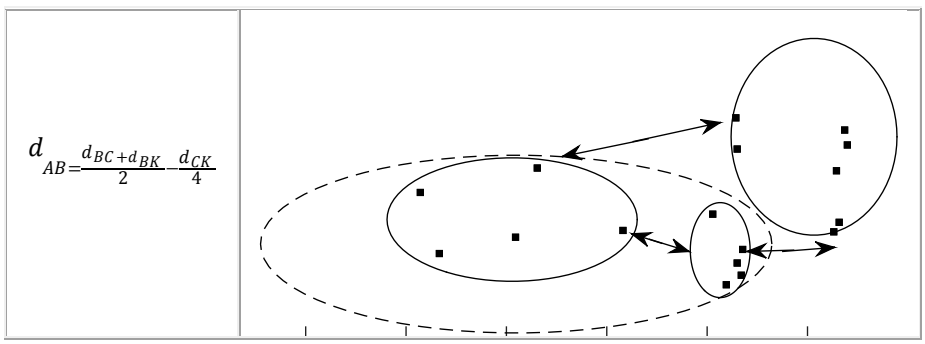


Figure 33 : Median clustering

- Flexible beta clustering method

This method calculates the distance between two data vectors on the basis of β which is a supplied by the user. By allowing β to vary, clustering results with various characteristics can be obtained. However, a value of β = -0.25 gives results similar to Ward's method. A detailed account on the mathematical properties of this method can be found in, e.g., [18] and [19].

- Mean proximity clustering method

This method maximizes the average of the within-cluster distances or minimizes the average of the between-cluster distances, for all cluster comparisons.

- Density search clustering using nearest- neighbor clustering approach

This method falls into a class of clustering methods particularly designed to seek dense patches, regions or areas in the data vectors in a metric space depending on the type of the density estimation to be used. The density nearest neighbor method uses either K^{th} nearest neighbor density estimates or smoothed K^{th} nearest neighbor estimates. The density estimation of the former is based on a fixed number of values and the density estimation for the latter on a large number of values K, where k is the contiguous or the nearest neighbors to the desired point. The distance between two clusters A and B is based on the value specified for K; the estimated value of k controls the amount by which the data are smoothed or unsmoothed to give the density estimate on which the clustering procedure is based: when the value of k is non-increased or small, the density estimation becomes unsmooth or jagged, when the value of k is increased or large, the destiny estimate becomes smoother or less bumpy. To be more precise, the problem is that all K neighbors must be close to the desired point. This may or may not be possible. Theoretically speaking, this is possible when infinite number of data vectors is available, in such a situation the larger the k value the better is calcification (error rate gets closer to the lowest possible error rate for a given classification). Because this is not always possible in practice due to data vectors are finite, K value should be large so that error rate is minimized; too small values of K may lead to noisy decision boundaries and too large may lead to over-smoothed boundaries. That is, K value should be small enough so that only nearby data vectors are included. However, whatever density estimation it may take, this method consists of two main basic steps: initially, a new distance, based on density estimates and adjacencies in the data vectors, is calculated. This step is obtained by: calculating the K^{th} nearest neighbor for the data vectors: given two clusters A and B, the data vectors X_A and X_B are said to be adjacent (the definition of adjacency depends on the method of density estimation), if $D^*(X_A, X_B) \leq D_K(X_A)$ or $D_K(X_B)$. Where D^* is the distance and $D_K(X_A)$ is the kth nearest neighbor distance to data vector (X_B). The distance $D(X_A, X_B)$ between the data vectors X_A and X_B can be obtained as:

$$D(X_A, X_B) = 0, \text{ if } X_A = X_B;$$
$$= \tfrac{1}{2}[D_K(X_A) + D_k(X_B)], \text{ if } D^*(X_A, X_B) \leq D_K(X_A)$$
$$\text{or } D^*(X_A, X_B) \leq D_K(X_B)$$
$$= \infty \text{ otherwise.}$$

Finally, a single linkage clustering method is then applied to the resulted distance D* to obtain high-density clusters [2], [3], [10], [16], [20], and [21].

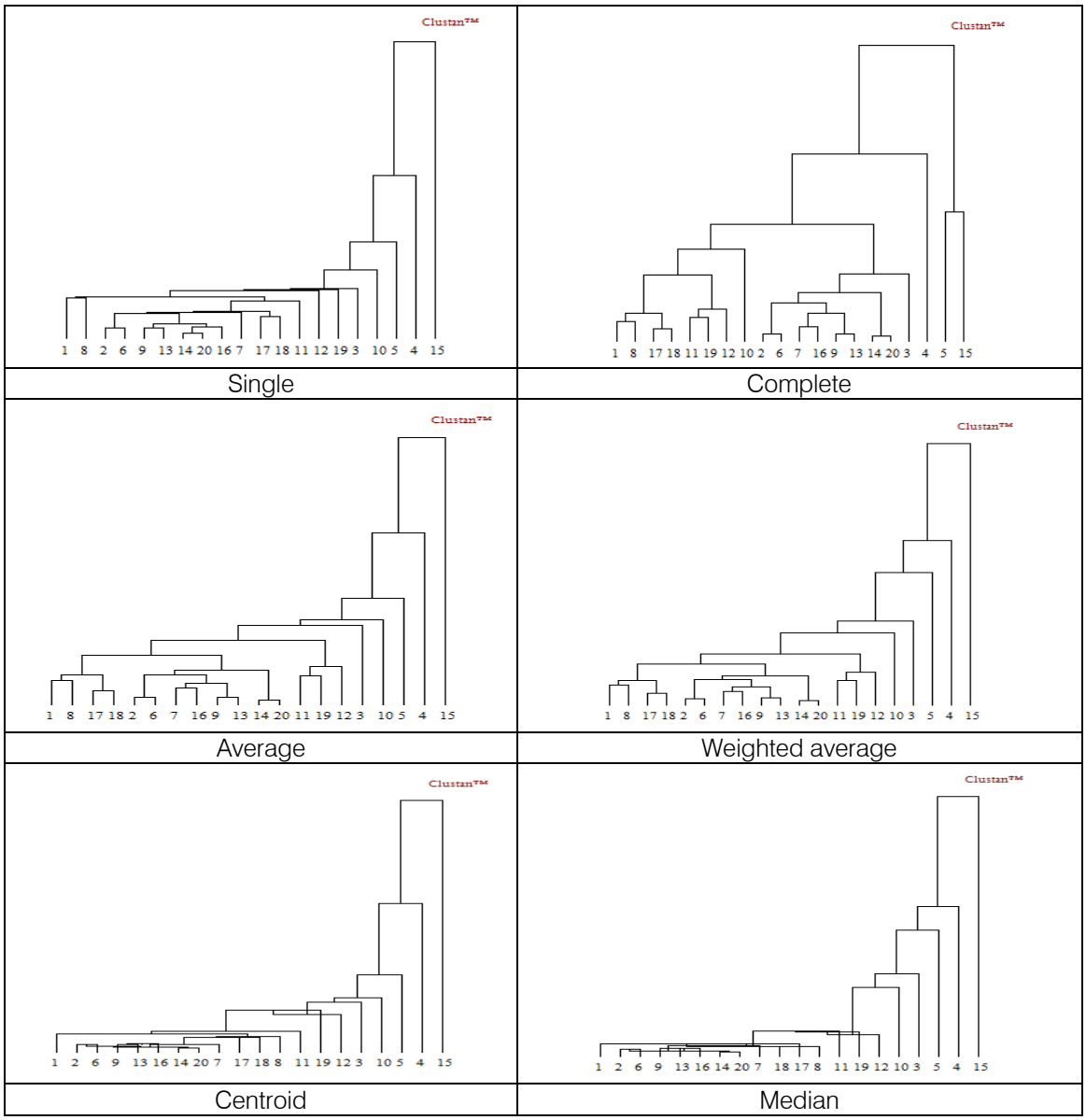
A detailed account on the mathematical properties of these methods can be found in, e.g., [5] and [16].

Since the calculation both of the values in the original distance matrix and of the distances between composite clusters are based on linear measurement, agglomerative hierarchical clustering is a collection of linear cluster analysis methods.

Extensive empirical clustering results, however, have shown that, relative to a given data matrix, each agglomerative clustering method has a 'signature' in the sense that the hierarchical tree it produces tend to have specific characteristics [2] and [5].The literature search on the application of hierarchical clustering methods reports, for example, that Single link famously tends to generate 'chained' structures, that is, trees with a strong tendency to either left or right branching but not both. It also reports that this method has satisfactory mathematical properties, which appears to give satisfactory results at identifying longated clusters that have curvy shapes instead of spherical or elliptical shapes, and it is somewhat robust to outliers in the set of data. Complete link tends to generate trees with extensive recursive embedding of left and right branching sub trees; also tends to generate very small compact clusters, which means that they have small diameter (max. distance between data vectors). In other words, group structure, all data vectors in the same cluster, will not be taken into account. On the other hand, this method is somewhat sensitive to outliers, and is suitable for compact but not well-separated clusters. Average linkage is intermediate between single and complete link; it is intermediate between single and complete linkage; it tends to generate small clusters of outliers and to find spherical clusters, i.e. ball-shaped clusters. Being relatively robust, this method can even deal with rather potato-shaped clusters. It is, however, more prone to chaining than Ward's method. Ward's method is like complete link, but in addition tends to find spherical clusters of roughly equal size. As such, some methods are more appropriate than others for data with a given density structure. If, for example, the data manifold has an elongated structure, single link would be best and Ward worst. Alternatively, a manifold with

well-defined spherical areas of vector density would reverse that. Ward's method tends to find spherical clusters of roughly equal size. It is sensitive to outliers. On the other hand, many researchers report satisfactory results with this method (i.e. provides interpretable results). Centroids linkage tends not to chain as much as single linkage. It is nevertheless subject to reversals. Median linkage tends to chain for large set of data and is also subject to reversals. However, they are both fairly robust to outliers. Flexible beta linkage tends to generate 100 % chained clusters if β approaches a value of +1. On the other hand, if β approaches zero and then becomes negative, this method tends to cluster data vectors more intensely. A value of β −0.25 gives results similar to Ward's method. Density nearest-neighbor linkage leads to a very simple approximation of the (most desired) smallest possible error rate for a given classification and data representation. However, it tends to generate different clusters with greater or lesser tendency to chain depending on different values of k. This method tends to overcome the chaining effects if $k = 2\log_2 n$ or several values around this value. On the other hands, this method is prone to produce noisy decisions boundaries. As such some methods are more appropriate than others for data with a given density structure; some methods work better for certain data sets, and other methods work better for other data sets. However, if, for example, the data manifold has an elongated structure, single or nearest neighbor linkage would be best and Ward worst.

As might be expected, different agglomerative clustering methods can and often do give different results for the same dataset. Different clustering structures are obtained when we cluster analysed a data matrix consisting of 20 data vectors applying the 11 methods introduced above.



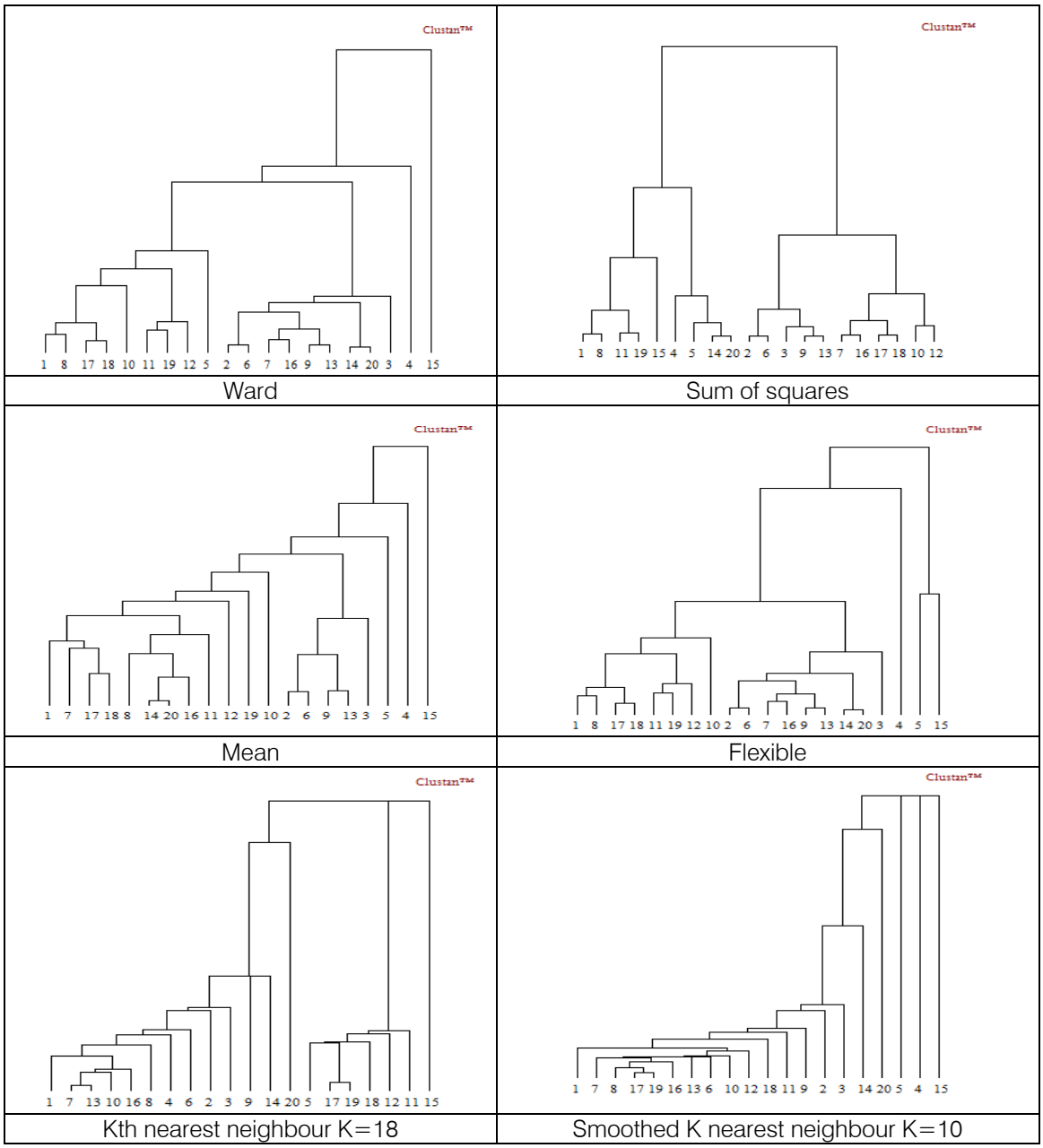
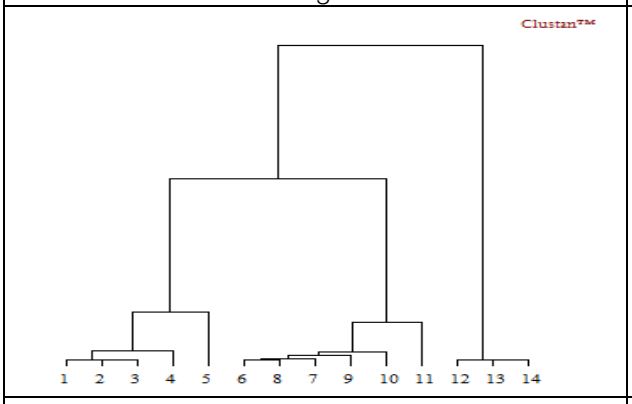
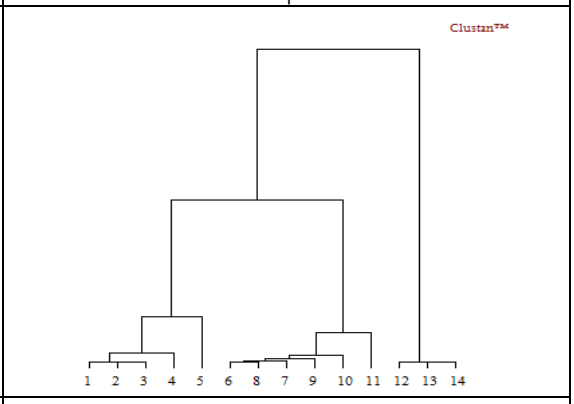
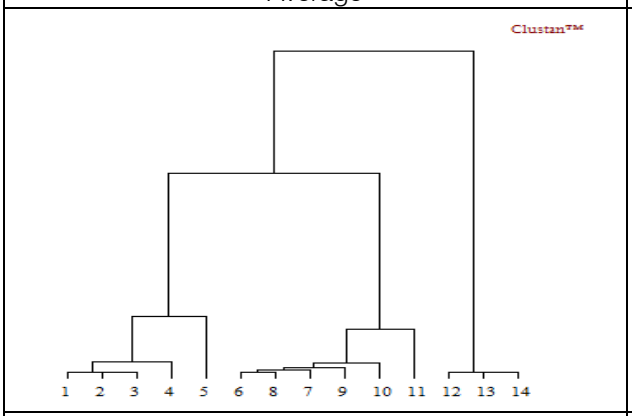
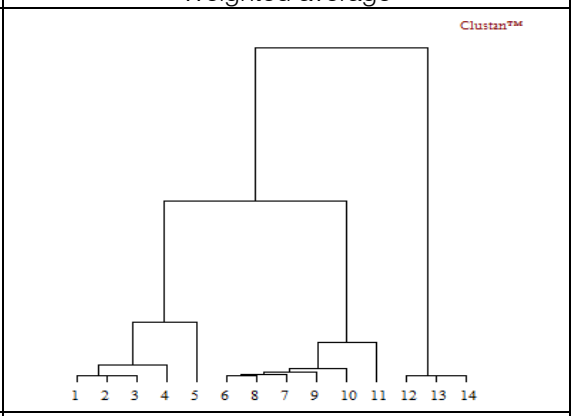
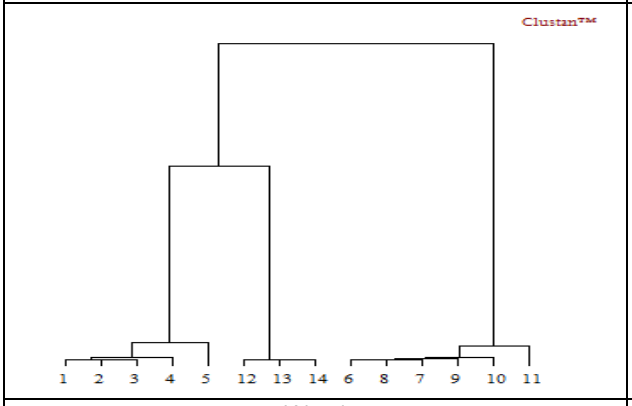
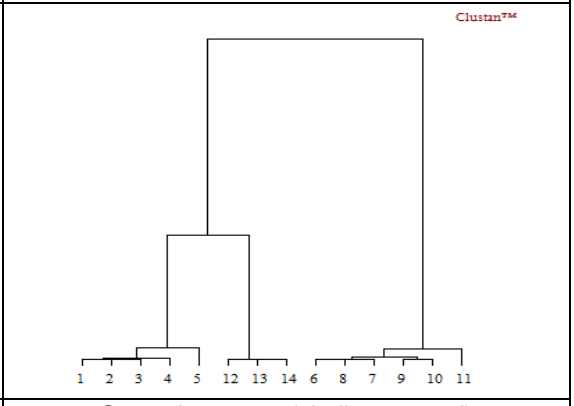
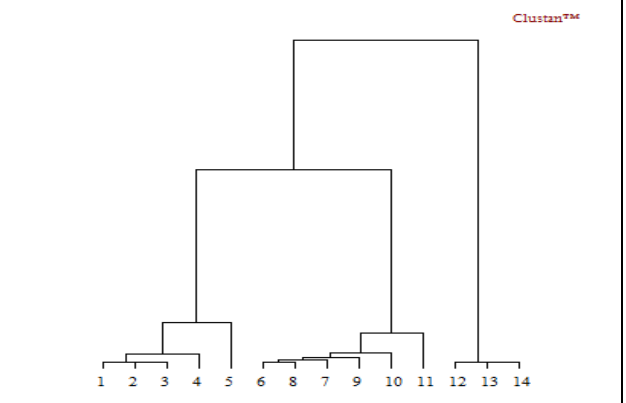
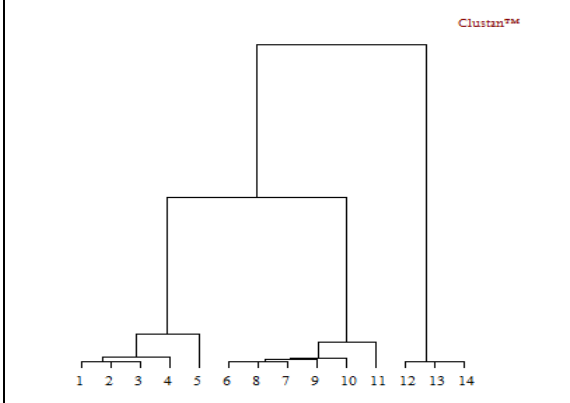


Figure 34 : The application of different hierarchical clustering methods on the same data set using squared Euclidean distance

Which hierarchical analysis is the best? None of these clustering analyses is uniformly the best. In this practice it is advisable to try several methods and then compare the clustering results to form an overall judgment about the final structures of clusters.

Occasionally, however, observed clustering results are very different from those expected. Here is a little example to illustrate this. The following dendrograms generated from the eleven hierarchical clustering methods applied on a small data matrix (i.e. having small measurements).

Single	Complete
	
Average	Weighted average
	
Centroid	Median
	
Ward	Sum of squares (similar to ward)
	
Mean	Flexible

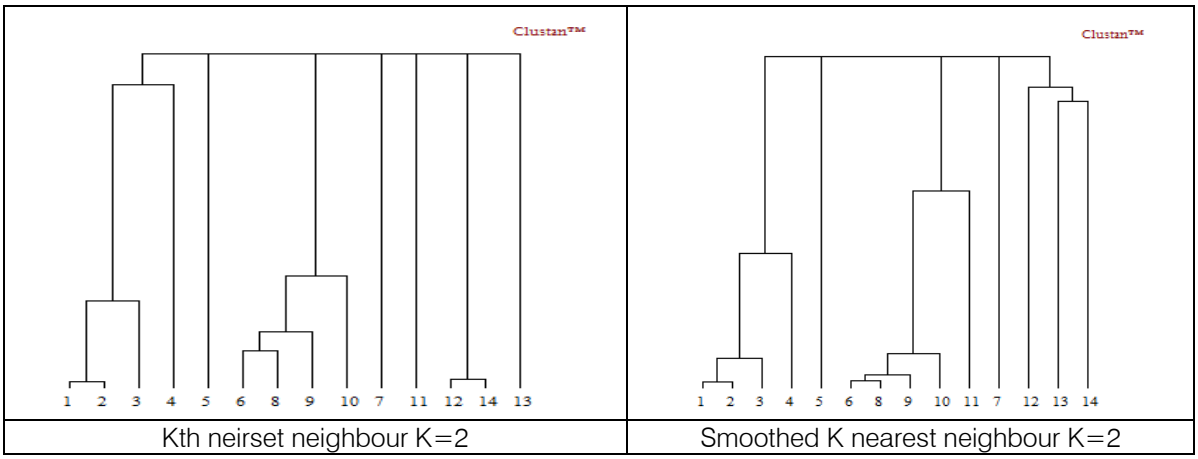


Figure 35 : the application of eleven hierarchical clustering methods on the same (small) data set using squared Euclidean distance

The clustering analyses in this figure show that the application of various agglomerative methods on the same dataset may not always produce quite different results because the clustering results may have generated as a result of highly precise data or a small data matrix size.

III. Conclusion

When using agglomerative hierarchical analysis to form clusters, we need to keep the following in mind:

- Agglomerative hierarchical cluster analysis is a multivariate method for finding structures or groups called clusters in data in relation to a research of interest. The clusters are based on the values of several variable measurements that describe data vectors. The accuracy of agglomerative hierarchical cluster analysis is unquestionable: Data vectors (objects, cases, observations) in a specific cluster share many characteristics, but are very dissimilar to data vectors not belonging to that cluster.
- Prior to analyzing data and applying a clustering method, we need to choose the appropriate proximity coefficient (i.e. measure of distance/ similarity) depending on type of data: interval, counts, binary. Distance is a measure of how far apart two data vectors are, while similarity measures how similar two data vectors are. For data vectors that are similar, distance measures are small and similarity measures are large.
- Proximity coefficients are stored in a proximity matrix. The proximity matrix identifies which cluster each data vector belongs to for any specified number of clusters.
- Agglomerative hierarchical cluster analysis starts with as many clusters as data vectors. Data vectors are successively agglomerated into clusters until only one data vector remains. The result of this can be shown in a dendrogram. The dendrogram is the tree-like diagram that can show the data vectors, which have been clustered at each agglomeration sequence.
- Often, but not always, different agglomerative clustering methods for analysing data can yield different results. In particular for small data sets, different methods might produce similar results.

IV. Acknowledgments

The author wishes to thank all those who dedicated their time answering my queries and providing me with valuable comments during the preparation of this study.

Conflicts of Interest

The author declares no conflict of interest.

References Références Referencias

1. Hermann Moisl. *Cluster analysis for corpus linguistics*. De Gruyter Mouton: Berlin, 2015.
2. Everitt, B. S.; Landau, S.; and Leese, M. *Cluster Analysis*. 4th ed.; Arnold: London, 2001.
3. Everitt, B. S. Unresolved Problems in Cluster Analysis. *Biometrics*1979, 35, 1, 169-181.
4. Charles Romesburg. *Cluster Analysis for Researchers*. Wadsworth Inc: USA, 1984.
5. Anil Jain and Richard Dubes, R. *Algorithms for clustering data*. Prentice-Hall Englewood Cliffs: NJ, USA. 1988.
6. Michael Steinbach. Available online: http://www-users.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf (accessed on 20 September 2015).
7. Patricia Pulliam Philips and Cathy A. Stawarski. *Data Collection Planning and Collecting.* All Types of Data, Pfeiffer: USA, 2008.
8. Craig J. Cleaveland. *An Introduction to Data Types*. Addison-Wesley Longman, Incorporated: Michigan, 1986.

9. Pyle D. *Data preparation for data mining*. CA: Morgan Kaufmann Publishers: San Francisco, 1999.
10. Michael Anderberg. *Cluster analysis for applications*. Academic Press, Inc: London, 1973.
11. Coxeter H. S. M. *Non-Euclidean Geometry*. The Mathematical Association of America: USA, 1998.
12. Berry Bonola. *Non-Euclidean Geometry*. Lighting Source: USA, 2007.
13. Deza, M. and Deza, E. *Encyclopedia of Distances*. Springer: Berlin, 2009.
14. Rui Xu and Donald C. Wunsch. *Clustering*. Wiley-IEEE Press: Hoboken, New Jersey, 2009.
15. Baker Kirk. Available online: https://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf (accessed on 10 September 2015).
16. Rencher, A. C. and Christensen, W. F. *Methods of multivariate analysis*. 3rd ed., John Wiley and Sons In: Hoboken, New Jersey, 2012.
17. Amit Singhal. Available online: http://singhal.info/-ieee2001.pdf (accessed on 2 July 2015).
18. Lance, G. N. and Williams, W. T. A general theory of classificatory sorting strategies I. hierarchical systems. *Computer journal*1967, 9, 373-80.
19. Milligan, G. W. A study of the beta-flexible clustering method. *Multivariate Behavioral Research* 1989, 24, 163-176.
20. Berry, M. W. and Browne, M. *Lecture notes in data mining*. Fu Island offset printing: Singapore, 2006.
21. Rosie Cornish. Mathematics Learning Centre. Available online: http://www.statstutor.ac.uk/resources/uploaded/clusteranalysis.pdf (accessed on 22 July 2015).