Benha University,
Faculty of Commerce
Department of Statistic,
Mathematic and Insurance

Clustering Multidimensional Scaling and It's Application

By

Maha Abdalla Ibrahim Moussa

Supervised By:

PROF.
ABD ELFATTAH MOHAMED KANDIL
Prof. of Statistic, Former Dean of Faculty of Commerce, Benha University

DR,
ZOHDY MOHAMMED NOFAL
Associate Prof. of Statistic, Dept. of Statistic, Mathematic and Insurance Faculty of Commerce Benha University

THESIS

Submitted to the Department of Statistic, Mathematic and Insurance, Faculty of Commerce, Benha University, in partial fulfillment of the requirements for the degree of Master of Statistics, Benha University

2015

Acknowledgement

First of all, I would like to express my deepest thanks to "ALLAH" for helping me to carry out and complete this work.

I am greatly indebted to my supervisor, ***Prof. Dr. Abd El Fattah Mohamed Kandil,*** Prof. of statistics and former Dean of Faculty of Commerce at Benha University, I'd like to thank him for his constructive supervision, research problem, providing me with all the facilities needed for the study to be feasible guidance, valuable advice and for his help in putting thesis in its final form.

Gratefulness and thanks are not enough to express my deep gratitude, and sincere appreciation to ***Dr. Zohdy Mohamed Nofal,*** Associate professor of statistic., Benha Univ., for his valuable comments, offering the support necessary for this investigation, supervision, guidance, encouragement and facilities that are offered and without it this work couldn't completed.

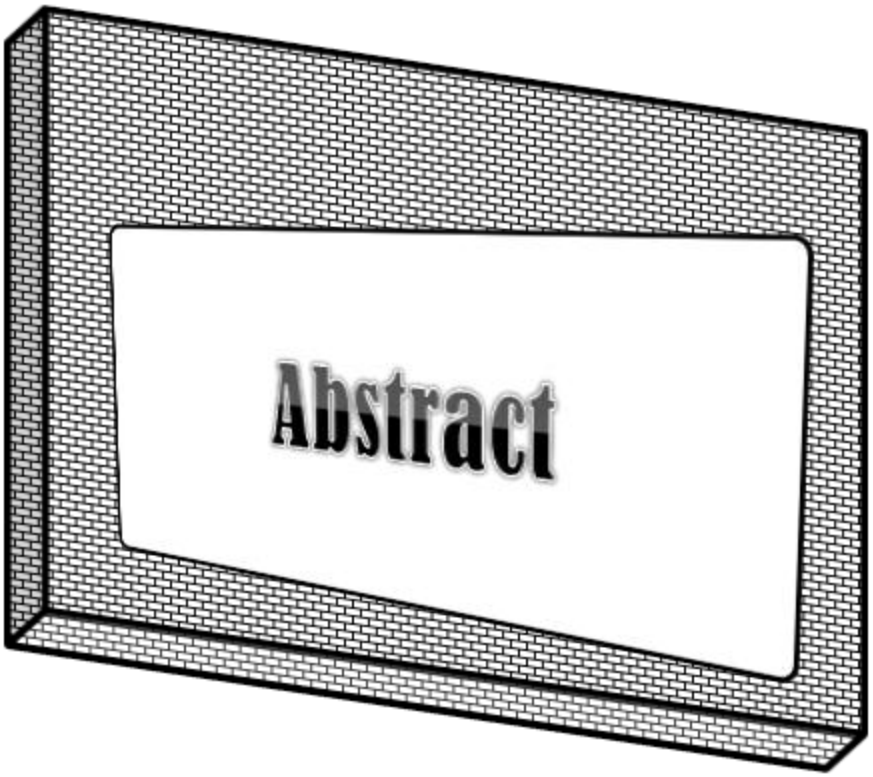
I'd like to thank my professors at the faculty of commerce, Benha University, who supported me throughout the course work of this thesis.

Finally, I am indebted to my parents and family for their great help, continuous encouragement, praying for me and patience during this work and special thanks for my husband and sons.

I fully thank everyone helped me in one way or another to make such a work successful.

I Ask Allah to Bless Them All

CONTENTS


Abstract

Abstract

Multidimensional scaling and cluster analysis are two numerical techniques that assist the researcher in ascertaining the structure of data in different spaces. Multidimensional scaling allows the researcher to convert large amounts of similarity or proximity data into a geometric picture while Cluster analysis represents an area of statistics that is concerned with sorting the observed data into some groups (clusters) based on the similarity.

It is highly recommended to perform cluster analysis in conjunction with MDS for many reasons:

(i) Cluster analysis may provide the researcher with ways of understanding similarity criteria when interpretations of geometric dimensions are not readily apparent.

(ii) In some clustering problems as in case of lacking metric data attributes. For example, we only have the dissimilarities between data objects. The dissimilarity between two data objects can be metric or nonmetric. To obtain data in the metric space from these dissimilarities, a possible solution is using multidimensional scaling (MDS).

There are several models of MDS and CA available to the researcher; the choice mainly depends upon the type of data believed to be under the study.

In this thesis, several models of MDS and CA were introduced. In addition, we provided a solved mathematical example for each models.

Since the MDS and cluster analysis are mainly based on the proximity data, we introduced the different patterns of proximity measures (similarity and dissimilarity) in addition to solved mathematical example for each measure.

In this study we performed an application of cluster analysis and multidimensional scaling on one data set from different car exhibitions and agencies in Benha city. The data was collected based on the responses we received in all the questionnaires which were distributed among different car exhibitions in Benha city. The sample size was 20 customers. The Twenty customers were asked to rate the 10 cars by showing the cards bearing the name of a pair of cars. All possible pair of cars were shown, and the customers were asked to rate their preferences of one car over the other on a scale of 100 points. If the customer perceived that the two cars were completely dissimilar, a score of 0 was given, and if the two cars were exactly similar a score of 100 was given. The Statistical Package for Social Sciences (SPSS) was used in order to apply the multi-dimensional scaling to convert cars market similarity data into a geometric picture. SPSS was then used to group different cars brands in this geometric map into some clusters. After finalizing the analysis and getting the result, we performed interpretations of the results and provided insights for some companies to know how their brand of products is rated among other similar competing brands of other companies.

To achieve the purpose of this study, the thesis consists of five chapters as follow:

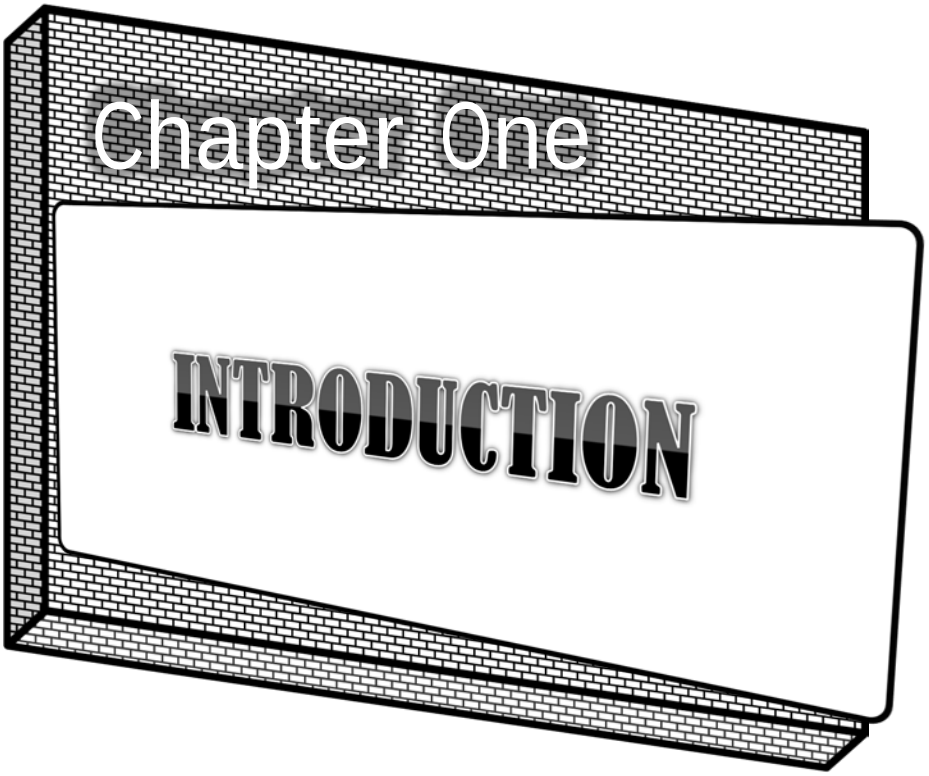
Chapter I: An introduction includes a background on multidimensional scaling and cluster analysis in addition to the aims of the study.

Chapter II: Measures of proximity which discuss the different patterns seen in proximity measures (similarity and dissimilarity).

Chapter III: Multidimensional scaling in terms of concepts and methods.

Chapter IV: Cluster analysis in terms of concepts and methods.

Chapter V: An application of cluster analysis and multidimensional scaling.


Chapter One
INTRODUCTION

I. INTRODUCTION

1.1 Introduction

The amount of data collected from various sources is increasing. With the invention of new technologies, preserving this enormous volume of data for future reference and analysis has become more manageable. In contrast, the task of discovering underlying patterns and hidden information from data has become more challenging and complex.

According to Witten et al. (2005) *"As the volume of data increases, inexorably, the proportion of it that people understand decreases, alarmingly1'*. As such, we need automated and practical tools and techniques to take full advantage of the information lying hidden in the data. This is where *Data Mining* techniques come to aid. *Data Mining* is defined as the process of automatic discovery of hidden, interesting, and previously unknown patterns in data stored electronically [Witten and Frank (2005)]. Some of the benefits of mining data are to extract previously unknown information and use it to predict future trends, make decisions, categorize or group data to discover common characteristics, amongst others. Among various data mining techniques, cluster analysis (CA) and multidimensional scaling (MDS) are interesting and fast growing topics.

Multidimensional scaling and cluster analysis are two numerical techniques that assist the researcher in ascertaining the structure of data in different spaces.

Multidimensional scaling allows the researcher to convert large amounts of similarity or proximity data into a geometric picture. Upon obtaining a geometric representation, it is the researcher's task to develop interpretations for the different dimensions in that picture.

Multidimensional scaling analyses typically report results in two or three dimensions for ease of viewing and interpretation by the researcher, but it is possible to search for better goodness-of-fit in higher dimensional spaces indeed.

Cluster analysis is a related visualization technique that returns a tree structure rather than a geometric configuration. It is particularly useful when used in conjunction with MDS since it may provide the researcher with ways of understanding similarity criteria when interpretations of geometric dimensions are not readily apparent. Cluster analysis is also appropriate for situations where the multiple frames of reference or other violations of modeling assumptions, geometric configurations provide poor fits to rating data.(**Tversky and Hutchinson (1986)).**

Generally, there are two types of attributes involved in the data to be clustered: metric and nonmetric. If all the data attributes are metric, a data object can be represented by a vector in the metric space. A metric space is a set S with a global distance function (the metric *d*) that, for every two points x, y in S, gives the distance between them as a nonnegative real number *d(y,x).*A metric space must also satisfy:

1. $d(x, y) = 0$, if $x = y$.
2. $d(x,y) = d(y,x)$.
3. The triangle inequality: $d(x,y) < d(x,z) + d(z,y)$.

In many clustering problems, we do not have metric data attributes. For example, we only have the dissimilarities between data objects. The dissimilarity between two data objects can be metric or nonmetric. To obtain data in the metric space from these dissimilarities, a possible solution is multidimensional scaling (MDS). Besides, MDS can be used to

transfer data from a higher dimensional metric space, say m-dimension, to a lower dimensional metric space, say p-dimension, where $p < m$.

There are several models of MDS and CA available to the researcher; the choice mainly depends upon the type of data believed to be under study.

1.2 Aim of the proposed study:

Clustering analysis and MDS will be applied to a data set of car brands and their ratings among customers in car market. In this study, we are trying to describe the relationships among the 10 car brands. The results produced by application of these methods together can be then used to investigate whether different car brands mentioned in the market are strongly related or not. MDS methods will be used to create separate displays for each car based on two factors (2 dimensions) in a geometric picture. Afterwards, Cluster analysis will be used to show the clustering structures of different cars within the market thus helping the car companies to know how their car brand is rated among other similar competing car brands and who their rival in the same cluster are.

1.3 Review of previous studies:

I-Cluster Analysis:

Cluster analysis is used in many disciplines, including biology, geology, anthropology, and marketing **(Tryon, 1939)**. Before cluster analysis can be performed, a set of objects must be arranged in a data matrix. In most cases, the columns of the matrix represent the individual objects, while the rows represent a set of determined attributes that each

object may or may not possess. For example, an archaeologist may be interested in determining the evolutionary link of an unspecified set of bones. The archaeologist can identify several physical, chemical, and other attributes of these bones and arrange them as rows on a matrix. Then, the bones and other bones that have already been classified are laid out as columns. Cluster analysis uses a variety of mathematical methods to determine which classified bones are the most similar to the unknown bones, based on the determined attributes **(Kaufman & Rousseauw, 1990).**

Romesburg (1984) outlined three research goals that cluster analysis can answer. The first goal is to create a question to be tested later. Creating a question is relatively simple, as the researcher can simply run a cluster analysis on a data matrix and observe what clusters form together. Though it would be irresponsible to draw any conclusions without a hypothesis, it is appropriate to further investigate any interesting patterns that emerge in subsequent studies. The second goal is to create a hypothesis. The researcher already has a question framed when running the analysis, but no testable hypothesis. Any patterns that emerge may answer the question and open up the possibility of a hypothesis. Finally, cluster analysis can be used to test a hypothesis. Typically, previous studies that may or may not have already used cluster analysis have presented evidence of a clear, testable hypothesis. The hypothesis must be made a priori and any conclusions must be directly related to the hypothesis. Most of the literature on psychometric measures already has a firmly developed hypothesis.

Once a researcher has put together a data matrix, the researcher determines how to analyze the data by choosing a resemblance coefficient. There are several resemblance coefficients to choose from, but each

coefficient is either a similarity or dissimilarity coefficient. This dichotomy simply expresses the direction of the data; when using a similarity coefficient; larger values indicate higher similarity between two objects while the opposite is true with a dissimilarity coefficient. In psychology literature indicates that the Euclidean distance coefficient is the most common distance measure in published studies **(Clatworthy et al., 2005)** which finds the least distance between two objects via Euclidean geometry. This coefficient can easily be visualized when only two attributes are compared across the objects. These two attributes are treated as coordinates on a two-dimensional plane, and the point on the plane represents an object. The Euclidean distance coefficient calculates the linear distance between objects by using the Pythagorean Theorem. Therefore, the farther two points are, the more dissimilar the represented objects are from each other.

In most matrices, objects are compared across more than two attributes. A three-attribute cluster analysis can be envisioned as a three-dimensional space, but higher attribute analyses cannot be pictured as easily. Nevertheless, the principle remains the same: the Euclidean distance coefficient calculates the overall distance that two objects are from each other in a hypothetical space. These distances are placed on a new matrix called the resemblance matrix, with which researchers can determine the similarity between individual objects. However, how objects actually combine to form clusters is determined by a second technique called the clustering method **(Thaler 2010).**

Like with distance coefficients, the researcher determines the optimal clustering method and there are many methods that he can select **(Kaufman and Rousseauw, 1990).** Clustering methods can be hierarchical or partitional in nature. Hierarchical methods are the preferred

form for most researchers, as they build dendograms, or trees, which are visual representations of the clusters. The majority of the previous studies used Ward's minimum variance clustering method **(Ward, 1963),** which is also the second most used clustering method across all scientific fields **(Romesburg, 1984).** Like all hierarchical methods, Ward's method is agglomerative, building clusters from individual objects and combining clusters based on their similarity to each other until the final cluster, which encompasses all the data, is formed. This final cluster can be visualized as the "trunk" of the tree, which in turn breaks into smaller and smaller branches, while the tips of the tree represent the original objects. Ward's method calculates similarity by using a sum-of-squares calculation to see which two items exhibit the least variance when combined into a hypothetical "average." All cluster combinations are compared at each level of the tree, and a new cluster is formed each time the smallest variance is found. This continues until all objects are formed into one unifying cluster.

Another hierarchical clustering method worth noting is the two-step clustering method, which has the advantage of automatically selecting the number of clusters and handling categorical as well as continuous variables **(Bacher, et al, 2004).** The two-step method clusters individual cases into small sub-clusters, and then clusters these sub-clusters into the cluster solution. In large datasets with only continuous variables, such as the dataset in this study, the Euclidean distance coefficient is used. A survey on agglomerative hierarchical clustering algorithms was performed by **Murtagh and Contreras** (2011) who discussed their efficient implementations. They look at hierarchical self-organizing maps, and mixture models.They described a recently

developed very efficient (linear time) hierarchical clustering algorithm, which can also be viewed as a hierarchical grid-based algorithm. They also touched on a number of application domains.

Once the dendogram is fully formed, researchers must determine where to "cut" the tree, or where the optimal cluster solution is found. The optimal cut is subjective, but typically a smaller cluster solution is preferred over a larger one. **Romesburg (1984)** recommends that the tree should be cut where the clusters are maximally related to other variables of interest. Therefore, cutting the tree in different ways may produce different results, and the one that fits the proposed hypothesis the best should be selected.

There may be some unforeseen complications that emerge from the data. Chaining is a term used to describe a cluster that repeatedly merges with individual objects; much like a black hole absorbs random pieces of debris **(Anderson, 1973).** Ideally we would want objects to clump into several smaller clusters and only merge together into the single cluster at the very end of the analysis. With chaining, it is more difficult to determine the similarity of objects as each object is added one at a time to a single, growing cluster. Another complication can emerge when the dendogram does not accurately represent the data matrix **(Romesburg, 1984).** This can occur because clustering methods mathematically calculate the similarity of objects using formulas that do not exactly match the actual similarity in Euclidean space (or, if another coefficient is used, whatever is determined to represent similarity among objects). Researchers typically avoid this problem by calculating the cophenetic correlation coefficient, a Pearson's correlation between the actual data matrix and the proposed matrix formed

from the dendogram. Correlations that are greater than 0.80 indicate that the distortion between the matrix and the dendrogram is not severe.

Sattath and Tversky (1977) criticized existing hierarchical clustering algorithms on the grounds that empirical rating data, which tend to be messy, often violated a basic assumption of such algorithms. This assumption is called the ultrametric inequality; Sattath and Tversky's description of its concise: given two disjoint clusters, all intra-cluster distances are smaller than all inter-cluster distances, and all the inter-cluster distances are equal. An additive tree algorithm is a method for generating a tree structure given a similarity or distance matrix that does not require the data to be constrained by the ultrametric inequality. As in other tree structures, leaf nodes of the tree correspond to stimuli and the distance (dissimilarity) between them is the length of the path joining them. Unlike other hierarchical schemes, however, additive trees perm it intra-cluster distances to exceed inter cluster distances. As a result, additive trees typically give better fits to rating data than other, simpler, hierarchical clustering models.

Additive clustering differs from both hierarchical and additive tree structures in that objects can belong to multiple groups simultaneously. For example, if subjects were to categorize the numbers 1 to 10, they might adopt a number of overlapping schemes: evens vs. odds, smaller (≤ 5) vs. larger, multiples of 3, powers of 2, primes vs. non-primes, and so forth. In any hierarchical or additive tree scheme, effectively only one of these features could be used to ascribe any object's location in the tree structure, creating a "winner-take-all" scenario. This sort of procedure goes directly against the view of similarity that **Tversky (1977)** argued for, where a total

judgment of similarity between any two objects is a comparison/summation across a range of different features.

The K-means as one of partitional clustering method was applied by **Tarpey (2007)** on functional data. He compared the differences in the clustering outcomes of the K-means method based on how the observed data were smoothed. In his study he applied the K-means method to the raw data, and then to three transformations of the raw data into curves including: B-spline basis, Fourier basis, and a power basis (using an L2 metric). For each of these transformations, the estimated regression coefficients were clustered by the K-means algorithm. The functional data used were estimated Hamilton Depression responses from a clinical trial.

Rehman and Mehdi (2013) set a comparison between density-based algorithms by implementing detailed study of density based algorithms (Density based spatial clustering of applications with noise (DBSCAN), Recursive density based clustering (RDBC).

II-Multidimensional scaling:

Much of this brief history was found in **Wish and Carroll (1982). Carroll and Arabie (1980)** provide a valuable summary within a taxonomy framework and supply an extensive bibliography. Multidimensional scaling has its origin with a paper by **Young and Householder (1938)**. This paper introduced a theorem which addressed the minimum number of dimensions needed to fit a set of distances to *N* points, and a method for building a space capturing the distances. Although

Richardson (1938) followed quickly with an application of this technique, little progress was made until the 1950's when the facility of computers made the large amount of required calculation feasible. In this period, **Torgerson (1952, 1958)** developed the techniques that embody classical multidimensional scaling. He showed the methods that have been used during this early period. The problem for researchers trying to apply MDS in actual practice was how to convert a distance rating to specific geometric information without knowing a priori the distance metric involved.

The breakthrough to practical implementations came in a two-part paper by **Shepard (1962)** and two papers by **Kruskal (1964a, 1964b)**. Shepard had the insight that the geometric configuration could be recovered without needing to know the specific distance metric by treating the perceived similarity between stimuli as reflecting some arbitrary monotonic function of an underlying distance metric, i.e., the subjects' estimated similarity S_{ij} between objects i and j was some function f of the true distance D_{ij} between them. The only requirement of the function was that it be monotonic (the value of the function always increases as the true distance increases); treating the estimated similarities as a rank-ordering, with some means for breaking ties, proved a simple way of generating such a monotonic function. Kruskal expanded on this notion by introducing the concept of stress, a measure of goodness-of-fit. With a means for computing the stress in a proposed configuration, an iterative computer program would be able to judge which of two possible configurations better fit the similarity data, and thus be able to converge on a numerical solution. KruskaL's own experience led him to characterize a stress value of .10 as "fair," while a value of 0.05 was considered "good"; stress values above 0.10 were deemed unfavorable, and values above .20 labeled "poor" (**Kruskal 1964a**).

Carroll and Chang (1970) introduced a metric model which incorporated individual differences among subjects in a MDS experiment. In addition to an object space, this model produces a subject space as well. The MDS program INDSCAL **(Carroll 1981)** is the computer implementation of Carroll and Chang's model. INDSCAL was later generalized into a family of multilinear MDS models called CANDECOMP (CANonical DECOMPosition of *N-way* tables) **(Carroll, et al., 1980 and Carroll and Pruzansky 1984). Takane, et al., (1977)** developed a nonmetric MDS model for individual differences which became the basis for the ALSCAL program **(Young 1981).**

Up to this point, MDS models and programs had all used a least squares criterion for determining how well the object space fit the raw data. **Ramsay (1977)** introduced a model which used a maximum likelihood criterion. The underlying distributional assumptions of the model allowed **Ramsay (1978)** to perform confirmatory MDS analyses. **Ramsay (1981)** implemented his model as the MDS program MULTISCALE. In this description of MULTISCALE, Ramsay outlined the use of diagnostic plots such as q-q plots for verifying the validity of the distributional assumptions and for detecting isolated wild departures from the distribution.

Parallel with these developments in MDS algorithms for deriving object spaces, research on interpreting the dimensions of the object space was being performed. **Carroll (1980)** presents a valuable summary of models for property (or preference) analysis. **Tucker (1960)** proposed a vector model in which the preference of objects is modeled by a vector through the object space. The vector model is a special case of the ideal point (or unfolding) model developed by **Coombs (1950)** (for the

unidimensional case) and by **Bennett and Hays (1960)** (for the multidimensional case). Several computer programs notably MDPREF **(Chang and Carroll 1969)** and PREFMAP **(Chang and Carroll 1972)** implemented the concepts of these models.

The use of procrustes statistics has attracted much interest as an area of research in comparing different object spaces. **Sibson (1978, 1979)** applied procrustes statistics to analyze the effects of small perturbations in distance on scaling applications. **Gower (1975)** presents a generalized technique for calculating a single procrustes statistic to compare *m* object spaces each containing the same *N* objects.

Recently, resampling techniques have been applied to MDS analyses to assess the stability of solutions. **Wish and Carroll (1982)** alluded to the use of the jackknife for these purposes. **Heiser and Meulman (1983)** used bootstrap techniques to compute confidence intervals for object space coordinates. **Weinberg, et al., (1984)** used the jackknife as a bootstrap technique to compute confidence intervals for object space coordinates. **DeLeeuw and Meulman (1986)** developed a specialized MDS jackknife to assess the stability and cross-validity of an object space solution.

A limited amount of research has explored diagnostic measures for MDS analyses. **Pruzansky, et al., (1982)** found two properties of proximity data which aided in identifying whether the data could be fit better by a spatial model or by a tree model. Proximity data with positive skewness and lesser elongation of triangles are better fit by spatial models (such as KYST object space solutions). Their study also supported the conclusions of an earlier study by **Graef and Spence (1979)** that small distances are less important in nonmetric MDS analyses than are large

distances. Graef and Spence generated proximities between 31 objects within a two-dimensional circle via random perturbation of the distances. They then deleted different thirds ? of the proximities according to proximity size (i.e., the largest third, the smallest third, and the middle third) and compared the recovered distances from the object spaces produced by the MDS program to the true distances between the objects on the circle.

With a practical algorithm available, an increasing number of researchers began to use MDS to explore similarity and categorization of data; a good survey of applications in various areas of investigation can be found in **Tversky and Hutchinson's (1986)** reanalysis paper. Problematic aspects of the method existed, however, and these would lead various researchers, notably Tversky, to question the appropriateness of MDS for various types of data. An entirely new method, cluster analysis, would result from examination of these problem areas, as well as a better understanding of the mathematics underlying both MDS and CA.

Hierarchical cluster analysis already existed as a concept; indeed, a basic paper by **Johnson (1967),** building on work by **Ward (1963)** as well as Shepard **(1962) and Kruskal (1964a, 1964b),** provided an alternate way to treat similarity within a few years of the first practical MDS algorithms. Hierarchical clustering and other types of clustering were now viewed as a way to deal with datasets that gave MDS methods problems. Two of the most severe problem areas were the presence of exemplars or prototypes in the dataset, and highly separable dimensions, an "apples and oranges" similarity situation.

Stress in an MDS configuration will be high (i.e., a poor goodness-of-fit will occur) if one of the objects in the stimulus set being evaluated is considered to be an exemplar of a larger class, or considered a prototype. For example, in the set (fruit, cherry, banana, watermelon, apple, orange, kiwi], "fruit" will be almost certainly be considered more similar to all the other objects than any other pair to each other since it is a generic, prototypical example of the category to which all the other objects belong. The only way to handle the presence of an exemplar geometrically without a significant amount of computed stress is by having all the other items distributed across the surface of a circle/sphere/equivalent higher-dimensional shape, while the exemplar sits at the center of the configuration. In general, this type of solution can deal with up to only N+2 items in an N-dimensional space (e.g., in two dimensions have three objects at the vertices of a triangle with the exemplar at the center of gravity of the triangle), although specific larger sets of objects might work **Tversky and Hutchinson (1986)** analyzed conditions when exemplar presence would cause difficulties for geometric configurations, and reanalyzed many prior studies by means of a nearest neighbor approach. In contrast to a geometric configuration, a tree structure of similarity can deal easily with the presence of an exemplar by locating the exemplar at the root node of the tree, while all other objects locate at the end nodes of the branches.

The other major problem area for traditional MDS, highly separable dimensions, is a situation where the different dimensions have little or nothing to do with each other. An example dating back to the great nineteenth-century psychologist William James is: the moon is like a ball because they are both round; the moon is also like a gas lantern because they both illuminate; but we do not think of a ball as being like a gas

lantern. Similarity in such cases can be thought of as a comparison/summation function across a range of independent or semi-independent features. In fact, this view of similarity as a matching process across a collection of object features led **Tversky (1977)** to propose cluster analysis as an alternate approach to similarity measurement. **Tversky and Gati (1982)** also demonstrated that for highly separable dimensions, basic mathematical assumptions of geometric modeling were violated. In particular, the triangle inequality, which states that for any objects i, j, and k and the distances D between them $d_{ij} + d_{jk} \geq d_{ik}$, is violated, because the "distance" along one dimension has nothing to do with the "distance" along another for the triangle inequality to be valid, a single distance metric must operate for all dimensions. Thus, two complimentary approaches to the analysis of similarity were necessary depending on the type of objects under study; as **Tversky and Hutchinson (1986**) state:

"Multidimensional scaling seems particularly appropriate for perceptual stimuli, such as colors and sounds that vary along a small number of continuous dimensions On the other hand, clustering representations seem particularly appropriate for conceptual stimuli, such as people or countries that appear to be characterized by a large number of discrete features"

.

There is a family of CA algorithms, but they all work in like fashion: given a similarity or distance matrix, some method is used to pick the pair of stimuli most like each other, group them into a single cluster, and derive a new reduced matrix. When the process is finished, the stimuli will be grouped into some sort of tree structure, where the distance between any pair of objects is related to the length of the path along the various branches separating them.

Hebert et al (2006) introduced fuzzy dissimilarity data, the fuzzy multidimensional scaling and the distance models which dissimilarities are expressed as intervals or fuzzy numbers. In these models each object is then no longer represented by a point but by a crisp or a fuzzy region in the chosen space. Furthermore they proposed two algorithms and illustrated to determine a fuzzy region in the chosen space.

III-Use of CA and MDS in Conjunction:

As noted above in the quote from **Tversky and Hutchinson (1986)**, there are various conditions under which it is more appropriate to use either MDS or CA. Even when standard MDS works for a given situation, however, CA can aid the researcher in interpreting an otherwise obscure set of dimensions. Figure 1 and 2 shows how cluster analysis can assist a researcher in interpreting the dimensions of a geometric configuration from an MDS. In figure 1, a reanalysis by **mmm** of letter similarity data collected by **Kuennapas and Janson (1969)**, the vertical dimension of the geometric configuration has rounded letters at the bottom and non-rounded ones at the top, but the interpretation of the horizontal dimensional is not obvious at first glance. Using a cluster analysis of the data, shown on figure 2, groupings of various sets of the letters become readily apparent.

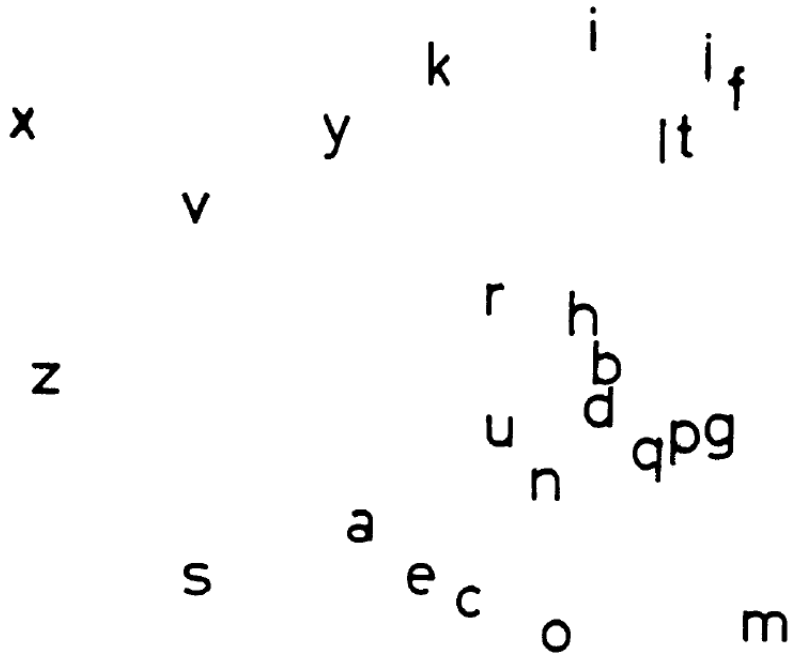


Figure 1

Representation of letter similarity (Kuennapas and Janson, 1969)

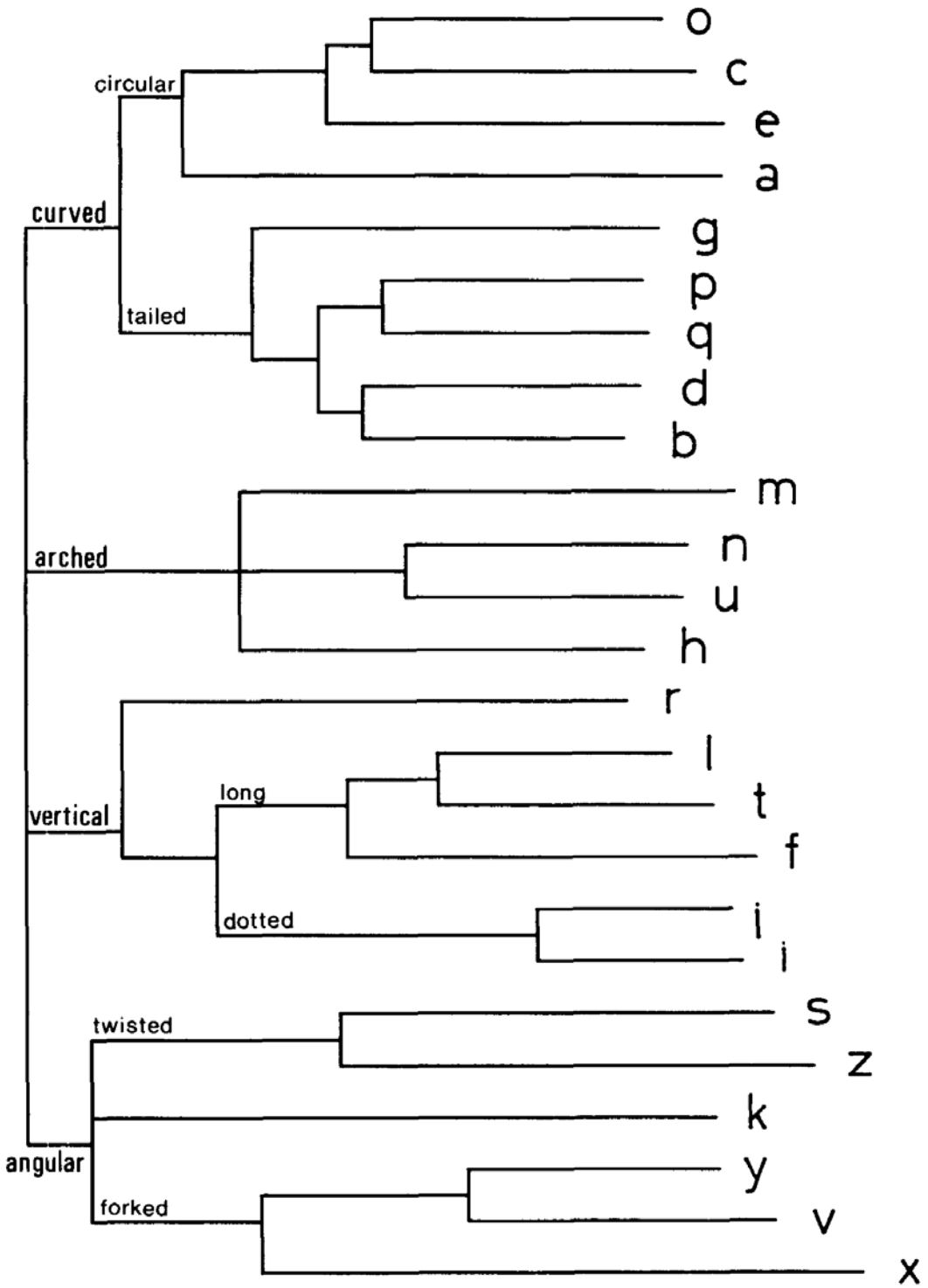


Figure 2

Representation of letter similarity (Kuennapas and Janson, 1969)

On the other hand, there are some situations where CA gives very bad fits even when separable dimensions are at work. For example, if the underlying data structure is a grid, any type of CA will yield a very poor fit since one point must be considered privileged (the root node of the clustering tree) and distances between the objects must be computed along the branches of the imposed tree structure. Figure 3 gives an example: the circles show the true configuration of the stimuli, with lines and intermediate nodes (the black dots and circle) connecting them in a tree. It is obvious that any two circles adjacent along a row or column should be rated as equidistant, but when one can move between circles only by traversing the tree, widely disparate "distances" will be registered. In practice, the researcher must be careful and explore many possibilities it is almost always better to use both MDS and CA on the same input data as cross-checks on each other.

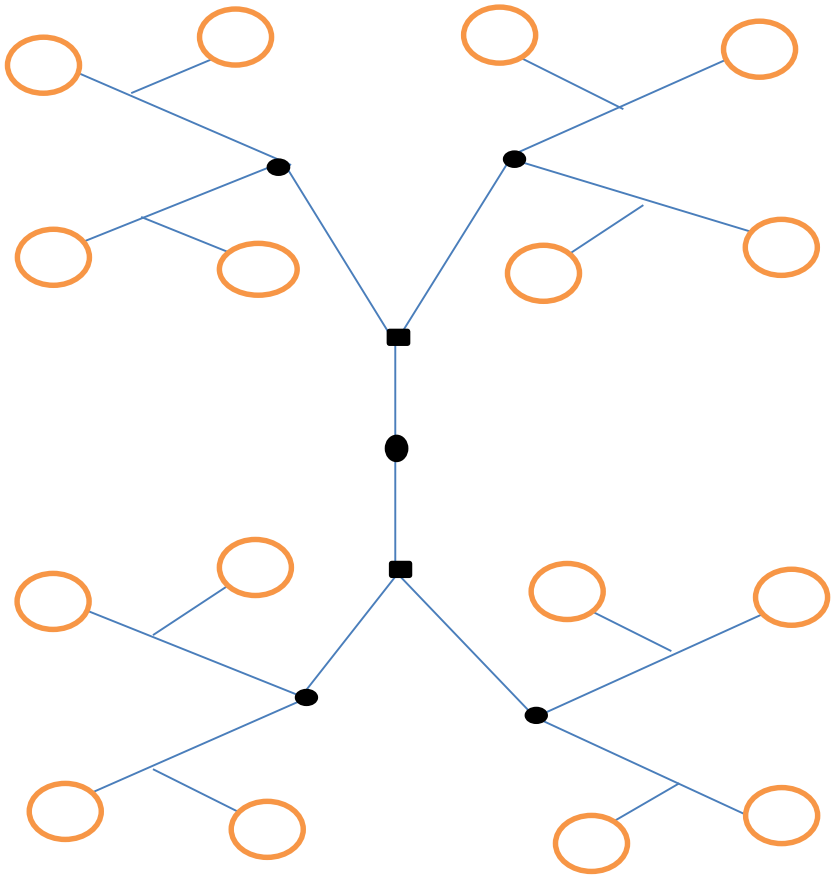


Figure 3: Poor Fit for a CA tree due to addition of privileged nodes.

1.4 The Outline of Research.

This thesis is organized as follows:

Chapter I : An introduction which represents a background on multidimensional scaling, cluster analysis and use of cluster analysis and multidimensional scaling in Conjunction, goals of our study and review studies.

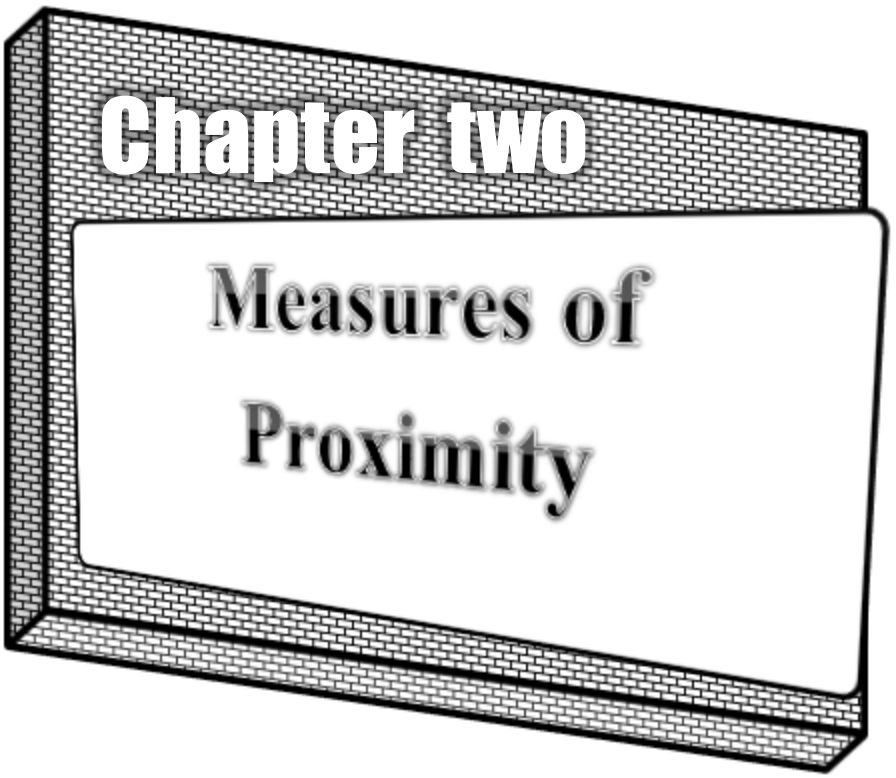
Chapter II: Measure of proximity which represents different patterns of proximity measures (similarity and dissimilarity).

Chapter III: Multidimensional scaling which represents Technique of multidimensional scaling in terms of concepts and methods.

Chapter IV: Cluster analysis (concepts and methods).

Chapter V : Clustering multidimensional scales.

Chapter VI: An application of cluster analysis and multidimensional scaling on some data obtained from Benha city car market


Chapter two
Measures of
Proximity

II. MEASURES OF PROXIMITY

2.1 Introduction:

In data mining, particularly in cluster analysis and multidimensional scaling, similarity, dissimilarity, and distance measures play an important role to calculate the proximity between data objects. The similarity matrix is constructed from the proximity measure. According to **Everitt (1980)**

There are many different measures available in the literature to calculate the proximity between data objects. One of the reasons for this variety is that these measures differ on the data type of the objects present in a given dataset. For instance, it follows that the proximity measures that are suitable for numeric variables may not be suitable for nominal data, as the attribute values from these two data types are represented differently. Therefore, a different set of measures is required to handle binary or nominal data. Moreover, the measures also differ on the properties they exhibit.

2.2 Similarity, Dissimilarity, and Distance measures:

2.2.1 Similarity

Similarity(s) is a numerical measure that represents the similarity (i.e. how alike various features and attributes). A similarity measure is considered a metric if it produces a higher value as the dependency between corresponding values in the sequences increases. A metric similarity satisfies the following properties (**Theodoridis and Koutroumba (2009).**

1. *Limited Range*: $S(X,Y) \leq S_0$, for some arbitrarily large number S_0.
2. *Reflexivity*: $S(X,Y) = S_0$ if and only if $X = Y$.

3. *Symmetry*: $S(X,Y) = S(Y,X)$.

4. *Triangle Inequality*: $S(X,Y)S(Y,Z) \leq [S(X,Y) + S(Y,Z)]S(X,Z)$.

S_0 is the largest similarity measure between all possible X and Y sequences.

This measure usually returns a non-negative value that falls in between 0 and 1. However, in some cases similarity may also range from -1 to +1. The Pearson Coefficient Correlation and the Angular Separation, are two examples where the similarity may take a negative value. When the similarity takes a value zero (0), it means that there is no similarity between the objects and these objects are very different from one another. In contrast, the value (1) denotes complete similarity, emphasizing that the objects are identical and possess the same attribute values.

2.2.2 Dissimilarity

The dissimilarity measure **[Webb (2002)]**, **[Han and Kamber (2006)]** is also a numerical measure, which represents the discrepancy or the difference between a pair objects. If two objects are very similar then the dissimilarity measure will have a lower value, where as if the objects are very different from one another, this measure will return a higher numeric value. Therefore, the measure is reversely related to the similarity measure. As such, when the similarity between two objects is high, the dissimilarity will be low and vice versa. As with the similarity score, the dissimilarity value also fall into the interval [0,1], but it may also take values ranging from —1 to +1.

2.2.3 Distance

The term distance, which is also commonly used as a synonym for the dissimilarity measure **[Meila and Shi (2000)],** computes the distance

between two data points in a multi-dimensional space. The distance measures always take a positive value between 0 and ∞. The distance measures also satisfy the following four properties **[Kandil, A. (2011) and Larose (2000)]**:

1. *d(x,y) = d(y,x),* for all points *x* and *y*. For instance, the distance from point *x* to point *y* is same as the distance from point *y* to point *x*.
2. *d(x, y)* = 0, if *x = y*. Distance is only 0 when both the coordinates are same.
3. *d(x, y)* > 0, for all points *x* and *y*. The distance is always non-negative.
4. *d(x,y)* < *d(x,z)* + *d(z,y),* for all points *x, y* and *z*. This is also known as the *Triangle Inequality*. This implies that introducing a third point may never shorten the distance between the two other points **[Larose (200)]**.

2.2.4 The relation between proximity measures:

Similarity and distance are, in a sense, inversely related to one another. When the distance in between two objects is large (meaning that the objects are different from one another), the similarity will be low. Conversely, when the distance is low the similarity will be high. Since it is inversely related, a common way to transform a distance measure to a similarity measure is by using the equation

$$s_{ij} = \frac{1}{d_{ij}}. \qquad (2.1)$$

Where

i and *j* are two objects.

One of the problems with this equation is that the similarity value will not always fall into the range [0,1].

For instance, if the distance between two objects is very small, such as $d_{ij=}0.25$, then the similarity value for these two objects will be

$$s_{ij=}\frac{1}{0.25}=4$$

There are various other ways to transform a distance or dissimilarity measure to a similarity measure such that the values for similarity measure ranges from 0 to 1.

If dissimilarity scores fall in between 0 and 1 then similarity is calculated using the following formula:

similarity = 1 — dissimilarity (2.2)

However, if the value for a distance measure is greater than 1, then there are different ways to transform a distance measure into a similarity measure. One such is the function given in Equation 2.2. This function is also known as the *Gaussian* function. **[Han and Kamber (2006)]**

$$s(x,y)=\exp(-\frac{d(x,y)^2}{2*\theta^2}) \quad (2.3)$$

Where

1. s(x,y) = similarity between points x and y.
2. d(x, y) = distance between points x and y.
3. θ = a user specified scaling variable. Shi and Malik suggested that the value of σ is set to 10 to 20 percent of the total range of the values obtained from the distance function *d(x, y)*.

There are several other ways to convert a distance measure into a similarity measure, as stated below:

$$s(x,y)=\frac{1}{1+d(x,y)^2} \quad (2.4)$$

$$s(x,y)=\frac{1}{1+d(x,y)} \quad (2.5)$$

$$s(x, y) = l - d(x, y) \qquad (d(x, y) \in [0,1]) \qquad (2.6)$$

2.3 Proximity Measures for Binary Variables:

Binary variables take only two values, such as: 0 (negative) and 1 (positive), *yes* (positive) and *no* (negative), or *agree* and *disagree*. These variables are usually categorized into two types:

1) *Symmetric binary variables* where both the positive and the negative values carry equal weight

2) *Asymmetric binary variables* where the positive and the negative values do not carry equal weight, and one (usually the positive value) carries more weight than the other.

Let *x* and *y* be two binary data points. Each proximity measure for binary data is represented by four variables (a, *b, c, d):*

a = number of occurrences of $x_i = 1$ and $y_i = 1$ (positive matches),

b = number of occurrences of $x_i = 0$ and $y_i = 1$ (disagreement),

c = number of occurrences of $x_i = 1$ and $y_i = 0$ (disagreement),

d = number of occurrences of $x_i = 0$ and $y_i = 0$ (negative matches),

and $a + b + c + d = p$ (total number of attributes in *x* and *y)*.

Therefore, numerous similarity coefficients were proposed by various researchers to calculate the proximities, which are also equally applicable to fields including data mining and statistics. A number of such coefficients give equal weight to the positive and negative values, whereas several coefficients ignore the negative matches. As such, for the same set of data, different coefficients may give different similarity values [**Everitt (1980**)].

We use the sample dataset given in Table 2.1 to compute the coefficients.

Object ID	Attribute 1	Attribute 2	Attribute 3	Attribute 4
Object 1	0	1	1	1
Object 2	1	1	1	1
Object 3	1	0	0	0

Table 2.1: Sample dataset for binary data type.

2.3.1 Jaccard Coefficient:

The *Jaccard* coefficient does not consider the negative matches. In terms of the four variables defined above, the Jaccard similarity coefficient is defined by Equation 2.7. Recall that, *a* denotes the number of positive matches whereas, b and c denote the total number of *disagreements.*

$$sim_{Jaccard} = \frac{a}{a+b+c} \qquad (2.7)$$

$$dis_{Jaccard} = \frac{b+c}{a+b+c} \qquad (2.8)$$

The values range from 0 to 1. The maximum similarity is achieved when $b = c = 0$ and the minimum similarity is achieved when there are no positive matches (when $a = 0$).

Example 2.3.1. The dissimilarity and the similarity between Object 1 and Object 2:

$$dis_{1,2} = \frac{1+0}{3+1+0} = \frac{1}{4} = 0.25$$

$$sim_{1,2} = 1 - 0.25 = 0.75$$

The dissimilarity and the similarity between Object 1 and Object 3:

$$dis_{1,3} = \frac{1+3}{0+1+3} = \frac{4}{4} = 1.0$$

$$sim_{1,3} = 1 - 1 = 0$$

2.3.2 Czekanowski Coefficient:

The Czekanowski similarity coefficient is also known as the Dice or Sorenson coefficient. The function is given in Equation 2.10. Recall that, *a* denotes the total number of positive matches. The total numbers of disagreements are denoted with the variables b and c.

$$sim_{Czekanowski} = \frac{2a}{2a+b+c} \quad (2.9)$$

$$dis_{Czekanowski} = \frac{b+c}{2a+b+c} \quad (2.10)$$

The coefficient is similar to the Jaccard coefficient. However, double weight is given to the variable *a* which denotes the total number of occurrences of the positive matches. By giving twice the weight to a, the function gives more emphasis to the positive matches. Variable *d* (when *x* = 0 and *y* = 0) is not present in this measure.

Example 2.3.2. The dissimilarity and the similarity between Object 1 and Object 2:

$$dis_{1,2} = \frac{1+0}{2*3+1+0} = \frac{1}{7} = 0.1429$$

$$sim_{1,2} = 1 - 0.1429 = 0.8571$$

The dissimilarity and the similarity between Object 1 and Object 3:

$$dis_{1,3} = \frac{1+3}{2*0+1+3} = \frac{4}{4} = 1.0$$

$$sim_{1,3} = 1 - 1 = 0$$

2.3.3 Sokal and Sneath Coefficient:

Sokal and Sneath proposed a similarity coefficient that is similar to the ones proposed by Jaccard and Czekanowski. This measure is defined as:

$$sim_{\text{Sokal and Sneath proposed}} = \frac{\frac{a}{2}}{\frac{a}{2}+b+c} = \frac{a}{a+2(b+c)} \quad (2.11)$$

$$dis_{\text{Sokal and Sneath proposed}} = \frac{b+c}{\frac{a}{2}+b+c} \quad (2.12)$$

However, in contrast to the Czekanowski coefficient which gives double weight to the positive matches (a), the Sokal and Sneath coefficient gives double weight to the disagreements in the denominator. The disagreements are represented by the variables *b* and c as denoted earlier. Thus, the Sokal and Sneath coefficient gives twice the weight on the combined disagreements denoted by *b* + c. By doing so, the coefficient actually gives slightly less weight to the positive matches compared to the Jaccard and Czekanowski coefficients

Example 2.3.3. The dissimilarity and the similarity between Object 1 and Object 2

$$dis_{1,2} = \frac{1+0}{1/2*3+1+0} = \frac{2}{5} = 0.4$$

$$sim_{1,2} = 1 - 0.4 = 0.6$$

The dissimilarity and the similarity between Object 1 and Object 3:

$$dis_{1,3} = \frac{1+3}{1/2 * 0 + 1 + 3} = \frac{4}{4} = 1.0$$

$$sim_{1,3} = 1 - 1 = 0$$

We suggest a new measurement which can represent all previous measures.

$$Sim_{m-general} = \frac{a}{a+m\,(b+c)} \quad , m \geq 0 \qquad (2.13).$$

$$dis_{m-general} = \frac{b+c}{ma+b+c} \quad , m \geq 0 \qquad (2.14).$$

2.3.4 Simple Matching Coefficient:

The Simple matching coefficient **[Webb (2002)],** also known as the Hamming distance, denotes the proportion of variables for which two variables have the same value [Webb (2002)]. As mentioned earlier, the variables *a* and *d* denote the total number of positive and negative matches, respectively. The variables *b* and c denote the total number of disagreement.

$$sim_{Simple\ matching\ coefficient} = \frac{a+d}{a+b+c+d} \qquad (2.15).$$

$$dis_{Simple\ matching\ coefficient} = \frac{b+c}{a+b+c+d} \qquad (2.16).$$

The Simple Matching Coefficient considers both, the positive matches (a) and the negative matches (d). Moreover, it gives equal weight to the positive and negative matches.

Example 2.3.4. The dissimilarity and the similarity between Object 1 and Object 2:

$$dis_{1,2} = \frac{1+0}{3+1+0+0} = \frac{1}{4} = 0.25$$

$$sim_{1,2} = 1 - 0.25 = 0.75$$

The dissimilarity and the similarity between Object 1 and Object 3:

$$dis_{1,3} = \frac{1+3}{0+1+3+0} = \frac{4}{4} = 1.0$$

$$sim_{1,3} = 1 - 1 = 0$$

2.3.5 Russell and Rao Coefficient:

The Russell and Rao similarity coefficient is sometimes known as the Positive matching coefficient [Webb (2002)]. The similarity function is defined in Equation 2.15.

$$sim_{\text{Russell and Rao}} = \frac{a}{a+b+c+d} \qquad (2.17).$$

$$dis_{S\,\text{Russell and Rao}} = \frac{b+c+d}{a+b+c+d} \qquad (2.18).$$

We suggest to write Russell and Rao coefficient in another formula

$$sim_{\text{Russell and Rao}} = \frac{a}{agrement\ values + disagrement\ values} \qquad (2.19).$$

$$dis_{\text{Russell and Rao}} = \frac{d+c+d}{agrement\ values + disagrement\ values} \qquad (2.20).$$

The Russell and Rao coefficient gives the proportion of the positive matches against the total number of variables (including the negative

matches). The coefficient is also sensitive to the meaning of positive and negative values. If the values are interchanged, then it will represent the proportion of the negative matches. The Russell and Rao coefficient achieves the maximum similarity when b = c = d = 0 (when there are only positive matches present) and scores the minimum when a = 0 (when there are no positive matches).

Example 2.3.5. The dissimilarity and the similarity between Object 1 and Object 2:

$dis_{1,2} = \frac{1+0+0}{3+1+0+0} = \frac{1}{4}$ =0.25

$$sim_{1,2} = 1 - 0.25 = 0.75$$

The dissimilarity and the similarity between Object 1 and Object 3:

$dis_{1,3} = \frac{1+3+0}{0+1+3+0} = \frac{4}{4} = 1.0$

$$sim_{1,3} = 1 - 1 = 0$$

2.3.6 Rogers and Tanimoto Coefficient:

The coefficient proposed by Rogers and Tanimoto is defined in Equation 2.17.

$$sim_{\text{Rogers and Tanimoto}} = \frac{\frac{(a+d)}{2}}{\frac{(a+d)}{2}+b+c} = \frac{a+d}{a+d+2(b+c)} \quad (2.21).$$

$$dis_{\text{Rogers and Tanimoto}} = \frac{b+c}{\frac{(a+d)}{2}+b+c} \quad (2.22).$$

The Rogers and Tanimoto coefficient is similar to the Simple Matching Coefficient. Where in matching Coefficient, the similarity coefficient considers both the positive and negative matches in the

equation and gives equal weight to them. However, in contrast to the Simple Matching Coefficient, the Rogers and Tanimoto coefficient gives double weight to the variables that represent the disagreements in the denominator (i.e. the variable b and c) [Sokal, R., and Sneath, P. (1963)]. We suggest anew coefficient from Rogers and Tanimoto coefficient defined in Equation 2.17.

$$dis_{\text{m general}} = \frac{m(b+c)}{(a+b)+m(b+c)} \quad ,m>0 \qquad (2.23).$$

$$sim_{\text{m general}} = \frac{a+d}{a+d+m(b+c)} \quad ,m>0 \qquad (2.24).$$

$$= \frac{agrement\ values}{agrement\ values + m(agrement\ values)}$$

Example 2.3.6. The dissimilarity and the similarity between Object 1 and Object 2:

$$dis_{1,2} = \frac{1+0}{\frac{(3+0)}{2}+1+0} = \frac{2}{5} = 0.4$$

$$sim_{1,2} = 1 - 40. = 0.6$$

The dissimilarity and the similarity between Object 1 and Object 3:

$$dis_{1,3} = \frac{1+3}{\frac{(0+0)}{2}+1+3} = \frac{4}{4} = 1.0$$

$$sim_{1,3} = 1 - 1 = 0$$

2.4 Proximity Measures for Numeric Variables:

There exist several distance measures for numeric or real-valued data. We present our discussion based on the measures presented in **[Webb (2002)], [Pedrycz (2005)], [Teknomo (2007)] and [Kandil, A. (2011)]** . For the purpose of clarification, we provide an example that shows the calculations for each of these distance measures. We use the sample dataset given in Table 2.2, which contains three data objects, and each of the objects is represented with four features.

Object ID	Attribute 1	Attribute 2	Attribute 3	Attribute 4
Object 1	10	5	8	2
Object 2	11	6	9	1
Object 3	1	20	0	8

Table 2.2: Sample dataset for numeric data type.

2.4.1 Euclidean Distance:

The *Euclidean distance* is one of the most widely used distance measures in the area of cluster analysis and multidimensional scaling **[Everitt (1980)].** The distance, in this case, is the straight-line distance between a given pair of data points. The distance is calculated as the summation of the differences between the coordinates of the data points x_i and x_j. The function is denoted as Equation 2.18.

$$d_{x_i x_j} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \qquad (2.25).$$

Example 2.5.1. The distance between Object 1 and Object 2 is calculated as:

$d_{1,2} = \sqrt{(10-11)^2 + (5-6)^2 + (8-9)^2 + (2-1)^2}$ =2

The distance between Object 1 and Object 3 is calculated as:

$d_{1,3} = \sqrt{(10-1)^2 + (5-20)^2 + (8-0)^2 + (2-8)^2}$ =20.1494

3.4.2 Manhattan Distance:

The *Manhattan distance* is also commonly known as the *city-block* distance. The Manhattan distance measure would travel from one point to another as if a grid-like path is followed. It is the summation of absolute differences between the coordinates of two data points *(x_i* and *x_j.)*.

$$d_{x_i x_j} = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \qquad (2.26).$$

Example 2.4.2. The distance between Object 1 and Object 2 is calculated as:

$d_{1,2} = |10-11|+|5-6| + |8-9| + |2-1|$ =4

The distance between Object 1 and Object 3 is calculated as:

$d_{1,3} = |10-1|+|5-20| + |8-0| + |2-8|$ =38

2.4.3 Minkowski Distance:

The *Minkowski distance* is defined in Equation 2.27

$$d_{x_i x_j} = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^{\lambda}\right)^{\frac{1}{\lambda}} \qquad (2.27).$$

In Equation 2.20, λ may take any value greater than 0. Depending on the value of λ, the Minkowski distance may take several different forms. For instance, when λ = 1, the Minkowski distance is similar to the Manhattan distance, whereas when λ = 2, the Minkowski distance is similar to the Euclidean distance **[Kandil, A. (2011)]**. A large value of λ indicates larger difference. However, a larger value of λ also indicates that the largest scale would dominate the total distance.

Example 2.4.3. The distance between Object 1 and Object 2 is calculated as (when λ = 3):

$d_{1,2} = \sqrt[3]{|10-11|^3 + |5-6|^3 + |8-9|^3 + |2-1|^3}$ =1.587

The distance between Object 1 and Object 3 is calculated as (when $\lambda = 3$):

$d_{1,3} = \sqrt[3]{|10-1|^3 + |5-20|^3 + |8-0|^3 + |2-8|^3}$ =16.9061

2.4.4 Chebyshev Distance:

The *Chebyshev distance* is a special case of the Minkowski distance with $\lambda = \infty$. In this case, the distance is measured as the distance between the coordinates of two data points where the absolute distance between the points in any single dimension is maximized.

$$d_{x_i x_j} = max_k |x_{ik} - x_{jk}| \qquad (2.28).$$

Example 2.4.4. The distance between Object 1 and Object 2 is calculated as:

$d_{1,2} = max(|10-11|, |5-6|, |8-9|, |2-1|)$ = max (1, 1, 1, 1) = 1

The distance between Object 1 and Object 3 is calculated as:

$d_{1,3} = max(|10-1|, |5-20|, |8-0|, |2-8|)$ = max (9, 15, 8, 6) = 15

2.4.5 Canberra Distance:

The *Canberra distance* is the summation of the series of fractional differences between coordinates of two data points (x_i and x_j). The Canberra distance is defined as Equation 2.29.

$$d_{x_i x_j} = \sum_{k=1}^{n} \frac{|x_{ik} - x_{jk}|}{|x_{ik} + x_{jk}|} \qquad (2.29).$$

The numerator of this equation signifies the difference between the objects, whereas the denominator normalizes the difference. Thus, the distance for each dimension may at most be 1.

Example 2.4.5. The distance between Object 1 and Object 2 is calculated as:

$$d_{1,2} = \frac{|10-11|+|5-6|+|8-9|+|2-1|}{|10+11|+|5+6|+|8+9|+|2+1|} = 0.5307$$

The distance between Object 1 and Object 3 is calculated as:

$$d_{1,3} = \frac{|10-1|+|5-20|+|8-0|+|2-8|}{|10+1|+|5+20|+|8+0|+|2+8|} = 3.0182$$

2.4.6 Mahalanobis Distance:

The *Mahalanobis distance* [Wikipedia (2008)], considers the correlation between variables. The Mahalanobis distance measure uses the *covariance matrix* to measure the variance and the correlation between the objects. Let $\vec{x}$and $\vec{y}$be two vectors and C^{-1}be the inverse covariance matrix. Then the Mahalanobis distance is calculated as:

$$d\,(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})c^{-1}(\vec{x}-\vec{y})^T} \qquad (2.30).$$

The main difference between the distance measures discussed so far and the Mahalanobis distance measure is that it considers the correlation between the variables. [Berkhin (2002)], [Xu and Wunsch (2005)], [**Kandil, A. (2011)]**.

However, one of the drawbacks of using the Mahalanobis distance measure is its high computational cost, which is due to the calculation required to construct the inverse covariance matrix. The Mahalanobis distance may not be suitable for high-dimensional datasets as covariance estimation may be inaccurate [Ghosh (2003)].

Example 2.4.6. The covariance matrix (C^{-1}) for the data in Table 2.2 is (as calculated by MATLAB):

$$\begin{vmatrix} 30.3333 & -45.6667 & 27.1667 & -20.8333 \\ -45.6667 & 70.3333 & -40.8333 & 31.1667 \\ 27.1667 & -40.8333 & 24.3333 & -18.6667 \\ 20.8333 & 31.1667 & -18.6667 & 14.3333 \end{vmatrix}$$

The distance between Object 1 and Object 2 is calculated as:

$(\vec{x}\text{- }\vec{y}) = [-1 \quad -1 \quad -1 \quad 1]$

$d_{1,2} =$

$\sqrt{[-1 \quad -1 \quad -1 \quad 1] * C^{-1} * [-1 \quad -1 \quad -1 \quad 1])^T}$ =1.9828

The distance between Object 1 and Object 3 is calculated as:

$(\vec{x}\text{- }\vec{y}) = [9 \quad -15 \quad 8 \quad -6]$

$d_{1,3} =$

$\sqrt{[9 \quad -15 \quad 8 \quad -6] * C^{-1} * [9 \quad -15 \quad 8 \quad -6])^T}$ =3.8609

2.4.7 Angular Distance:

The *Angular Separation* or *Cosine Distance* [Teknomo (2007)], measures the angular distance between the coordinates of two data points (x_i and x_j). Even though, this measure is called the Angular distance, it is a similarity measure rather than a distance measure. It represents the cosine angle between the unit vectors in the direction of the two pattern vectors [Webb (2002)] and thus the value lies between -1 and +1 as of the range of cosine angle. Though the angles are measured, it is meant to give the linear distance between the data points. A higher value of this function denotes

that the data objects are very similar to one another. The similarity and distance measure between object x_i and x_j is given below:

$$s_{x_i x_j} = \frac{\sum_{k=1}^{n} x_{ik} \cdot x_{jk}}{(\sum_{k=1}^{n} {x_{ik}}^2 . \sum_{k=1}^{n} {x_{jk}}^2)^{\frac{1}{2}}} \quad (2.31).$$

$$d_{x_i x_j} = 1 - \frac{\sum_{k=1}^{n} x_{ik} \cdot x_{jk}}{(\sum_{k=1}^{n} {x_{ik}}^2 . \sum_{k=1}^{n} {x_{jk}}^2)^{\frac{1}{2}}} \quad (2.32).$$

Type equation here.

The Angular distance is also scale invariant, and thus, the different units do not affect the result. The Angular distance considers the relative distance between the objects from a fixed point (the origin).

Example 2.4.7. The distance between Object 1 and Object 2 is calculated as:

$$d_{1,2} = 1 - \frac{(10 * 11) + (5 * 6) + (8 * 9) + (2 * 1)}{\sqrt{(10^2 + 5^2 + 8^2 + 2^2) * (11^2 + 6^2 + 9^2 + 1^2)}} = 0.0036$$

The distance between Object 1 and Object 3 is calculated as:

$$d_{1,3} = 1 - \frac{(10 * 1) + (5 * 20) + (8 * 0) + (2 * 8)}{\sqrt{(10^2 + 5^2 + 8^2 + 2^2) * (1^2 + 20^2 + 0^2 + 8^2)}} = 0.5794$$

2.4.8 Pearson Correlation Distance:

The *Pearson correlation coefficient* **[Teknomo (2007)]** measures similarity between data points. The values of this function ranges from -1 to +1. Since this measurement shows whether two data points are linearly related or not, a value of 1 shows that the points are lying on the same line and are positively correlated. A value of -1 indicates that the points are negatively correlated, whereas 0 means there is no linear correlation between the data points.

$$s_{ij}=\frac{\sum_{k=1}^{n}(x_{ik}-\overline{x_i}).(x_{jk}-\overline{x_j})}{(\sum_{k=1}^{n}(x_{ik}-\overline{x_i})^2.(\sum_{k=1}^{n}(x_{jk}-\overline{x_j})^2)^{\frac{1}{2}}} \qquad (2.33).$$

Where

$\overline{x_i}=\frac{1}{n}\sum_{k=1}^{n}x_{ik.}$, $\overline{x_j}=\frac{1}{n}\sum_{k=1}^{n}x_{jk}$.

The similarity function may be changed to correlation distance measure by subtracting from 1.

$$d_{ij}=1-\frac{\sum_{k=1}^{n}(x_{ik}-\overline{x_i}).(x_{jk}-\overline{x_j})}{(\sum_{k=1}^{n}(x_{ik}-\overline{x_i})^2.(\sum_{k=1}^{n}(x_{jk}-\overline{x_j})^2)^{\frac{1}{2}}} \qquad (2.33).$$

The Pearson coefficient correlation is used in the areas of microarray analysis and the document cluster analysis, amongst others. Since this distance measure considers the correlation between the objects, the outliers may affect the end results.

Example 2.4.8.

$\overline{x_1}=\frac{10+5+8+2}{4}=6.25$, $\overline{x_2}=\frac{11+5+9+1}{4}=6.75$, $\overline{x_3}=\frac{1+20+0+8}{4}=7.25$

The distance between Object 1 and Object 2 is calculated as:

$$d_{1,2}=1-\frac{(10-6.25)*(11-6.75)+(5-6.25)*(6-6.75)+(8-6.25)*(9-6.75)+(2-6.25)*(1-6.75)}{\sqrt{[(10-6.25)^2+(5-6.25)^2+(8-6.25)^2+(2-6.25)^2]*[(11-6.75)^2+(6-6.75)^2+(9-6.75)^2+(1-6.75)^2]}}$$

$$=1-\frac{45.2498}{5.6680}=.0092$$

The distance between Object 1 and Object 3 is calculated as:

$$d_{1,3}=1-\frac{(10-6.25)*(1-7.25)+(5-6.25)*(20-7.25)+(8-6.25)*(0-7.25)+(2-6.25)*(8-7.25)}{\sqrt{[(10-6.25)^2+(5-6.25)^2+(8-6.25)^2+(2-6.25)^2]*[(1-7.25)^2+(20-7.25)^2+(0-7.25)^2+(8-7.25)^2]}}$$

$$=1-\frac{-55.25}{96.7586}=1.57$$

2.5 Proximity Measures for Mixed Variables:

In the previous section, the discussion mostly focused on datasets of a particular variable type (e.g. binary). Nevertheless, in practical applications, it is possible to have more than one type of attribute in the same dataset. For instance, a dataset may have numeric and binary attributes to describe the objects **[Kandil, A. (2011)]**. In such cases, the conventional proximity measures for these two data types may not work well, as they are suitable to deal with one kind of variable at a time. Therefore, some similarity measures are proposed that incorporate information from various data types into a single similarity coefficient. The coefficients present in literature to calculate the similarity for mixed data type are, the Gower's General Dissimilarity Coefficient **[Han& Kamber (2006)]** and the Laflin's General Coefficient **[Laflin (1998**)], **[Kaufman& Rousseeuw (2005)].**

2.5.1 Gower's General Dissimilarity Coefficient:

The dissimilarity measure was introduced by Gower (1971)**.** The function is defined as follows:

$$d_{ij} = \frac{\sum_u \delta_{iju} d_{iju}}{\sum_u \delta_{iju}} \qquad (3.34).$$

Where

i. i , j are objects

ii. u is the variables

iii. $\sum_u \delta_{iju}$ is the number of variables

iv. The indicator $\delta_{ijU} = \begin{cases} 1 & if\ i \text{ and } j \text{ are nonmising for variabel } u \\ 0 & otherwise \end{cases}$

v. d_{iju} is the distance between object i and j for a variable u

d_{iju} is calculated using different distance measures that already exist for various variables types. For example:

• *If u is Numeric*

$$d_{iju} = \frac{|x_{iu} - x_{ju}|}{max_h x_{iu} - min_h x_{iu}}$$

where

h runs over all the non-missing objects of variable u.

• *If u is binary*

$$d_{iju} = \begin{cases} 0 & if\ x_{iu}=x_{ju} \\ 1 & otherwise \end{cases}$$

• *If u is Ordinal*

1. First compute the rank r_{iu} for object *i* assuming that the attribute *u* has M_u ordered states and $r_{iu} \in 1, \dots, M_u$
2. Replace x_{iu} by its corresponding rank.
3. Normalize r_{iu} by using the following formula:

$$Z_{iu} = \frac{r_{iu} - 1}{M_u - 1} \quad , \quad Z_{iu} \in [0.0,1.0].$$

4. Treat Z_{iu} as a numeric variable and a distance metric for the numeric variable is used to calculate the distance between the objects.

• *If u is Ratio-scaled*

(According to **Han and Kamber (2006)**], *"a ratio-scaled variable makes a positive measurement on a nonlinear scale, such as an exponential scale.")*, then the distance between its objects may be calculated in one of two ways.

Firstly by performing a logarithmic transformation and treating this transformed data as numeric values.

Secondly, by treating u as continuous ordinal data and calculating the distance as mentioned above.

	Attribute 1 (Numeric)	Attribute 2 (Numeric)	Attribute 3 (Nominal)	Attribute 4 (Nominal)	Attribute 5 (Ordinal)	Attribute 6 (Ordinal)
Object 1	12	10	A	A	Good	First
Object 2	9	12	A	A	Excellent	First
Object 3	3	4	B	C	Fair	Third

Example 2.5.1. For this example, we used a dataset (Table 2.3) similar to the one given in **[Han& Kamber (2006)].**

Table 2.3: Sample dataset for mixed data type.

To calculate the similarity between Object 1 and Object 2; we proceed as follows:

Attribute 1 (Numeric):

max = 12 and min = 3

$d_{1,2}(Attribute\ 1)=\frac{|12-9|}{|12-3|}$= 0.3333

Attribute 2 (Numeric):

max = 12 and min = 4

$d_{1,2}(Attribute2)=\frac{|10-12|}{|12-4|}=0.25$

Attribute 3 (Nominal):

$d_{1,2}(Attribute3) = 0$

Attribute 4 (Nominal):

$d_{1,2}(Attribute4) = 0$

Attribute 5 (Ordinal):

Rank: $Fair$ = 1, $Good$ = 2 and $Excellent$ = 3 and M_u = 3

The normalized values for Attribute 5 will be:

Object 1 $=\frac{2-1}{3-1}=0.5$

object2 $=\frac{3-1}{3-1}=1$

object3 $=\frac{1-1}{3-1}=0$

$d_{1,2}$(*Attribute5)*$=\frac{|.05-1|}{|1-0|}=0.5$

Attribute 6 (Ordinal):

Rank: *Third* =1, *Second* = 2 and First= 3 and M_u = 3

The normalized values for Attribute 6 will be:

Object 1 $=\frac{3-1}{3-1}=1$

object2 $=\frac{3-1}{3-1}=1$

object3 $=\frac{1-1}{3-1}=0$

$d_{1,2}$(*Attribute 6)*$=\frac{|1-1|}{|1-0|}=0$

The total dissimilarity between Object 1 and Object 2 are thus calculated as,

$d_{1,2}\frac{(1*0.3333)+(1*0.25)+(1*0)+(1*1)+(1*0.5)+(1*0)}{1+1+1+1+1+1}=\frac{1.0833}{6}=0.18055$

Next, the similarity may be derived by using Equation 2.5 as follows:

$similarity_{1,2}$= 1- 0.18055= 0.81945

To calculate the similarity between Object 1 and Object 3; we proceed as follows:

Attribute 1 (Numeric):

max = 12 and *min* = 3

$d_{1,3}$(*Attribute 1)*$=\frac{|12-3|}{|12-3|}=1$

Attribute 2 (Numeric):

max = 12 and *min* = 4

$d_{1,3}$(*Attribute2)*$=\frac{|10-4|}{|12-4|}=\frac{6}{8}=0.75$

Attribute 3 (Nominal):

$d_{1,3}$(*Attribute3)* = 1

Attribute 4 (Nominal):

$d_{1,3}$(*Attribute4)* = 1

Attribute 5 (Ordinal):

Rank: *Fair* = 1, *Good* = 2 and *Excellent* = 3 and M_u = 3

The normalized values for Attribute 5 will be:

Object 1 $=\frac{2-1}{3-1} = 0.5$

object2 $=\frac{3-1}{3-1} = 1$

object3 $=\frac{1-1}{3-1} = 0$

$d_{1,3}$(*Attribut5)*$=\frac{|.05-0|}{|1-0|}$*=0.5*

Attribute 6 (Ordinal):

Rank: *Third* =1, *Second* = 2 and First= 3 and M_u = 3

The normalized values for Attribute 6 will be:

Object 1 $=\frac{3-1}{3-1} = 1$

object2 $=\frac{3-1}{3-1} = 1$

object3 $=\frac{1-1}{3-1} = 0$

$d_{1,3}$(*Attribute2)*$=\frac{|1-0|}{|1-0|}$=1

The total dissimilarity between Object 1 and Object 3 are thus calculated as,

$d_{1,3}\frac{(1*1)+(1*0.75)+(1*1)+(1*1)+(1*0.5)+(1*1)}{1+1+1+1+1+1}=\frac{5.25}{6} = 0.8750$

Next, the similarity may be derived by using Equation 2.5 as follows:

$similarity_{1,3}$*= 1*- 0.8750= 0.1250

2.5.2 Laflin's General Coefficient:

The Laflin's coefficient is measured as follows. Let there *be* N_1 Binary attributes and N_2 Numeric attributes in a dataset. Let S_1 and S_2 be the similarity measures calculated for the Binary and the Numeric data respectively using some existing similarity measures (as discussed in Section 2.3 and Section 2.4 respectively). Then Laflin's coefficient [**Laflin (1998)]** is calculated as follows:

$$S_{(i,j)}= \frac{N_1.s_1+N_2.s2}{N_1+N_2} \qquad (2.35).$$

This function may be extended to include additional data types in a similar manner. For example, if each instance in a dataset contains four types of variables (i.e. *Binary, Numeric, Ordinal* and *Nominal)* then N_1, N_2, N_3 and N_4 will represent the total number of attributes for these four types of variables, respectively. Next, we calculate the similarity between each pair of objects using existing similarity measures, as discussed earlier, for each of these set of attributes separately. Let S_1, S_2, S_3 and S_4 be the similarity measure associated with the set of attributes N_1, N_2, N_3 and N_4, respectively. All these similarity values should be scaled so that they fall in between 0 and 1. The general similarity coefficient for this mixed set of attributes is calculated as:

$$S_{(i,j)}= \frac{N_1.s_1+N_2.s2+N_3.s_3+N_4.s4}{N_1+N_2+N_3+N_4} \qquad (2.36).$$

This equation ensures that each attribute makes an equal contribution to the measure of similarity between two objects *i* and *j* [Laflin (1998)].

Example 2.5.2. For the dataset given in Table 2.3, Laflin's coefficient is calculated as follows.

There are three different variable types in this dataset each type containing 2 variables. Thus, $N_1 = N_2 = N_3 = 2$.

To calculate the distance between nominal variables we use the formula given in **[Han& Kamber (2006)]:**

$$d_{(i,j)} = \frac{p-m}{p} \qquad (2.31)$$

Where

p is the total number of variables and

m is the number of variables for which i and j have the same value.

For numeric variables, the Euclidean distance measure as defined in Equation 2.18 is used and for all the cases distance measure is converted into a similarity measure by using Equation 2.4.

The similarity between Object 1 and Object 2 is calculated as follows.

Numeric variables:

$$d_1 = \sqrt{(12-9)^2 + (10-12)^2} = 3.4641$$

$$s_1 = \frac{1}{1+d_1} = \frac{1}{1+3.4641} = 0.2240$$

Nominal variables: $P = 2$ (total number of variables of type nominal)

$$d_{2=} \frac{2-2}{2} = 0$$

$$s_2 = \frac{1}{1+d_2} = \frac{1}{1+0} = 1$$

Ordinal variables:

Attribute 5 (Ordinal):

Rank: *Fair* = 1, *Good* = 2 and *Excellent* = 3 and M_u = 3

The normalized values for Attribute 5 will be:

Object 1 $= \frac{2-1}{3-1} = 0.5$

object2 $= \frac{3-1}{3-1} = 1$

object3 $= \frac{1-1}{3-1} = 0$

Attribute 6 (Ordinal):

Rank: *Third* =1, *Second* = 2 and First= 3 and M_u = 3

The normalized values for Attribute 6 will be:

Object 1 =$\frac{3-1}{3-1} = 1$

object2 =$\frac{3-1}{3-1}$ =1

object3 =$\frac{1-1}{3-1}$ =0

$$d_3 = \sqrt{(0.5-1)^2 + (1-1)^2} = 0.5$$

$$s_3 = \frac{1}{1+d_3} = \frac{1}{1+0.5} = 0.6667$$

When substituting the values of S_1, S_2 and S_3 in Equation 2.30, we obtain:

$$similarityx_{1,2} = \frac{(2*0.2240) + (2*1) + (2*0.667)}{2+2+2} = 0.6302$$

To calculate the similarity between Object 1 and Object 3, we proceed as follows.

Numeric variables:

$$d_1 = \sqrt{(12-3)^2 + (10-4)^2 = 10.8167}$$

$$s_1 = \frac{1}{1+d_1} = \frac{1}{1+10.8167} = 0.0846$$

Nominal variables:

$$d_{2=} \frac{2-2}{2} = 1$$

$$s_2 = \frac{1}{1+d_2} = \frac{1}{1+1} = 0.5$$

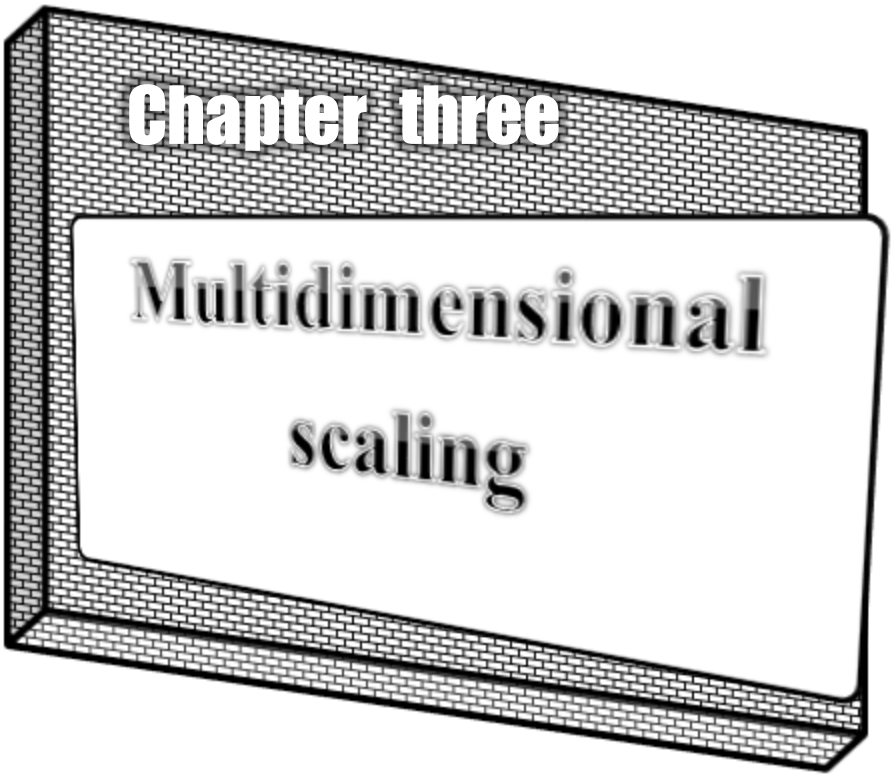
Ordinal variables:

$$d_3 = \sqrt{(0.5-0)^2 + (1-0)^2} = 1.118$$

$$s_3 = \frac{1}{1+d_3} = \frac{1}{1+1.118} = 0.4721$$

When substituting the values of S_1, S_2 and S_3 in Equation 2.30, we obtain:

$$similarityx_{1,3} = \frac{(2*0.0846) + (2*0.5) + (2*0.4721)}{2+2+2} = 0.352$$


Chapter three
Multidimensional
scaling

III. MULTIDIMENSIONAL SCALING

3.1 Introduction:

Multidimensional scaling (MDS) is a set of methods for discovering hidden structures in proximity (similarity or dissimilarity) measures between pairs of objects (**Borg and Groenen 2005**). Its primary objective is to display multivariate data in a lower dimensional space (usually Euclidean). The mapping roughly preserves the most important metric relationships of the original data and inherently clusters the data.

The MDS attempts to estimate the coordinates for each object in a lower dimensional space such that the distance for each pair matches the original dissimilarity measure as closely as possible. For example, MDS can be used to construct a 2-dimensional map based on distances between different locations. The estimated configuration of the objects and the dimensionality are two important issues for MDS. One main application of MDS is visualization (**Borg and Groenen 2005**), where we can represent a complex set of similarities or dissimilarities in a graphical map that is easier to see. Another application is exploration, where we can explore the main dimensions or clusters underlying the dissimilarities. MDS has its roots in the social and behavioral sciences. It has been widely used in many fields including the mapping of computer usage, the dimension reduction of marketing segmentation, the layout of sensor networks, and recently the construction of antigenic maps **(Borg and Groenen 2005, Garten, Davis, Russell and Smith 2009).**

3.2 What is multidimensional scaling?

In many situations, we have data on the interrelationships between a set of objects. These interrelationships might be, for example:

• Distances or the travel times between cities

• Words shared between members of a group of languages

• Frequencies with which libraries lend items to each other

• Frequencies with which journals cite each other

• Similarities between shades of colors

• Correlation between adjectives used to describe people.

In each of the cases listed above, the data take the form of a matrix **D**, whose components d_{ij} represent some measure of the similarity or dissimilarity between object *i* and object *j*. Each case is an example of a general and common situation. It would be useful to produce a mapping of the objects.

Multidimensional Scaling (MDS) Multidimensional scaling (MDS) **[Borg and Groenen (2005), Kruskal and Wish (1978), Torgerson (1952)]** is a general term that refers to techniques for constructing a map of generally high-dimensional data in to a target dimension(typically a low dimension)with respect to the given pairwise proximity information. Mostly, MDS is used to visualize given high dimensional data or abstract data by generating a configuration of the given data which utilizes Euclidean low-dimensional space, i.e. two-dimension or three-dimension.

Generally, proximity information, which is represented as an $n \times n$ dissimilarity matrix ($\Delta = [\delta_{ij}]$), where *n* is the number of points (objects) and δ_{ij} is the dissimilarity between point *i* and *j*, is given for the MDS problem, and the dissimilarity matrix (Δ) should agree with the following constraints:

(1) symmetricity ($\delta_{ij} = \delta_{ji}$)

(2) nonnegativity ($\delta_{ij} \geq 0$)

(3) zero diagonal elements($\delta_{ii} = 0$).

The objective of the MDS technique is to construct a configuration of a given high-dimensional data into low-dimensional Euclidean space, where each distance between a pair of points in the configuration is approximated to the corresponding dissimilarity value as much as possible.

The output of MDS algorithms could be represented as an $n \times m$ configuration matrix X, whose rows represent each data point x_i (i = 1, ...,n) in m-dimensional space. It is quite straight forward to compute the Euclidean distance between x_i and x_j in the configuration matrix X, i.e. $d_{ij} = \|x_i - x_j\|$, and we are able to evaluate how well the given points are configured in the m-dimensional space by using the suggested objective functions of MDS, called STRESS[**Kruskal (1964)]**or SSTRESS[**Takane et al. (1977)].**which are defining by the following definition:

STRESS difinition $\sigma(x) = \sum_{i<j\leq n} w_{ij} (d_{ij}(x) - \delta_{ij})^2$ (3.1)

SSTRESS difinition $\sigma^2(x) = \sum_{i<j\leq N} w_{ij} \left[(d_{ij}(x))^2 - (\delta_{ij})^2\right]^2$ (3.2)

where $1 \leq i < j \leq N$ and w_{ij} is a weight value, so $1 \geq w_{ij} \geq 0$.

As shown in the STRESS and SSTRESS functions, the MDS problems could be considered to be nonlinear optimization problem, which minimizes the STRESS or the SSTRESS function in the process of configuring an L-dimensional mapping of the high-dimensional data.

3.3 Multidimensional scaling methods:

Multidimensional scaling techniques can provide metric or non-metric solutions for the definition and interpretation of the object space.

Metric multidimensional scaling can be classical MDS (principal coordinates analysis) or least squares scaling.

In **metric scaling**, the object space distances must match as closely as possible the proximities of the proximity matrix; in metric property analysis, the vector or ideal point model must fit the degrees of the attribute for each object as closely as possible.

If δ_{ij} satisfies the triangle inequality ($\delta_{ij} < \delta_{ik} + \delta_{kj}$), the Euclidean distances d_{ij} between these coordinates match or nearly match the original dissimilarities. This is the metric MDS (Gordon, 1999).

Non-metric scaling is less restrictive than metric scaling. Instead of exactly matching the proximities, the object space distances must preserve only the ordering of the proximities; that is, if the proximity between objects *i* and *j* is greater than that between objects *k* and *l*, then the distance between objects i and j must be greater than the distance between objects *k* and *l* in the object space. Similarly, in non-metric property analysis, the vector or ideal point model must fit only the ordering of the degrees of the attribute; that is, if the attribute degree for object *i* is greater than that of object *j* , the property model of the attribute attempts only to preserve this ordering.

If δ_{ij} is an unknown monotonic increasing function $\delta_{ij} = f(d_{ij})$, Where d_{ij} is the Euclidean distance between objects i and j. δ_{ij} is The rank order of Euclidean distances between objects i and j in the new configuration match the original rank order of dissimilarities δ_{ij}, no matter δ_{ij} satisfies the triangle inequality or not. This is the nonmetric MDS **(Gordon, 1999**).

Metric least squares scaling and the nonmetric MDS method find a suitable configuration of points by minimizing a certain loss function. Classical scaling uses spectral decomposition on a doubly centered matrix of dissimilarities to find a lower dimensional display space (**Gower and Hand, 1996**).

The decision to use metric or non-metric MDS depends on the nature of the proximity and attributes data. If the data represent quantitative evaluations, then metric analysis is preferred. If the data consists of rankings (which do not have absolute quantitative value), non-metric analysis must be used. If metric analysis does not provide meaningful solutions, non-metric analysis is often applied on the chance that a more easily interpreted solution may be obtained. Usually, though, there is little difference between metric and non-metric solutions to the same proximity matrix. Some MDS programs provide statistical significance tests which are meaningful only for metric analysis (**Gower and Hand, 1996).**

3.3.1 Metric multidimensional scaling:

The purpose of the metric MDS is to find a new configuration (or coordinates) probably in a low dimensional space, such that the Euclidean distance of any pair of the new coordinates closely approximates the prescribed value. For example, how can we draw a map of Egypt if we only know the distances between all pairs of Egyptian cities?.

To complete the metric MDS, a principal coordinates analysis is employed first to find a new configuration from the given dissimilarity matrix. Then, a least squares scaling is applied afterwards to minimize the disparities between the original data's dissimilarities and the new configuration's dissimilarities.

3.3.1.1 Principal coordinates analysis (classical MDS):

Consider the following problem: looking at a map showing a number of cities, one is interested in the distances between them. These distances are easily obtained by measuring them using a ruler. Apart from that, a mathematical solution is available: knowing the coordinates x and y, the Euclidean distance between two cities i and j is defined by

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Now consider the inverse problem: having only the distances is it possible to obtain the map? Classical MDS, which was first introduced by **Torgerson (1952),** addresses this problem. It assumes the distances to be Euclidean. Euclidean distances are usually the first choice for an MDS space. There exist, however, a number of non-Euclidean distance measures, which are limited to very specific research questions (**Borg & Groenen, 1997)**. In many applications of MDS the data are not distances as measured from a map, but rather proximity data. When applying classical MDS to proximities it is assumed that the proximities behave like real measured distances. This might hold e. g. for data that are derived from correlation matrices, but rarely for direct dissimilarity ratings. The advantage of classical MDS is that it provides an analytical solution, requiring no iterative procedures.

Procedure for metric MDS (developed by Torgerson)

The classical scaling process of finding the configuration of points in the lower dimensional space for a set of dissimilarities δ_{ij} will be described in this section.

Suppose there are n objects with dissimilarities d_{ij} measured between all pairs of objects.

Construct the $n \times n$ matrix $\mathbf{A} = a_{ij} = -\frac{1}{2}\delta_{ij}{}^2$

Construct the $n \times n$ matrix $\mathbf{B} = b_{ij}$, with elements

$$b_{ij} = a_{ij} - a_{i.} - a_{.j} + a_{..},$$

Where

(i) $a_{i.} = \frac{1}{n}\sum_{j=1}^{n} a_{ij},$

(ii) $a_{j.} = \frac{1}{n}\sum_{i=1}^{n} a_{ij},$

(iii) $a_{..} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}.$

The matrix of squared Euclidean distances of the given coordinates $\Delta^2(X)$ or simply Δ^2 can be expressed by a simple matrix equation with respect to the coordinate matrix(X), as shown:

$$\Delta^2 = c\ 1^t + 1\ c^t - 2XX^t$$
$$= c\ 1^t + 1\ c^t - 2B$$

Where

(i) c is the diagonal elements of XX^t,

(ii) $1 = (1,1,\ldots,1)^T$ a column vector of n ones,

(iii) 1^t is transpose of 1,

(iv) c^t is transpose of c,

(v) X^t is transpose of X,

(vi) $B = XX^t$.

The centering $(n \times n)$ matrix **H** can be defined as

$$\mathbf{H}=\boldsymbol{I}_n - \frac{1}{n}\underline{1}\,\underline{1}^t$$

Where

I is the identity matrix, which translates a matrix to a column centered matrix by multiplying them. By multiplying the left and the right sides by the centering matrix **H**, a process called the *double centering* operation, we can introduce the following equations:

$$H\Delta^2 H = H(c1^t + 1c^t - 2XX^t)H$$
$$= Hc1^tH + H1c^tH - H2BH$$
$$= Hc0^t + 0c^tH - 2HBH$$
$$= -2HBH$$
$$= -2B$$

Since the centering of a vector of ones turns out to be a vector of zeros ($1^tH = H1 = 0$), the first two terms are eliminated. Without a loss of generality, we can assume that the coordinate matrix(X) is a column centered matrix. Then, the result of the double centering operation on the B matrix is equal to B itself, since X is a column centered matrix. Therefore, we can define the relation between B and D^2as follow:

$$B = -\frac{1}{2}H\Delta^2 H$$

$$\mathbf{B=HAH}$$

Where

$$\boldsymbol{A} = -\frac{1}{2}\Delta^2$$

The configuration of points can be found by expressing **B** in terms of its spectral decomposition (**Gower and Hand, 1996**) as

$$\mathrm{B=V}\Lambda V^T,$$

Where

$\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, the diagonal matrix of eigenvalues λ_i of B

$\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$.

The matrix of corresponding eigenvalues is $\mathbf{V} = [\underline{V_1}, \underline{V_2}, \ldots, \underline{V_n}]$ where the eigenvectors are normalized such that $\underline{V_i}^T \underline{V_i} = 1$ for all $i = 1,2,\ldots,n$.

The configurations of the points in r dimensional display space can be represented by the coordinated matrix X: $n \times r$ given by

$$\mathbf{X} = \boldsymbol{V}_r \boldsymbol{\Lambda}_r^{1/2},$$

Where

The columns of matrix $\boldsymbol{V}_r : n \times r$, consists of the first r eigenvectors of **B** that correspond to the r largest eigenvalues of **B**,

The matrix $\boldsymbol{\Lambda}_r^{1/2} = \text{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \ldots, \lambda_r^{1/2})$.

The coordinate matrix **X** will be used to display the points which represent the objects. It must be remembered that the arbitrary sign of the eigenvectors $\boldsymbol{V}_i$ leads to invariance of the configurations with respect to reflection in the origin. The display space will not necessarily be Euclidean.

Cox and Cox (2001) points out that if **B** is positive semi-definite of rank r, then a configuration in r dimensional Euclidean space can be found, so that the associated distances between the points δ_{ij} are such that

$$\delta_{ij} = d_{ij} \text{ for all } i, j.$$

How many dimensions should be used in the display space?

It is easily shown that B has at least one zero eigenvalue, since

$$\mathbf{B}\underline{\mathbf{1}} = \mathbf{HAH}'\underline{\mathbf{1}} = \underline{0}$$

Where

$\underline{0}$ represents a vector of n zeroes.

A configuration of points in any $r = n$-1 dimensional Euclidean space can therefore always be found. The configuration obtained could be rotated to its principal axes in the principal component sense (**Cox and Cox, 2001**). The principle axes are orthogonal to each other. Only the first $r(r \leq n-1)$ principal axes are chosen for representing the objects, as this will explain the maximum variation in *r* dimensions. It turns out that **X** already has the points referred to their principal axes, since

$$\boldsymbol{XX}^T = (\boldsymbol{V}_r \boldsymbol{\Lambda}_r{}^{1/2})(\boldsymbol{V}_r \boldsymbol{\Lambda}_r{}^{1/2})^t$$
$$= \boldsymbol{\Lambda}_r{}^{1/2} \boldsymbol{V}_r{}^T \boldsymbol{V}_r \boldsymbol{\Lambda}_r{}^{1/2} = \boldsymbol{\Lambda},$$

Where

Λ is a diagonal matrix.

The distances between the points in the full $n-1$ dimensional Euclidean space are given by

$$\delta_{ij}{}^2 = \sum_{s=1}^{n-1} \lambda_s (x_{is} - x_{js})^2 ,$$

And hence relatively small eigenvalues contribute far less to the squared distance $\delta_{ij}{}^2$. If only r eigenvalues of **B** are retained as being significantly large, then *r* dimensional Euclidean space spanned by the first *r* eigenvectors of B can be used to represent the objects.

Definition 3.1. a goodness of fit measure

The Eigen decomposition is variance-maximizing. That is, each successive dimension (i.e., eigenvector) "explains" the maximum amount of variance remaining in the data, after taking any previous dimensions into account.

The eigenvalues measure the variance explained by each dimension, and the sum of the eigenvalues is equal to the variance of the entries in **B**.

The proportion of variance accounted for by the m dimensions in the

MDS solution is given by the sum of the first r eigenvalues, divided by the sum of all eigenvalues (there will usually be n nonzero eigenvalues):

$$\text{Metric MDS Fit} = \frac{\sum_{s=1}^{r} \lambda_s}{\sum_{s=1}^{n-1} \lambda_s}$$

Cox and Cox (2000) suggest a measure when B is not positive semi-definite:

$$\text{Metric MDS Fit} = \frac{\sum_{s=1}^{r} \lambda_s}{\sum_{s=1}^{n-1} |\lambda_s|}$$

Choice of *r* can then be assessed with this measure, but for practical purpose, *r* will usually be chosen to be 2 or 3.

Basic steps in a classical MDS algorithm are:

1. Construct the $n \times n$ matrix $\mathbf{A} = a_{ij} = -\frac{1}{2}\delta_{ij}^{\,2}$

2. Construct the $n \times n$ matrix **B=HAH**

Where

H is $n \times n$ the centering matrix

$$\mathbf{H} = \boldsymbol{I}_n - \frac{1}{n}\underline{1}\,\underline{1}^T$$

3. Extract the n largest positive eigenvalues of $\lambda_1 \ldots \lambda_n$ of the matrix B and the corresponding n eigenvectors $e_1 \ldots e_n$.

4. m-dimensional spatial configuration of the n objects is derived from the coordinate matrix $\mathbf{X} = \boldsymbol{V}_r \boldsymbol{\Lambda}_r^{\,1/2}$ Where the columns of matrix $\boldsymbol{V}_r$ with size $n \times r$,consists of the first *r* eigenvectors of **B** that correspond to the *r* largest eigenvalues of **B**, and the matrix

$\boldsymbol{\Lambda}_r^{\,1/2} = \text{diag}(\lambda_1^{\,1/2}, \lambda_2^{\,1/2}, \ldots, \lambda_r^{\,1/2})$.

Example 3.3.1.1:

In order to illustrate classical MDS, assume that we have measured the distances between A, B, C, and D on a map. Therefore, the proximity matrix (showing the distances in millimeters) might look like

	A	B	C	D
A	0	93	82	133
B	93	0	52	60
C	82	52	0	111
D	133	60	111	0

The matrix of squared proximities is

$$[\mathbf{A}]_{ij} = [d_{ij}]^2 = \begin{bmatrix} 0 & 8649 & 6724 & 17689 \\ 8649 & 0 & 2704 & 3600 \\ 6724 & 2704 & 0 & 12321 \\ 17689 & 3600 & 12321 & 0 \end{bmatrix}$$

Since there are n = 4 objects, the matrix H is calculated by

$$\mathbf{H} = \boldsymbol{I}_n - \frac{1}{n}\underline{1}\,\underline{1}^T$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - 0.25 \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0{:}75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0{:}75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0{:}75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0{:}75 \end{bmatrix}$$

$$\mathrm{B} = -\frac{1}{2}\mathrm{HAH} = \begin{bmatrix} 5035.0625 & -1553.0625 & 258.9375 & -3740.9375 \\ -1553.0625 & 507.8125 & 5.3125 & 1039.9375 \\ 258.9375 & 5.3125 & 2206.8125 & -2471.0625 \\ -3740.9375 & 1039.9375 & -2471.0625 & 5172.062 \end{bmatrix}$$

The eigenvalues of B

$\lambda_{1=}$ 9724.168

$\lambda_{2=}$ 3160.986

$\lambda_{3=}$ -0.001

$\lambda_{4=}$ 36.596

For a two-dimensional representation of the four points, the first two largest eigenvalues and the corresponding eigenvectors of B have to be extracted

$\lambda_{1=}9724{:}168,$ $\lambda_{2=}3160{:}986,$

$$e_1 = \begin{bmatrix} -0{:}637 \\ 0{:}187 \\ -0{:}253 \\ 0{:}704 \end{bmatrix}, \qquad e_2 = \begin{bmatrix} -0{:}586 \\ 0{:}214 \\ 0{:}706 \\ 0{:}334 \end{bmatrix}$$

Finally the coordinates of the points (up to rotations and reflections) are obtained by multiplying eigenvalues and -vectors

$$X = \begin{bmatrix} -0.637 & -0.586 \\ 0.187 & 0.214 \\ -0.253 & 0.706 \\ 0.704 & -0.334 \end{bmatrix} \begin{bmatrix} \sqrt{9724.168} & 0 \\ 0 & \sqrt{3160.986} \end{bmatrix} = \begin{bmatrix} -62.831 & -32.97448 \\ 18.403 & 12.02697 \\ -24.960 & 39.71091 \\ 69.388 & -18.76340 \end{bmatrix}$$

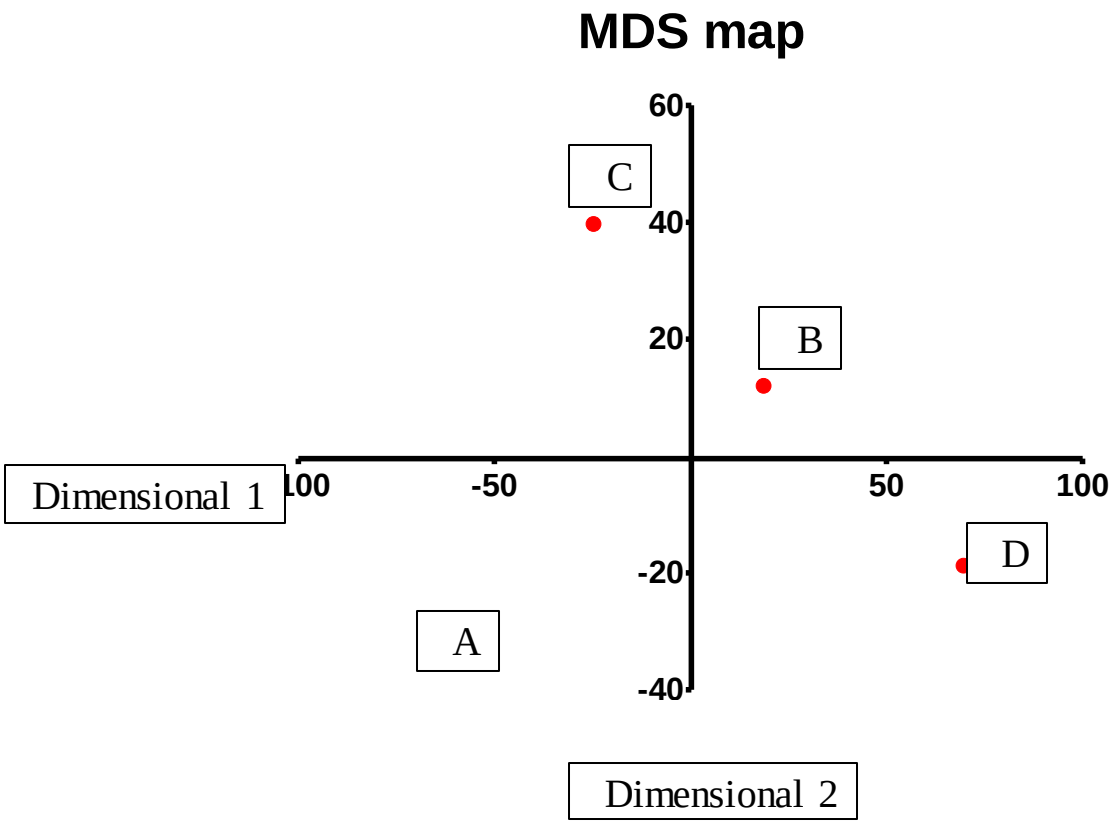


Figure 3.1: Classical MDS representation of the four points

3.3.1.2Metric least square scaling:

Metric least square scaling is a metric MDS method that find configuration $\boldsymbol{X}: n \times r$ by matching δ_{ij} to d_{ij} by minimizing a certain loss function (Cox and cox 2001).

where

δ_{ij} is the distance between points *i* and *j* in this *m*- multidimensional space $\boldsymbol{X}: n \times r$

d_{ij} is the Euclidian distance between points *i* and *j*

. A tow dimensional space (m=2) is usually used.

Loss function for Metric MDS

Various loss functions have been suggested in the past. Minimizing different loss function produce different optimal configuration $\boldsymbol{X}: n \times r$. **Borg and Groenen (2005)** used a general loss function, which will be referred to as Raw Stress:

$$\text{Raw Stress} = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2 \qquad (3\text{-}3)$$

where:

(i) d_{ij}: is the Euclidean distance between points i and j in the graphical display,

(ii) δ_{ij}:is the dissimilarity between objects i and j,

(iii) w_{ij}: are weights. Which can be specified to emphasize different pairs. For instance, if there are missing data, we may set

The weights are usually chosen as

$$w_{ij} = 0 \text{ if } \delta_{ij} \text{ is missing}$$

$$w_{ij} = 1 \text{ if} \delta_{ij} \text{is known}$$

Other values of w_{ij} are also allowed and different choices of w_{ij} lead to different loss functions **(Borg and Groenen, 2005)**.

More generally, let

$$w_{ij} = d^q{}_{ij}\,.$$

Different choices of q can be used to emphasize the representation of small or large dissimilarities. Large negative values of q may lead to a better representation of small dissimilarities, but not large dissimilarities.

Conversely, large positive values of q lead to a better representation of large dissimilarities, but not small dissimilarities. For a relative presentation of both small and large dissimilarities, choose q=−2. If the

dissimilarities have some clustering, then choosing a large value of q may reveal a clearer clustering structure **(Borg and Groenen, 2005).**

Normalized Stress valus

Normalized Stress should be used to avoid scale dependency, where

$$\text{Normalized Stress} = \frac{\sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2}{\sum_{i<j} w_{ij} d_{ij}^{\,2}} \qquad (3\text{-}4)$$

Normalized Stress values in (3-4) depend on many factors **(Borg and Groenen, 2005) wich is follows :**

(i) The higher n, representing the number of points, the higher the normalized in general.

(ii) The higher r, the dimensionality of display space, the lower the normalized Stress values.

(iii) The larger the squared errors ($\delta_{ij} - d_{ij}$), the higher the normalized Stress value

Loss functions, such as normalized Stress in (3-4), are indices that assess the mismatch between the dissimilarities and corresponding distances.

The residual plot and bubble plot

The residual plot and bubble plot can be used to describe this mismatch between the dissimilarities and corresponding distances.

The residual plot

The residual plot is a scatter diagram of the distances and the dissimilarities. A bisector is draw from the lower left corner to the upper

right corner, and the dissimilarities are drawn on this bisector. The size of dissimilarities can therefore be noted immediately. The corresponding distances, that should match the original dissimilarities as well as possible, are also draw in this residual plot. The vertical distance between the dissimilarities and the corresponding distance is a measure of the corresponding error $e_{ij} = (\delta_{ij} - d_{ij})$. The error gives an indication of the size of mismatch between of the dissimilarities and the corresponding distances. Large error will cause a higher Normalized Stress value in (3-4). The residual plot gives an indication of which dissimilarities are better represented in the metric MDS display, but the residual plot does not give an indication of how well the original objects are represented by points in the display.

The bubble plot

The bubble plot can be used to assess the fit of each point. The bubble plot uses the Stress per point measure, which is defined by **Borg and Groenen (2005)** as follows:

Stress per point is the average of the squared errors between the current object and all other objects.

The bubble plot still uses the same configuration of points as the metric MDS plot to display the objects. The only different is that the bubble plot uses bubbles to represent the objects, where bubbles with a larger radius indicate points with better fit. The viewer can therefore immediately see which objects see better represented in the display.

In practice, a two-dimensional display is mostly used to display the final configuration **X**: $n \times r$ with r = 2. It is also possible to display the final configuration X: $n \times r$ in a three-dimensional graph, with r=3. A three-dimensional display will have a lower final Normalized Stress (3-4) and Row Stress (3-3) value than a two-dimensional display. The normalized

Stress (3-4) value in a two-dimensional display can be assessed by considering the upper and lower bounds of The Normalized Stress (3-4) value. The Normalized Stress value in (3-4) has the following lower and upper bounds in a two-dimensional display: [0 , 0.4352] which were derived by De Leeuw and Stoop (1984) Stress by assuming that the points lie equally spaced on a circle. Then, Stress is smaller than $[12\cot^2(\pi 2n)/(n2 - n)]^{1/2}$ with the limit $[1 - 8/\pi^2]^{1/2} = .4352$ **(Borg and Groenen 2005).**

Local Minima

MDS algorithms usually end up in a *local minimum*. Various methods can be used to minimize Normalized Stress (3-4) or Raw Stress (3-3). The aim of these methods is to find an optimal configuration X:$n \times r$ of points, from which distances d_{ij} can be calculated. The optimal configuration will be the configuration that produces distances that best match the dissimilarities δ_{ij} , in the sense that a minimum stress value is reached. These methods usually operate in an iterative manner by changing the configuration of points in each step, until either a minimum stress value or a specified maximum number of iterations is reached. These minimizing methods will usually require an initial configuration of points. It is common practice to use the configuration produced by classical scaling as the initial configuration **(Borg and Groenen 2005).** Random initial configurations, where points are randomly produced using a uniform distribution, can also be used.

The *method of dimension reduction* repeats the MDS analysis, starting from a high dimensionality (say, 10) and then reducing the dimensionality of the solution space stepwise (down to 2, say). The local minimum configuration of the higher-dimensional analysis is used as a

start configuration for the MDS analysis in one dimension lower by dropping the dimension that accounts for the least variance (i.e., the last principal component). Proceeding in this manner, one hopes that the low-dimensional solution is a global minimum ***multiple random starts*, or *multistart***, consists of running the MDS analysis from many (say, 100) different random starting configurations and choosing the one with the lowest Stress. Using multistart and making some mild assumptions, an estimate for the expected total number of local minima can be given. Then, the total expected number of local minima n_t is

$$n_t = \frac{n_m(n_s - 1)}{n_s - n_m - 2}$$

n_t is the total expected number of local minima.

n_s *is* the number of multistart start configurations.

n_m is the number of different local minima obtained.

If n_s is approximately equal to n_t, then we may assume that all local minima are found. The one with the lowest Stress is the candidate global minimum.

The SMACOF algorithm for metric MDS

The SMACOF algorithm used for metric MDS methods operates in an iterative manner by changing the configuration of points in each step of the algorithm.

Borg and Groenen (2005) suggest using Normalized Stress (3-2) rather than Raw Stress (3-1) as loss functions, because using the latter may lead to degenerate solutions.

These degenerate solutions are configurations that were obtained by making the loss function very small, irrespective of the relationship between distance and the data.

The SMACOF algorithm ensures that the Normalized Stress value in (3-2) reaches a local minimum, but the local minimum may not be a global minimum.

The steepest descent methods can also not guarantee that the local minimum found is indeed the global minimum. Borg & Groenen (2005, p.276) point out that local in MDS are not necessarily bad.

A final configuration with a slightly worse fit is acceptable if it has a clearer interpretation than a configuration with a better fit. The problem of whether the local minimum is indeed the global minimum can be overcome in several ways.

One possibility is to use multiple starting configurations where the whole SMACOF algorithm is repeated for each starting configuration and a minimum Normalised Stress value in (3-4) is noted. The final chosen configuration will be the overall configuration of all the configurations, produced from each starting configuration, which leads to the lowest Normalised Stress value in (3-2) another possibility is to use the tunneling method (Borg and Groenen 200).

The SMACOF algorithm for metric MDS can be summarized by:

1. Set $\mathbf{Z} = \mathbf{X}^{[0]}$, where $\mathbf{X}^{[0]}$ is some (non)random start configuration.

Set $k = 0$. Set ε to a small positive constant.

2. Compute $\delta_r{}^{[0]} = \delta_r\ (\mathbf{X}^{[0]})$. Set $\delta_r{}^{[-1]} = \delta_r{}^{[0]}$.

3. While $k = 0$ or ($\delta_r{}^{[k-1]} - \delta_r{}^{[k]} > \varepsilon$ and $k \leq$ maximum iterations) do

4. Increase iteration counter *k* by one.

5. Compute the Guttman transform $\mathbf{X}^{[k]}$). by

$X^{u} = n^{-1}B(Z)Z$ if all $w_{ij} = 1$,

$X^{u} = v^{+}B(Z)Z$ otherwise.

Where

i. $B(Z)$has elements $b_{ij} = -\frac{w_{ij}\delta_{ij}}{d_{ij}}$,

$b_{ij} = -\sum_{j=1}^{n} b_{ij}$,$\forall i \neq j$.

ii. $v^{+} = n^{-1}\mathbf{H}$.

iii. $\mathbf{H}=\boldsymbol{I}_n - \frac{1}{n}\underline{1}\,\underline{1}^{T}$.

6. Compute $\delta_r{}^{[k]} = \delta_r\ (\mathbf{X}^{[k]})$.

7. Set $\mathbf{Z} = \mathbf{X}^{[k]}$.

8. End while

Example 3.2

To illustrate the **SMACOF** algorithm, consider the following example the dissimilarities $\boldsymbol{\Delta}$ and the starting configuration $\mathbf{X}^{[0]}= \mathbf{Z}$ as following

$$\boldsymbol{\Delta} = \begin{bmatrix} 0 & 5 & 3 & 4 \\ 5 & 0 & 2 & 2 \\ 3 & 2 & 0 & 1 \\ 4 & 2 & 1 & 0 \end{bmatrix} \qquad \mathbf{Z}= \begin{bmatrix} -0.266 & -0.539 \\ 0.451 & 0.252 \\ 0.016 & -0.238 \\ -0.200 & 0.524 \end{bmatrix}$$

The elements of the D(Z) are given by $\boldsymbol{d_{ij}} = \sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{t}}(\boldsymbol{x_i} - \boldsymbol{x_j})}$

$d_{12} = \sqrt{(-0.266 - 0.451)^2 + (-0.539 - 0.252)^2}$=1.068

$d_{13} = \sqrt{(-0.266 - 0.016)^2 + (-0.539 - -0.238)^2}$=0.412 `

$d_{14} = \sqrt{(-0.266 - -0.200)^2 + (-0.539 - 0.524)^2}$=1.065

$d_{23} = \sqrt{(0.451 - 0.016)^2 + (0.252 - -0.238)^2}$=0.655

$d_{24} = \sqrt{(0.451 - -0.200)^2 + (0.252 - 0.524)^2}$=0 .706

$d_{34} = \sqrt{(0.016 - -0.200)^2 + (-0.238 - 0.524)^2}$=0.792

D(Z)= $\begin{bmatrix} 0.000 & 1.068 & 0.412 & 1.065 \\ 1.068 & 0.000 & 0.655 & 0 .706 \\ 0.412 & 0.655 & 0.000 & 0.792 \\ 1.065 & 0.706 & 0.792 & 0.000 \end{bmatrix}$

Compute B(Z)

The elements of the B(Z) are given by $b_{ij} = b_{ij} = -\frac{w_{ij}\delta_{ij}}{d_{ij}}$,

$$b_{ij} = -\sum_{j=1}^{n} b_{ij} \ ,\forall i \neq j$$

We assume that all w_{ij} = 1.

$b_{12} = -w_{12}\delta_{12}/d_{12}(z) = -5/1.068 = -4.682$

$b_{13} = -w_{13}\delta_{13}/d_{13}(z) = -3/0.412 = -7.273$

$b_{14} = -w_{14}\delta_{14}/d_{14}(z) = -4/1.065 = -3.756$

$b_{11} = -(b_{12} + b_{13} + b_{14}) = -(-4.682 - 7.273 - 3.756) = 15.712.$

$b_{23} = -w_{23}\delta_{23}/d_{23}(z) = -2/0.655 = -3.052$

$b_{24} = -w_{24}\delta_{24}/d_{24}(z) = -2/0.706 = -2.835$

$b_{22} = -(b_{21} + b_{23} + b_{24}) = -(-4.682 - 3.052 - 2.835) = 10.570$

$b_{34} = -w_{34}\delta_{34}/d_{34}(z) = -1/0.792 = -1.263$

$b_{33} = -(b_{31} + b_{32} + b_{34}) = -(-7.273 - 3.052 - 1.263) = 11.588$

$b_{44} = -(b_{41} + b_{42} + b_{43}) = -(-3.756 - 2.835 - 1.263)$
$= 7.853$

$$B(Z) = \begin{bmatrix} 15.712 & -4.682 & -7.273 & -3.756 \\ -4.682 & 10.570 & -3.052 & -2.835 \\ -7.273 & -3.052 & 11.588 & -1.263 \\ -3.756 & -2.835 & -1.263 & 7.853 \end{bmatrix}$$

Compute $\delta_r^{[0]}$

$\delta_r^{[0]} = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2 = 34.2992$

(i,j)	δ_{ij}	d_{ij}	$(\delta_{ij} - d_{ij})^2$
(1,2)	5	1.068	15.4606
(1,3)	3	0.412	6.6977
(1,4)	4	1.065	8.6142
(2,3)	2	0.655	1.8090
(2,4)	2	0.706	1.6744
(3,4)	1	0.792	0.0433
$\sum$			34.2992

Compute the first update X^u by the Guttman transform

$X^u = n^{-1}B(Z)Z$

$$= \frac{1}{4}\begin{bmatrix} 15.712 & -4.682 & -7.273 & -3.756 \\ -4.682 & 10.570 & -3.052 & -2.835 \\ -7.273 & -3.052 & 11.588 & -1.263 \\ -3.756 & -2.835 & -1.263 & 7.853 \end{bmatrix}\begin{bmatrix} -0.266 & -0.539 \\ 0.451 & 0.252 \\ 0.016 & -0.238 \\ -0.200 & 0.524 \end{bmatrix}$$

$$X^u=\begin{bmatrix}-1.415 & -2.471\\ 1.633 & 1.107\\ 0.249 & -0.067\\ -0.468 & 1.431\end{bmatrix}$$

The elements of the D (X^u) are given by $d_{ij}=\sqrt{(x_i-x_j)^t(x_i-x_j)}$

$d_{12}=\sqrt{(-1.415-1.633)^2+(-2.471-1.107)^2}$=4.700

$d_{13}=\sqrt{(-1.415-0.249)^2+(-2.471--0.067)^2}$=2.923

$d_{14}=\sqrt{(-1.415--0.468)^2+(-2.471-1.431)^2}$=4.016

$d_{23}=\sqrt{(1.633--0.249)^2+(1.107--0.067)^2}$=1.815

$d_{24}=\sqrt{(1.633--0.468)^2+(1.107-1.431)^2}$=2.126

$d_{34}=\sqrt{0.249--0.468)^2+(-0.067-1.431)^2}$=1.661

$$D(X^u)=\begin{bmatrix}0.000 & 4.700 & 2.923 & 4.016\\ 4.700 & 0.000 & 1.815 & 2.126\\ 2.923 & 1.815 & 0.000 & 1.661\\ 4.016 & 2.126 & 1.661 & 0.000\end{bmatrix}$$

To find The elements of the B(X^u) We assume that all w_{ij} = 1.

The elements of the B(X^u) are given by $b_{ij}=b_{ij}=-\frac{w_{ij}\delta_{ij}}{d_{ij}}$,

$$b_{ij}=-\sum_{j=1}^{n}b_{ij}\ ,\forall i\neq j.$$

$b_{12}=-w_{12}\delta_{12}/d_{12}(z)=-5/4.700=-1.064$

$b_{13}=-w_{13}\delta_{13}/d_{13}(z)=-\frac{3}{2.923}=-\ 1.026$

$b_{14}=-w_{14}\delta_{14}/d_{14}(z)=-4/4.016=-\ 0.996$

$b_{11} = -(b_{12} + b_{13} + b_{14}) = 3.086$

$b_{23} = -w_{23}\delta_{23}/d_{23}(z) = -2/1.815 = -1.102$

$b_{24} = -w_{24}\delta_{24}/d_{24}(z) = -2/2.126 = -0.941$

$b_{22} = -(b_{21} + b_{23} + b_{24}) = 3.107$

$b_{34} = -w_{34}\delta_{34}/d_{34}(z) = -1/1.661 = -0.6020$

$b_{33} = -(b_{31} + b_{32} + b_{34}) = 2.539$

$b_{44} = -(b_{41} + b_{42} + b_{43}) = 2.539$

$$B(X^u) = \begin{bmatrix} 3.086 & -1.064 & -1.026 & -0.996 \\ -1.064 & 3.107 & -1.102 & -0.941 \\ -1.026 & -1.102 & 2.539 & -0.6020 \\ -0.996 & -0.941 & -0.6020 & 2.539 \end{bmatrix}$$

set $\boldsymbol{X}^{[u]} = \boldsymbol{X}^{[1]}$ and compute $\delta_r\ \boldsymbol{X}^{[1]}$

$\delta_r\ \boldsymbol{X}^{[1]} = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2 = 0.6758$

(i,j)	δ_{ij}	d_{ij}	$(\delta_{ij} - d_{ij})^2$
(1,2)	5	4.700	0.09
(1,3)	3	2.923	0oo6
(1,4)	4	4.016	0.003
(2,3)	2	1.815	0.034
(2,4)	2	2.126	0.1059
(3,4)	1	1.661	0.4369
$\sum$			0.6758

The difference of $\delta_r\ \boldsymbol{X}^{[0]}$ and $\delta_r\ \boldsymbol{X}^{[1]}$

is large, 33.71531530, so it makes sense to continue the iterations.

The second update is

$$\boldsymbol{X}^{[2]} = n^{-1}\mathrm{B}\left(\boldsymbol{X}^{[1]}\right)\boldsymbol{X}^{[1]}$$

=

$$\frac{1}{4}\begin{bmatrix} 3.086 & -1.064 & -1.026 & -0.996 \\ -1.064 & 3.107 & -1.102 & -0.941 \\ -1.026 & -1.102 & 2.539 & -0.6020 \\ -0.996 & -0.941 & -0.6020 & 2.539 \end{bmatrix}\begin{bmatrix} -1.415 & -2.471 \\ 1.633 & 1.107 \\ 0.249 & -0.067 \\ -0.468 & 1.431 \end{bmatrix}$$

$$\boldsymbol{X}^{[2]}=\begin{bmatrix} 1.473 & -2.540 \\ 1.686 & 1.99 \\ 0..154 & 0678 \\ -0..366 & 1.274 \end{bmatrix}$$

Continue the iterations until the difference in subsequent Stress values is less than 10^{-6}.

3.3.2Nonmetric MDS:

Nonmetric multidimensional scaling is also known as ordinal multidimensional scaling.

The assumption that proximities behave like distances might be too restrictive, when it comes to employing MDS for exploring the perceptual space of human subjects. In order to overcome this problem, **Shepard (1962) and Kruskal (1964**) developed a method known as nonmetric multidimensional scaling. In nonmetric MDS, only the ordinal information in the proximities is used for constructing the spatial configuration.

As mention before, the nonmetric MDS method abandon the metric nature of the transformation

$$\delta_{ij} = f(\mathrm{d}_{ij}),$$

Where

$f(function)$ can now be arbitrary.

The only requirement for nonmetric MDS is that the transformation must preserve the rank order of dissimilarities. The aim with Nonmetric MDS is to find an optimal configuration X: $n \times r$ by matching the disparities $\hat{d}_{ij}$($\hat{d}_{ij}$ is the disparity between objects *i* and *j*) to d_{ij} by minimizing a certain loss function. This is similar to the metric least squares scaling method, the difference being that the dissimilarities δ_{ij} in the loss functions are now replaced by disparities, $\hat{d}_{ij}$. The actual dissimilarities value (δ_{ij}) are only used to determine the rank-order of the disparities, $\hat{d}_{ij}$, This means

$$\delta_{ij} < \delta_{kl} \Rightarrow \quad \hat{d}_{ij} < \hat{d}_{kl}.$$

The disparities are also sometimes called pseudo distances. These disparities which are chosen in an optimal manner will be discussed later.

The loss function of nonmetric MDS

The loss function used by **Brog and Groenen (2005)** for nonmetric MDS is very similar to the loss function used for metric MDS. This loss function will also be referred to as Raw stress, with

$$\text{Raw Stress} = \sum_{i<j} w_{ij}(d_{ij} - \hat{d}_{ij})^2 \qquad (3\text{-}5)$$

Where

d_{ij} is the Euclidean distance between points *i* and *j*

$\hat{d}_{ij}$ is the disparity between objects *i* and *j*

w_{ij} the weights must contain non-negative values.

The weights are usually chosen as

$w_{ij} = 0$ if *i=j* and

$w_{ij} = 1$ otherwise.

Other value of w_{ij} are also allowed and different choices of w_{ij} lead to different loss functions **(Brog andGroenen, 2005).**

The Raw Stress value in (3-5) is a badness-of-fit measure, but it is not very informative. A large value does not necessarily indicate a bad fit, as it depends on the scale of distances in the configuration $\mathbf{X}$:$n \times r$. Normalized Stress can be used remove the scale dependency where

$$\text{Normalized Stress} = \frac{\sum_{i<j} w_{ij}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} w_{ij} d_{ij}^{\ 2}} . \qquad (3\text{-}6)$$

The aim of the nonmetric MDS method:

The aim of the nonmetric MDS method is to find an optimal configuration $\mathbf{X}$: $n \times r$ of points, from which distances d_{ij} and disparities $\hat{d}_{ij}$ can be calculated, that will minimize the Normalized Stress (3-6) or Raw Stress (3-5) loss functions. However, the minimizing of the Normalized Stress (3-6) function is not an easy task. The minimizing is usually done by an iterative process. The difference between the iterative process of the metric least squares scaling method and this iterative process is that the disparities $\hat{d}_{ij}$ also need to be optimally chosen, which depends on the distances d_{ij} . The distances d_{ij} , in turn, depend on the configuration $\mathbf{X}$: $n \times r$, which changes during each iteration. Therefore, the disparities $\hat{d}_{ij}$and the distances d_{ij} need to be optimally chosen during each step of the iteration. The SMACOF algorithm can again be minimize the Normalized Stress (3-6) or Raw Stress (3-5) loss functions.

Scaling by a MAjorizing of a COmplicated Function (SMACOF)

The SMACOF algorithm used for the nonmetric MDS method is described in detail by **Brog and Groenen (2005).**

1. Set $\mathbf{Z} = \mathbf{X}^{[0]}$, where$\mathbf{X}^{[0]}$ is some (non)random start configuration.

Set $k = 0$. Set ε to a small positive constant.

2. Find optimal disparities $\hat{d}_{ij}$ for fixed distances d_{ij} ($\mathbf{X}^{[0]}$).

3. Compute $\delta_r{}^{[0]} = \delta_r\ (\hat{d}\ , \mathbf{X}^{[0]})$. Set $\delta_r{}^{[-1]} = \delta_r{}^{[0]}$.

4. While $k = 0$ or ($\delta_r{}^{[k-1]} - \delta_r{}^{[k]} > \varepsilon$ and $k \leq$ maximum iterations) do

5. Increase iteration counter k by one.

6. Compute the Guttman transform $\mathbf{X}^{[k]}$). by

$X^u = n^{-1}B(Z)Z$ if all $w_{ij} = 1$,

$X^u = v^+B(Z)Z$ otherwise.

Where

i. $B(Z)$ has elements $b_{ij} = -\frac{w_{ij}\delta_{ij}}{d_{ij}}$,

$b_{ij} = -\sum_{j=1}^{n} b_{ij}\ , \forall i \neq j.$

ii. $v^+ = n^{-1}\mathbf{H}$.

iii. $\mathbf{H} = \boldsymbol{I}_n - \frac{1}{n}\underline{1}\,\underline{1}^T$.

7. Find optimal disparities $\hat{d}_{ij}$ for fixed distances d_{ij} ($\mathbf{X}^{[k]}$).

8. Compute $\delta_r{}^{[k]} = \delta_r\ (\hat{d}, \mathbf{X}^{[k]})$.

9. Set $\mathbf{Z} = \mathbf{X}^{[k]}$.

10. End while

Brog and Groenen (2oo5) suggest using Normalized Stress (3-7) rather than Raw Stress (3-6) as loss function, because using the latter may lead to degenerate solutions.

Heiser (1991) also points out that negative disparities could lead to degenerate solutions.

The SMSCOF algorithm ensures that the Normalized Stress value in (3-7) reaches a local minimum, but the local minimum may not be a global minimum. The problem of whether the local minimum is indeed the global

minimum can be overcome using multiple initial configurations or by using the tunneling method.

Monotone regression with Kruskal's up-and-down-blocks algorithm:

Kruskal's least-squares monotonic transformation (or *monotone regression)* is used MDS techniques for fitting object space distances to the raw proximity data. We use the following example to illustrate this way.

Example 3.

Table 3.1 presents a proximity matrix for 5 objects and Table 3.2 presents the distances between the objects in the object space derived at this point in the MDS program.

	A	B	C	D	E
A	0	**1**	**8**	**3**	**4**
B	**1**	**0**	**7**	**2**	**10**
C	**8**	**7**	**0**	**5**	**9**
D	**3**	**2**	**5**	**0**	**6**
E	**4**	**10**	**9**	**6**	**0**

Table 3.1 Proximity Matrix

	A	B	C	D	E
A	**0**	**1.0**	**6.4**	**1.5**	**3.0**
B	**1.0**	**0**	**6.2**	**2.5**	**8.4**
C	**6.4**	**6.2**	**0**	**4.2**	**8.2**
D	**1.5**	**2.5**	**4.2**	**0**	**8.4**
E	**3.0**	**8.4**	**8.2**	**8.4**	**0**

Table 3.2. Distance

It is the goal of the monotone regression procedure to find a least-square monotonic fit of the distances to the proximities. In this way, a

comparison may be made to see if the current space is a proper solution to the MDS analysis.

Since the data and distance matrices are symmetric, we only need handle the lower-half matrix of each in the procedure; the upper-half of the resulting disparity matrix is merely a symmetric reflection of the lower-half.

The first step is to arrange the proximity cells into ascending order of proximity. The outcome of this step is shown in the first two columns of Table 3.3. The distances for these cells are shown in the third column of Table3.3. If the distances perfectly fit the given proximities in the proximity matrix, the distances in column three should also be in ascending order. Since they are not in such order, they are transformed into disparities to measure the departure from the perfect fit.

The transformation consists of a series of comparisons of distances in the order given in Table 3.3. Each time a distance is found out of place (i.e. the series descends instead of ascends), the distances of concern are equalized to satisfy minimally the monotonic requirement.

In the example, the series of comparisons proceeds from top to bottom.

1. *The first distance* 1.0 does not exceed the second distance 2.5; so, these distances fit the monotonic relation established by the proximity matrix.
2. *The second distance* 2.5 exceeds the third distance 1.5. To correct this relation, each of these two distances is replaced with their mean 2.0. Thus, *the second distance and third distance* have been replaced by disparities 2.0.
3. *The forth distance* 3.0 does not exceed the fifth distance 4.2; so, these distances fit the monotonic relation established by the proximity matrix.

4. *The fifth distance* 4.2 does not exceed the sixth distance 8.4; so, these distances fit the monotonic relation established by the proximity matrix.
5. *The sixth distance* 8.4 is compared to the seventh distance 6.2. *The sixth* and *seventh disparities* become 7.3, the mean of 8.4 and 6.2. Now, however, the *seventh disparity* 7.3 exceeds the *eighth distance* 6.4. In this case, *the sixth, seventh,* and *eighth disparities* become 7.0, which is the mean of the sixth, seventh, and eighth distances (8.4 + 6.2 + 6.4)73. This disparity exceeds the fifth distance 4.2 and does not exceed the ninth distance 8.2.
6. *The ninth distance* 8.2 does not exceed the tenth distance 8.4; so, these distances fit the monotonic relation established by the proximity matrix.

The calculation of the disparities has been completed with the result shown in the last column of Table3.3.

OBJECT PAIR	PROXIMITY	DISTANCE	DISPARITY
AB	1	1.0	1.0
BD	2	2.5	2.0
AD	3	1.5	2.0
AE	4	3.0	3.0
CD	5	4.2	4.2
DE	6	8.4	7.0
BC	7	6.2	7.0
AC	8	6.4	7.0
CE	9	8.2	8.2
BE	10	8.4	8.4

Table 3.3. Example of Disparity Calculation

Example 3.3

To illustrate the **SMACOF** algorithm, consider the following example the dissimilarities $\mathbf{\Delta}$ and the starting configuration $\mathbf{X}^{[0]}= \mathbf{Z}$ as following

$$\mathbf{\Delta} = \begin{bmatrix} 0 & 3 & 2 & 5 \\ 3 & 0 & 1 & 4 \\ 2 & 1 & 0 & 6 \\ 5 & 4 & 6 & 0 \end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix} 3 & 2 \\ 2 & 7 \\ 1 & 3 \\ 10 & 4 \end{bmatrix}$$

The elements of the D($\mathbf{X}^{[0]}$) are given by $d_{ij} = \sqrt{(x_i - x_j)^t (x_i - x_j)}$

$d_{12} = \sqrt{(3-2)^2 + (2-7)^2} = 5.1$

$d_{13} = \sqrt{(3-1)^2 + (2-3)^2} = 2.2$

$d_{14} = \sqrt{(3-10)^2 + (2-4)^2} = 7.3$

$d_{23} = \sqrt{(2-1)^2 + (7-3)^2} = 4.1$

$d_{24} = \sqrt{(2-10)^2 + (7-4)^2} = 8.5$

$d_{34} = \sqrt{(1-10)^2 + (3-4)^2} = 9.1$

$$\mathrm{D}(\mathbf{X}^{[0]}) = \begin{bmatrix} 0 & 5.1 & 2.2 & 7.3 \\ 5.1 & 0 & 4.1 & 8.5 \\ 2.2 & 4.1 & 0 & 9.1 \\ 7.3 & 8.5 & 9.1 & 0 \end{bmatrix}$$

Compute disparities $\hat{d}_{ij}$ for D ($\mathbf{X}^{[0]}$)

- *The first distance* 4.1 exceeds the *second* distance 2.2. To correct this relation, each of these two distances is replaced with their mean. Thus, *the first distance and second distance* have been replaced by disparities 3.17.

- *The third distance* 5.1 does not exceed the forth distance 8.5; so, these distances fit the monotonic relation established by the proximity matrix.
- *The forth distance* 8.5 exceeds the *fifth* distance 7.3. To correct this relation, each of these two distances is replaced with their mean. Thus, *the forth distance and fifth distance* have been replaced by disparities 7.9

(i,j)	δ_{ij}	d_{ij}	$\hat{d}_{ij}$
2,3	1	4.1	3.17
1,3	2	2.2	3.17
1,2	3	5.1	5.1
2,4	4	8.5	7.9
1,4	5	7.3	7.9
3,4	6	9.1	9.1

Compute B($\mathbf{X}^{[0]}$)

The elements of the B(Z) are given $b_{ij} = -\frac{w_{ij}\delta_{ij}}{d_{ij}}$,

$$b_{ij} = -\sum_{j=1}^{n} b_{ij} \ ,\forall i \neq j.$$

We assume that all w_{ij} = 1.

$b_{12} = -w_{12}\hat{d}_{12}/d_{12}(z) = -5.1/5.1 = -1$

$b_{13} = -w_{13}\hat{d}_{13}/d_{13}(z) = -3.17/2.2 = -1.44$

$b_{14} = -w_{14}\hat{d}_{14}/d_{14}(z) = -7.9/7.3 = -1.08$

$b_{11} = -(b_{12} + b_{13} + b_{14}) = -(-1 - 1.44 - 1.08) = 3.52$

$b_{23} = -w_{23}\hat{d}_{23}/d_{23}(z) = -3.17/4.1 = -0.773$

$b_{24} = -w_{24}\hat{d}_{24}/d_{24}(z) = 7.9/8.5 = -0.929$

$b_{22} = -(b_{21} + b_{23} + b_{24}) = -(-0.929 - 0.773 - 1) = 2.702$

$b_{34} = -w_{34}\hat{d}_{34}/d_{34}(z) = -9.1/9.1 = -1$

$b_{33} = -(b_{31} + b_{32} + b_{34}) = -(-1.44 - 0.773 - 1) = 3.213$

$b_{44} = -(b_{41} + b_{42} + b_{43}) = -(-1.08 - 0.929 - 1) = 2.009$

$$\mathbf{B}(\mathbf{X}^{[0]}) = \begin{bmatrix} 3.52 & -1 & -1.44 & -1.08 \\ -1 & 2.702 & -0.773 & -0.929 \\ -1.44 & -0.773 & 3.213 & -1 \\ -1.08 & -0.929 & -1 & 2.009 \end{bmatrix}$$

Compute $\delta_r{}^{[0]}$

$$\delta_r{}^{[0]} = \frac{\sum_{i<j} w_{ij}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} w_{ij} d_{ij}{}^2} = \frac{2.6}{256.0} = 0.1$$

(i,j)	d_{ij}	$\hat{d}_{ij}$	$d_{ij}{}^2$	$(d_{ij} - \hat{d}_{ij})^2$
2,3	4.1	3.17	16.8	0.9
1,3	2.2	3.17	4.8	0.9
1,2	5.1	5.1	26.0	0
2,4	8.5	7.9	72.3	0.4
1,4	7.3	7.9	53.3	0.4
3,4	9.1	9.1	82.8	0
$\sum$			256.0	2.6

Compute the first update $X^{[1]}$by the Guttman transform

$$\boldsymbol{X}^{[1]} = n^{-1}\mathrm{B}\left(\mathbf{X}^{[0]}\right)\mathbf{X}^{[0]}$$

$$= \frac{1}{4}\begin{bmatrix} 3.52 & -1 & -1.44 & -1.08 \\ -1 & 2.702 & -0.773 & -0.929 \\ -1.44 & -0.773 & 3.213 & -1 \\ -1.08 & -0.929 & -1 & 2.009 \end{bmatrix}\begin{bmatrix} 3 & 2 \\ 2 & 7 \\ 1 & 3 \\ 10 & 4 \end{bmatrix}$$

$$\boldsymbol{X}^{[1]} = \begin{bmatrix} -3.460 & -3.910 \\ 0.408 & -0.851 \\ -3.163 & -7.143 \\ 3.498 & -5.767 \end{bmatrix}$$

The elements of the D ($\boldsymbol{X}^{[1]}$) are given by $d_{ij} = \sqrt{(x_i - x_j)^t (x_i - x_j)}$

$$d_{12} = \sqrt{(-3.460 - 0.408)^2 + (-3.910 - -0.851)^2} = 4.931$$

$$d_{13} = \sqrt{(-3.460 - -3.163)^2 + (-3.910 + 7.143)^2} = 3.073$$

$$d_{14} = \sqrt{(-3.460 - 3.498)^2 + (-3.910 + 5.767)^2} = 11.919$$

$$d_{23} = \sqrt{(0.408 - -3.163)^2 + (-0.851 + 7.143)^2} = 7.234$$

$$d_{24} = \sqrt{(0.408 - 3.498)^2 + (-0.851 + 5.767)^2} = 5.806$$

$$d_{34} = \sqrt{(-3.163 - 3.498)^2 + (-7.143 + 5.767)^2} = 6.802$$

$$\text{D}(\boldsymbol{X}^{[1]}) = \begin{bmatrix} 0.000 & 4.931 & 3.073 & 11.919 \\ 4.931 & 0.000 & 7.234 & 5.806 \\ 3.073 & 7.234 & 0.000 & 6.802 \\ 11.919 & 5.806 & 6.802 & 0.000 \end{bmatrix}$$

Compute disparities $\hat{d}_{ij}$ for D ($\boldsymbol{X}^{[1]}$

- *The first distance* 7.234 exceeds the *second* distance 3.073. To correct this relation, each of these two distances is replaced with their mean5.154. But, *the first distance and second distance* exceed *the third distance* 4.931. To correct this relation, each of these

distances is replaced with their mean Thus, *the first distance, second and the third distance* have been replaced by disparities 5.079.

- *The forth distance* 5.806 does not exceed the *fifth* distance 11.919; so, these distances fit the monotonic relation established by the proximity matrix.
- *The fifth distance* 11.919 exceeds the *sixth distance* 6.802. To correct this relation, each of these two distances is replaced with their mean. Thus, *the fifth distance and sixth distance* have been replaced by disparities 9.3605.

(i,j)	δ_{ij}	d_{ij}	$\hat{d}_{ij}$
2,3	1	7.234	5.079
1,3	2	3.073	5.079
1,2	3	4.931	5.079
2,4	4	5.806	5.806
1,4	5	11.919	9.3605
3,4	6	6.802	9.3605

Compute B($\boldsymbol{X}^{[1]}$)

The elements of the B$\left(\boldsymbol{X}^{[1]}\right)$ are given by $b_{ij} = \frac{w_{ij}\hat{d}_{ij}}{d_{ij}}$.

We assume that all w_{ij} = 1.

$b_{12} = -w_{12}\hat{d}_{12}/d_{12}(\mathrm{X^u}) = -5.1/5.1 = -1$

$b_{13} = -w_{13}\hat{d}_{13}/d_{13}(\mathrm{X}^{\mathrm{u}}) = -3.17/2.2 = -1.44$

$$b_{14} = -w_{14}\hat{d}_{14}/d_{14}(\mathrm{X}^{\mathrm{u}}) = -7.9/7.3 = -1.08$$

$b_{11} = -(b_{12} + b_{13} + b_{14}) = -(-1 - 1.44 - 1.08) = 3.52$

$b_{23} = -w_{23}\hat{d}_{23}/d_{23}(\mathrm{X}^{\mathrm{u}}) = -3.17/4.1 = -0.773$

$$b_{24} = -w_{24}\hat{d}_{24}/d_{24}(z\mathrm{X}^{\mathrm{u}}) = 7.9/8.5 = -0.929$$

$$b_{22} = -(b_{21} + b_{23} + b_{24}) = -(-0.929 - 0.773 - 1) = 2.702$$

$b_{34} = -w_{34}\hat{d}_{34}/d_{34}(\mathrm{X}^{\mathrm{u}}) = -9.1/9.1 = -1$

$b_{33} = -(b_{31} + b_{32} + b_{34}) = -(-1.44 - 0.773 - 1) = 3.213$

$$b_{44} = -(b_{41} + b_{42} + b_{43}) = -(-1.08 - 0.929 - 1) = 2.009$$

$$\mathrm{B}(\boldsymbol{X}^{[1]}) \begin{bmatrix} 3.52 & -1 & -1.44 & -1.08 \\ -1 & 2.702 & -0.773 & -0.929 \\ -1.44 & -0.773 & 3.213 & -1 \\ -1.08 & -0.929 & -1 & 2.009 \end{bmatrix}$$

Compute δ_r $\boldsymbol{X}^{[1]}$

$$\delta_r{}^{[0]} = \frac{\sum_{i<j} w_{ij}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j} w_{ij} d_{ij}{}^2} = \frac{13.808}{308.129} = 0.045$$

(i,j)	d_{ij}	$\hat{d}_{ij}$	${d_{ij}}^2$	$(d_{ij}-\hat{d}_{ij})^2$
2,3	7.234	5.079	52.331	4.644
1,3	3.073	5.079	9.443	4.024
1,2	4.931	5.079	24.315	0.022
2,4	5.806	5.806	33.710	0
1,4	11.919	9.3605	142.063	2.559
3,4	6.802	9.3605	46.267	2.559
$\sum$			308.129	13.808

The second update is

$$\boldsymbol{X}^{[2]} = n^{-1}\mathrm{B}\left(\boldsymbol{X}^{[1]}\right)\boldsymbol{X}^{[1]}$$

$$\boldsymbol{X}^{[2]} =$$

$$\frac{\mathbf{1}}{\mathbf{4}}\begin{bmatrix} 3.52 & -1 & -1.44 & -1.08 \\ -1 & 2.702 & -0.773 & -0.929 \\ -1.44 & -0.773 & 3.213 & -1 \\ -1.08 & -0.929 & -1 & 2.009 \end{bmatrix}\begin{bmatrix} -3.460 & -3.910 \\ 0.408 & -0.851 \\ -3.163 & -7.143 \\ 3.498 & -5.767 \end{bmatrix}$$

$$\boldsymbol{X}^{[2]} = \begin{bmatrix} -8.127 & -2.471 \\ 7.007 & 7.132 \\ -5.496 & -16.662 \\ 4.916 & 5.459 \end{bmatrix}$$

Continue the iterations until the difference in subsequent Stress values is less than 10^{-6}.

3.3.2.1 Judging the goodness of fit

The amount of stress may also be used for judging the goodness of fit of an MDS solution: a small stress value indicates a good fitting solution, whereas a high value indicates a bad fit. **Kruskal (1964a)** provided some guidelines for the interpretation of the stress value with respect to the goodness of fit of the solution (Table 3.4).

Caution: These simple guidelines are easily misused. In order to avoid misinterpretation, the following should be kept in mind:

- There are many different stress formulae in the MDS literature. The guidelines, however, apply only to the stress measure computed by equation (3-4).
- Stress decreases as the number of dimensions increases. Thus, a two-dimensional solution always has more stress than a three-dimensional one.

Stress	Goodness of fit
> .20	poor
0.10	fair
0.05	good
0.025	excellent
0.00	perfect

Table 3.4: Stress and goodness of fit.

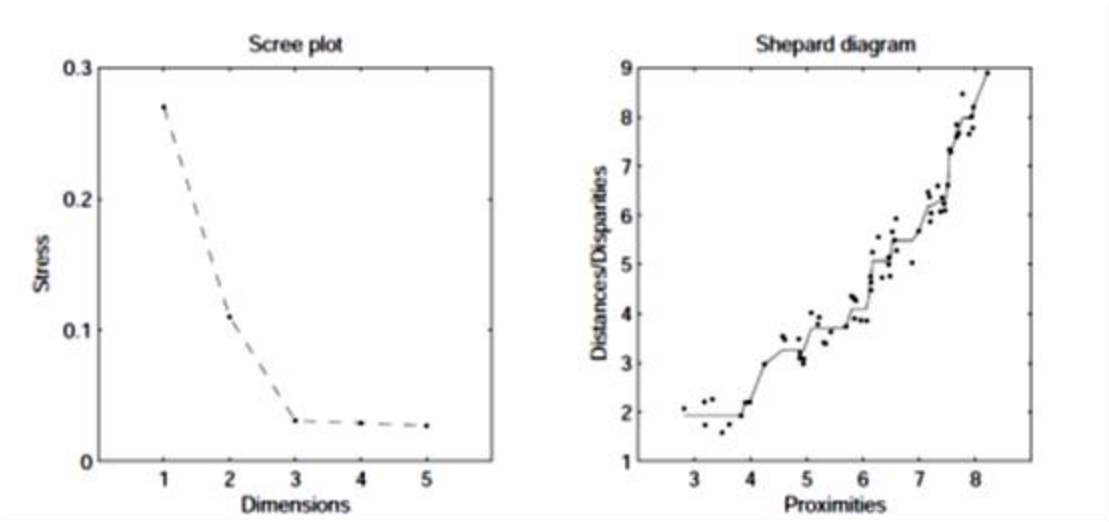
The Shepard diagram and scree plot can be used to describe the badness-of-fit.

The Shepard diagram

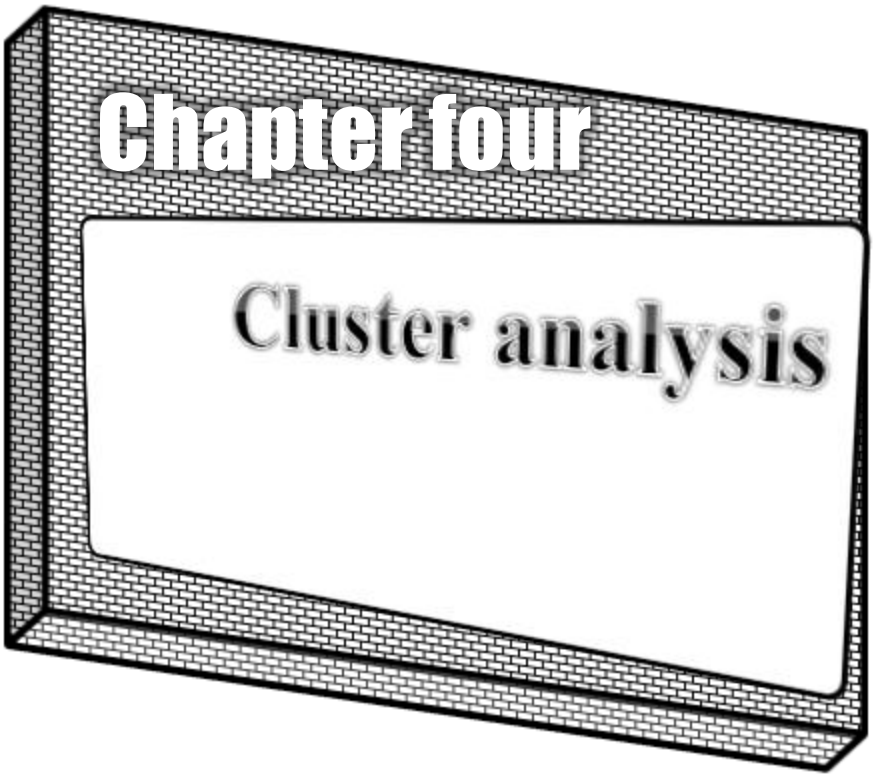
The Shepard diagram plots the disparities $\hat{d}_{ij}$ and distances d_{ij} on the same graph, which gives an indication of how well the disparities are fitted to the distances. The (d_{ij} , $\hat{d}_{ij}$) pairs are plotted and these pairs all lie on a monotonically increasing regression line. The (d_{ij} , $\hat{d}_{ij}$) pairs are also plotted. The vertical distance between these points gives a measure of the corresponding error $e_{ij} = (d_{ij} - \hat{d}_{ij})$. The error gives an indication of the size of mismatch between the distances and the corresponding disparities. Larger errors will cause a higher Normalized Stress (3-7) and Raw stress (3-6) value.

Scree plot

In a scree plot, the amount of stress is plotted against the number of dimensions. Since stress decreases monotonically with increasing dimensionality, one is looking for the lowest number of dimensions with acceptable stress. An "elbow" in the scree plot indicates, that more dimensions would yield only a minor improvement in terms of stress. Thus, the best fitting MDS model has as many dimensions as the number of dimensions at the elbow in the scree plot**. (Borg and Groenen, 2005).**



Left panel: Scree plot displaying an elbow at three dimensions.
Right panel: Shepard diagram with the optimally scaled proximities.


Chapter four
Cluster analysis

IV.CLUSTER ANALYSIS

4.1 Introduction

Cluster analysis is an area of statistics that involves sorting observed data into natural groupings based on similarity. Grouping data is important because it can reveal a lot of information about the data such as outliers, dimensionality, or interesting relationships that may have previously gone unnoticed. Many think of clustering methods much like classification; however, there are important differences. In classification, there is some pre-specified number of groups or categories into which variables or data are placed. There are also specific rules for placing items into each category, depending on the method of grouping the data. Unlike classification, in cluster analysis there is no prior specification about the number of groups or types of groups to which different variables or data points will be assigned. The grouping is done based solely on similarity measures and the number of groups that seems to suit the data best is often determined within the clustering algorithm. These characteristics can make cluster analysis difficult. The groupings really depend on the definition of similarity.

4.2 Cluster Analysis

The word *clustering* is defined as: *"a grouping of a number of similar things"* [**University (2006)].**Here, the word *similar* refers to the objects present in the same group, which possess like characteristics. In data mining, the goal of cluster analysis methods is to cluster unlabeled data, with no or little prior information about the class labels, into groups, such that objects in the same subgroup are very similar to one another and objects in two different subgroups are very different [**Witten and Frank (2005**)] **Han& Kamber (2006**)] [**Dunham (2002**]. Let *D* be a dataset with n objects. When a cluster analysis algorithm is applied to this dataset

D, it groups the data in $C_1, C_2, \ldots C_k$ clusters given that the total number of clusters is k.

The main objective of a cluster analysis method is to minimize the distance between the objects located in the same cluster and to maximize the distance between the objects located in different clusters. Figure 4.1 (a) depicts a sample dataset in a 2-dimensional space and Figure 4.1 (b) shows the clusters marked with circles when k = 3. The results after applying a cluster analysis algorithm show that the clusters are generated in such a way so that the objects in each cluster are very close to one another. However, in the real world, the datasets are not as simple as the one depicted above. The objects are not always so clearly separated and the clusters are not usually as well-defined. Moreover, the datasets may contain hundreds or even thousands of objects and the feature space of these objects may also be very high dimensional. As a result, the task of clustering is often more complex and challenging.

(a) (b)

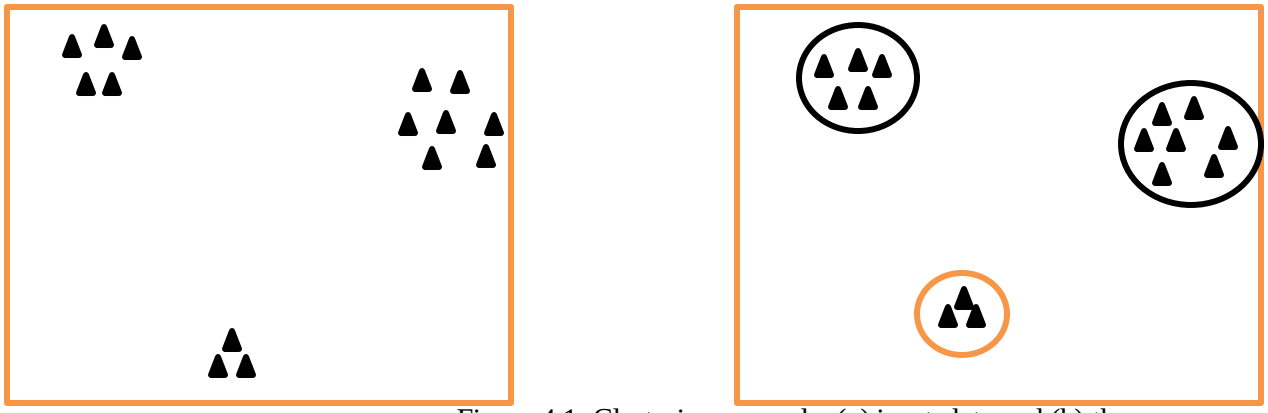


Figure 4.1: Clustering example: (a) input data and (b) the clusters.

In the following subsection we discuss the fundamental steps of a typical cluster analysis task.

4.2.1 Cluster Analysis Procedure

Cluster analysis methods usually follow a number of sequential steps **[Jain & etal (1999)], [Xu & Wunsch (2005)].** Figure 4.2 illustrates the basic steps of a cluster analysis procedure as discussed in **[Xu & Wunsch (2005)]**. According them, the four main steps that most clustering algorithms follow are:

a) Feature selection or feature extraction.

b) Design or selection of cluster analysis algorithm.

c) Cluster validation.

d) Interpretation of results.

We briefly discuss each of the four components below.

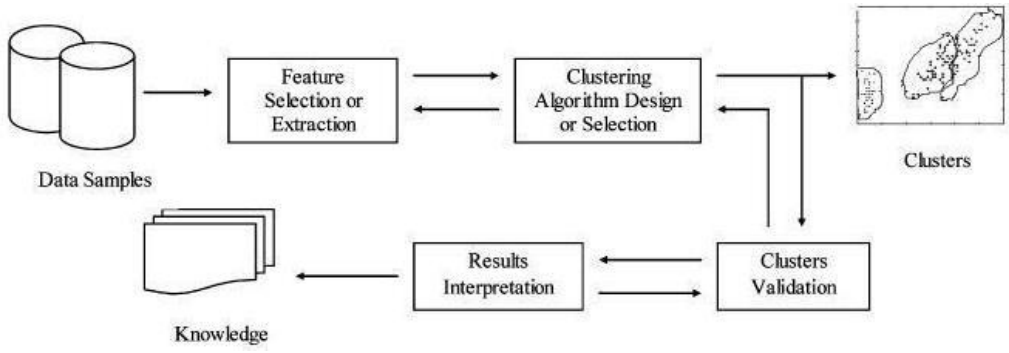


Figure 4.2: Sequential procedure of a cluster analysis process **[Xu and Wunch (2005)**].

a-Feature Selection or Extraction: In practical applications, datasets often contain a large number of features to represent the objects. However, not all the features are useful for the learning process. Most of the time, there are several features.

The experimental studies of **Witten et al., (2005)** show that, adding such features to the cluster analysis process usually deteriorates the performance of the algorithms. As such, techniques such as *Feature Selection* and *Feature Extraction* often prove to be useful to carefully reduce the size of

the original feature set. According to Jain et al. **[Jain et al (1999)]** *Feature Selection* is the process of identifying the most effective subset of features from the original feature set. In contrast, *Feature Extraction* is the process of producing a new set of features by performing transformations on the original feature **set [Xu and Wunsch (2005)], [Jain et al (1999)].** Both of the processes reduce the feature size by removing the redundant or irrelevant features and in doing so, simplify the clustering process.

b-Design or Selection of Cluster Analysis Algorithm:

This step involves the selection of a proximity measure and a cluster analysis algorithm. The selection of a proximity measure directly affects the formation of the clusters. One of the commonly used distance measures is the *Euclidean distance* measure. There are, however, a number of other proximity measures available in the literature which we discussed in detail in Chapter 2. In addition to the selection of a proximity measure, the results from cluster analysis also vary depending on the clustering algorithm that has been selected **[Jain et al (1999)].** Several algorithms partition the data into a predefined number of groups (i.e. K-means), whereas other algorithms output a nested series of clusters **[Jain et al (1999)].** Some of the algorithms are suitable for large datasets, whereas other methods handle outliers better. We discuss various cluster analysis methods in Section 4.3.

c-Cluster Validation:

Given a dataset, a cluster analysis algorithm will always produce proximity functions may produce different results. Therefore, it is necessary to assess the results to compare, evaluate, and measure the goodness of the cluster analysis methods. There are several evaluation and validation measures proposed in the literature that help to perform such an assessment. According to Jain et al. **[Jain et al (1999)]** and. [**Xu et**

al(2005)], these cluster validation measures are categorized into three groups:

1) *External measures.*

2) *Internal measures.*

3) *Relative measures.*

The external measures consider the prior knowledge about the data (i.e. class labels) against the cluster analysis results for the assessments.

In contrast, *the internal measures* compute the assessment without any reference to the external information; they only consider the information present in the original dataset.

The relative measures perform the evaluation by comparing the results from various cluster analysis methods with one another.

d-Interpretation of Results:

The ultimate goal of any cluster analysis task is to partition the data into *meaningful* groups. As such, in this step, domain experts often analyze the clusters to discover the hidden patterns among the objects in a cluster and to assign a label to the clusters based on the underlying patterns.

4.2.2 Limitations:

A number of application domains to which the cluster analysis algorithms are often applied. The areas include data mining, machine learning, pattern recognition, bioinformatics, image processing, and many others. Nevertheless, when the cluster analysis techniques are applied to real-world datasets, several problems arise. In this section, we briefly state the drawbacks of cluster analysis as addressed by Dunham in **[Dunham (2002)].**

• One of the main difficulties that arise with respect to a cluster analysis task is to correctly and automatically determine the number of

clusters *k*. In cluster analysis, most of the time the prior knowledge or additional information about the data is not available to the users. As such, the algorithms that require the number of clusters A; as input need special consideration. Intuitively, providing an incorrect value for *k* may result in unsatisfactory results. For instance, selecting a smaller value for *k* may over-generalize the results as it will try to combine natural clusters to achieve the user-specified number of clusters. In contrast, if *k* is set to a very high value it may decompose the natural clusters into many smaller subsets to achieve the desired number of clusters. Both the cases will have significant impact on the results.

• Interpreting the clustering results or more specifically, interpreting the clusters, is also considered to be one of the major problems in cluster analysis. As class labels are not available during the process, it may not always be possible to correctly interpret the semantic meaning of each of the individual clusters without any domain-specific knowledge.

• Handling outliers is another fundamental problem in cluster analysis. In a dataset, outliers are objects that are very different from the other objects in the dataset, and as such, they usually form their own clusters. Placing an outlier in a cluster that contains objects that are very different from it (i.e. to achieve the desired number of clusters), may result in the formation of poor clusters [**Dunham (2002)].**

• Because dynamic data change over time, cluster membership may also change over time and therefore requires careful consideration to accommodate the changes.

• Another problem that may be encountered during the cluster analysis process, is that there may be no exact or correct answer to the clustering solution. Given a dataset, different algorithms may return different sets of clusters. Moreover, different users may also have different views and therefore may interpret the clusters differently. These

difficulties may make the decision making task more complex and ambiguous.

• Finally, with the increasing amount of data, problems surrounding high dimensionality and handling of large datasets have also become a point of concern. However, these problems also open the door to new research ideas. Various algorithms have been proposed to solve one or more of these problems efficiently. In the next section, we provide an overview of the cluster analysis methods and briefly address their advantages and disadvantages.

4.3 Overview of Cluster Analysis Methods

There have been many cluster analysis algorithms proposed in the literature. A number of these algorithms are particularly suitable for a certain type of data (e.g. numeric or nominal). Several algorithms are also suitable for a particular purpose or the application domain **[Han and Kamber (2006)], [Kaufman and Rousseeuw (2005)].** We briefly present several cluster analysis methods as discussed in **[Dunham (2002)]** and **[Han and Kamber (2006)].** We place particular emphasis on the first two methods, partitional and hierarchical, as they are strongly related to this study.

4.3.1 Partitional Methods:

A partitioning method creates k partitions, called clusters, from given set of n data objects. Initially, each data objects are assigned to some of the partitions. An iterative relocation technique is used to improve the partitioning by moving objects from one group to another. Here, each partition is represented by either a centroid or a medoid.

A centroid is an average of all data objects in a partition, while *the medoid* is the most representative point of a cluster **[Velmurugan,T. and Santhanam,T.,(2011)**]. The fundamental requirements of the partitioning based methods are each cluster must contain at least one data object, and each data objects must belong to exactly one cluster. In this category of clustering, various methods have been developed.

4.3.1.1 K-means Algorithm:

K-means is an iterative algorithm where a cluster is represented by the *centroids* (the mean value of the objects in a cluster).**McQueen (1967)** proposed the K-means cluster method **[kandil (2011)].**

Given a dataset and the number of clusters *k*, the algorithm works as follows **[kandil (2011)]** the first step of this algorithm is to initialize the centroids. There are a number of different ways to assign the initial values to the centroids. We may either randomly select any *k* objects from the data, or select the first *k* objects and assign them as the centroids of the clusters. Once the algorithm is initialized with the centroids, the next step is to calculate the distance from each centroid to all the objects in the dataset. A distance measure, such as the Euclidean distance, is often used to calculate this distance. Next, the objects are assigned to the respective clusters based on the minimum distance from the centroids. Therefore, an object will be assigned to a cluster if the distance between its centroid and the object is minimum (compared to the distances between the centroids of other clusters and this object). Once all the objects are assigned to their respective clusters, we recalculate the centroids with the new cluster assignments. The centroid, as mentioned above, is the mean value of all the objects in a cluster. We then iterate the process a number of times until the stopping criterion is satisfied. This is usually satisfied when the objects are no longer reallocated to different clusters or when the maximum number of iterations is reached.

The K- means ***Algorithm:***

Input:

Input : 'k', the number of clusters to be partitioned; 'n', the number of objects.

Output:

A set of 'k' clusters based on given similarity function

Algorithm:

1. Arbitrarily choose 'k' objects as the initial cluster centers;
2. Repeat,

a. (Re)assign each object to the cluster to which the object is the most similar; based on the given similarity function;

b. Update the centroid (cluster means), i.e., calculate the mean value of the objects for each cluster;

3. until no change.

Example4.3.1.1 . In this example, the dataset contains 9 items:

D = {2, 4, 10, 12, 3, 20, 30, 11, 25}. Let k = 2, the desired number of clusters. We use the Euclidean distance as the distance measure. The first step of the algorithm consists in assigning any two items as the cluster centroids. These items are either selected randomly or the first k items are selected. We used the later approach for this example. Below we show the calculations for each phase.

Iteration 1: centroid1 = 2 and centroid 2 = 4

The distance between centroid l and each item in

D:{0,2,8,10,1,18,28,9,23}

The distance between centroid 2 and each item in

D:{2,0,6,8,1,16,26,7,21}

According to the minimum distance between the centroids and each of the items, the clusters are:

Cluster1 = {2, 3} Since the item 3 is equally close to centroid l and centroid 2, we arbitrarily selected cluster 1.

Cluster2 = {4, 10, 12, 20, 30, 11, 25}

Iteration 2: centroid l = $\frac{2+3}{2}$= 2.5 and

centroid2 = $\frac{4+10+12+20+30+11+25}{7}$= 16

The distance between centroid l and each item in D:

{0.5, 1.5, 7.5, 9.5, 0.5, 17.5, 27.5, 8.5, 22.5}

The distance between centroid2 and each item in D:

{14, 12, 6, 4, 13, 4, 14, 5, 9}

According to the minimum distance between the centroids and each of the items, the clusters are:

Cluster 1 = {2, 3, 4} Since the item 4 is equally close to centroid l and *Cluster 2* = {10, 12, 20, 30, 11, 25}

Iteration 3: centroid l = $\frac{2+3+4}{3}$ = 3 and

centroid 2 = $\frac{10+12+20+30+11+25}{5}$= l8

The distance between centroid l and each item in D:

{1, 1, 7, 9, 0, 17, 27, 8, 22}

The distance between centroid2 and each item in D:

{16, 14, 8, 6, 15, 2, 12, 7, 7}

According to the minimum distance between the centroids and each of the items, the clusters are:

Cluster 1 = {2, 3, 4, 10} Since the item 10 is equally close to centroid l and *Cluster2* = {12, 20, 30,11,25}

Iteration 4: centroid l = $\frac{2+3+4+10}{4}$= 4.75

and centroid 2 =$\frac{12+20+30+11+25}{5}$ = 19.6

The distance between centroid l and each item in D:

{2.75, 0.75, 5.25, 7.25, 1.75, 15.25, 25.25, 6.75, 20.25} The distance between centroid2 and each item in D:

{17.6, 15.6, 9.6, 7.6, 16.6, 0.4, 11.4, 8.6, 5.4}

Cluster 1 = {2, 3, 4, 10, 11} Since the item 11 is equally close to centroid l

and *Cluster2* = {12, 20, 30,25}

Iteration 5: *centroid l* = $\frac{2+3+4+10+11}{6}$ = *5*

and centroid 2 = $\frac{12+20+30+25}{3} = 21.75$

The distance between centroid l and each item in D:

{4, 24, 6, 3, 14, 24, 5, 19}

The distance between centroid2 and each item in D:

{19.75, 17.75, 11.75, 9.75, 1.75, 8.25, 10.75, 3.25}

The clusters are:

Cluster l = {2, 3, 4, 10, 11, 12} Since the item 12 is equally close to centroid l

and Cluster 2 = {20, 30, 25}

Iteration 6: centroid l = $\frac{2+3+4+10+11+12}{6}$ = 7

and centroid 2 = $\frac{20+30+25}{3} = 25$

The distance between centroid l and each item in D:

{5, 3, 3, 5, 4, 13, 23, 6, 18}

The distance between centroid2 and each item in D:

{23, 21, 15, 13, 22, 5, 5, 14 , 0}

The clusters are:

Cluster l = {2, 3, 4, 10, 11, 12} and *Cluster 2* = {20, 30, 25}

We stop at this step because none of the items were relocated in iteration 6 (iteration 5 and 6 are identical)

The result for this example, which is returned at the end of the process is: *Cluster l* = {2, 3, 4, 10, 11, 12} and *Cluster2* = {20, 30, 25}.

Advantages of the K-means Algorithm:

1.According to Han et al. **[Han and Kamber (2006)],** the K-means algorithm works well for compact clusters in which the clusters are well separated from one another.

2.Moreover, the algorithm also works well for large datasets, since the computational complexity of the algorithm is *O(n)*, where *n* is the number of objects present in the dataset **[Jain et al (1999)], [Han and Kamber (2006)].**

Limitations of the K-means Algorithm

1- One of the disadvantages of the K-means algorithm is that it only considers numeric attribute types and is therefore not applicable to datasets with nominal or categorical attributes.

2-The performance of the K-means algorithm depends in part on the initial values selected as the cluster centroids in the initialization stage that may later affect the quality of the clusters.

3-Dunham [**Han and Kamber (2006)]** also states that, the K-means algorithm is very sensitive to outliers.

4-Not suitable to discover clusters with non-convex shape, or clusters of very different size. **[Aiello et al (2007)].**

4.3.1.2 K-medoids method:

The most well-known K-medoids algorithms are *PAM* (Partitioning Around Medoids) **[Kaufman and Rousseeuw (2005)].** The k-means method uses centroid to represent the cluster and it is sensitive to outliers. This means, a data object with an extremely large value may disrupt the

distribution of data. K-medoids method overcomes this problem by using medoids to represent the cluster rather than centroid. A medoid is the most centrally located data object in a cluster **[Berkhin (2002)].**

Here k data objects are selected randomly as medoids to represent k cluster and remaining all data objects are placed in a cluster having medoid nearest (or most similar) to that data object. After processing all data objects, new medoid is determined which can represent cluster in a better way and the entire process is repeated. Again all data objects are bound to the clusters based on the new medoids. In each iteration, medoids change their location step by step. Or in other words, medoids move in each iteration. This process is continued until no any medoid move. As a result, k clusters are found representing a set of n data objects. An algorithm for this method is given below.**[Han& Kamber (2006)].**

The K- medoids ***Algorithm***

Input:

Input : 'k', the number of clusters to be partitioned; 'n', the number of objects.

Output:

A set of 'k' clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid.

Algorithm:

1. Arbitrarily choose 'k' objects as the initial medoids;
2. Repeat,

a. Assign each remaining object to the cluster with the nearest medoid;

b. Randomly select a non-medoid object;

c. Compute the total cost of swapping old medoid object.

d. If the total cost of swapping is less than zero, then perform that swap operation to form the new set of k- medoids.

3. Until no change.

The strengths and weaknesses of this algorithm are mentioned as below.

Strengths:

More robust than k-means in the presence of noise and outliers; because a medoid is less influenced by outliers or other extreme values than a mean.

Weaknesses:

1. Relatively more costly
2. Relatively not so much efficient.
3. Need to specify k, the total number of clusters in advance.
4. Result and total run time depends upon initial partition

4.3.2 Hierarchical Methods

In this section, we discuss another type of cluster analysis method known as the Hierarchical Clustering methods. A hierarchical method builds a hierarchy or a tree of clusters.

The tree is also commonly referred to as a dendrogram **[Dunham (2002)].** The root of a tree often contains all the data objects in one cluster, whereas the leaves of the tree usually contain each object in a single cluster. There are two variations of this method discussed in the literature: agglomerative or bottom-up approach and divisive or top-down approach **[Han and Kamber (2006)], [Jain et al (1999`)], [kandil (2011)].**

4.3.2.1The agglomerative (bottom-up)

In this approach, the algorithm starts from the bottom of the tree where each object has its own unique cluster. It gradually groups these clusters by recursively merging two or more similar clusters together. This process

is continued until all the clusters are merged into a single cluster (the root) or a given termination criterion is satisfied.

4.3.2.1.1The steps for the agglomerative hierarchical clustering algorithm

Given a proximity matrix $D_{n\times n}$= $[d_{rs}]$, the steps for the agglomerative hierarchical clustering algorithm are as follows.

1. Begin with *n* clusters, each containing only a single object.
2. Search the dissimilarity matrix **D** for the most similar pair. Let the pair chosen be associated with element d_{rs}so that object *r* and *s* are selected.
3. Combine objects *r* and *s* into a new cluster *(rs)* employing some criterion and reduce the number of clusters by deleting the row and column for objects *r* and *s*. Calculate the dissimilarities between the cluster *(rs)* and all remaining clusters, using the criterion, and add the row and column to the new dissimilarity matrix.
4. Repeat steps 2 and 3, $(n-1)$ times until all objects form a single cluster. At each step, identify the merged clusters and the value of the dissimilarity at which the clusters are merged.

By changing the criterion in Step 3 above, we obtain several agglomerative hierarchical clustering methods.

Agglomerative hierarchical clustering methods:

4.3.2.1.1.Single Link (Nearest-Neighbor) Method

This method also has been referred to as the elementary linkage, minimum method, and nearest neighbor cluster analysis **(Johnson, 1967; Lance and Williams, 1967).**

Seeath (1957) and McQuitty (1957) proposed the signal link in which merges groups based on the minimum distance between the closest points

between two groups. Letting r represent any element in cluster R, $r \in R$, and s be any element in cluster S, $s \in S$, from the clusters in Step 3 of the agglomerative clustering algorithm, distances between R and S are calculated using the rule:

$$d_{(R)(S)} = \min\{d_{rs}\} \quad , r \in R \; and \; s \in S\} \qquad (4.3.1)$$

Example4.3.2.1.1.there are 5 samples ($x_1 = 2$), ($x_2 = 11$), ($x_3 = 0$), ($x_4 = 6$), and ($x_5 = -4$). Each sample represents one cluster and the distance matrix D is

$$D = d_{rs} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 9 & 2 & 4 & 6 \\ 9 & - & 11 & 5 & 15 \\ 2 & 11 & - & 6 & 4 \\ 4 & 5 & 6 & - & 10 \\ 6 & 15 & 4 & 10 & - \end{bmatrix} \end{array} .$$

Merge x_1 and x_3 ($d_{\min} = d(x_1, x_3) = 2$), and the distance matrix D is updated as follows:

$d_{(x_1x_3)(x_2)} = \min\{ d_{x_1x_2}, d_{x_3x_2}\} = \min\{9,11\} = 9$

$d_{(x_1x_3)(x_4)} = \min\{ d_{x_1x_4}, d_{x_3x_4}\} = \min\{4,6\} = 4$

$d_{(x_1x_3)(x_5)} = \min\{ d_{x_1x_5}, d_{x_3x_5}\} = \min\{6,4\} = 4$

$$\begin{array}{c} \\ \{x_1, x_3\} \\ x_2 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccc} \{x_1, x_3\} & x_2 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 9 & 4 & 4 \\ 9 & - & 5 & 15 \\ 4 & 5 & - & 10 \\ 4 & 15 & 10 & - \end{bmatrix} \end{array}$$

Merge $\{x_1, x_3\}$ and x_4 ($d_{\min} = d(\{x_1, x_3\}, x_4) = 4$), and the distance matrix is updated as follows:

$$\begin{array}{c} \\ \{x_1, x_3, x_4\} \\ x_2 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccc} \{x_1, x_3, x_4\} & x_2 & x_5 \end{array} \\ \begin{bmatrix} - & 5 & 4 \\ 5 & - & 15 \\ 4 & 15 & - \end{bmatrix} \end{array}$$

Merge $\{x_1, x_3, x_4\}$ and x_5 ($d_{\min} = d(\{x_1, x_3, x_4\}, x_5) = 4$), and the distance matrix is updated as follows:

$$\begin{array}{c} \\ \{x_1, x_3, x_4, x_5\} \\ x_2 \end{array} \begin{array}{c} \begin{array}{cc} \{x_1, x_3, x_4, x_5\} & x_2 \end{array} \\ \begin{bmatrix} - & 5 \\ 5 & - \end{bmatrix} \end{array}$$

Finally, all samples are merged into one cluster $\{x_1, x_2, x_3, x_4, x_5\}$ (see Figure 4.3)

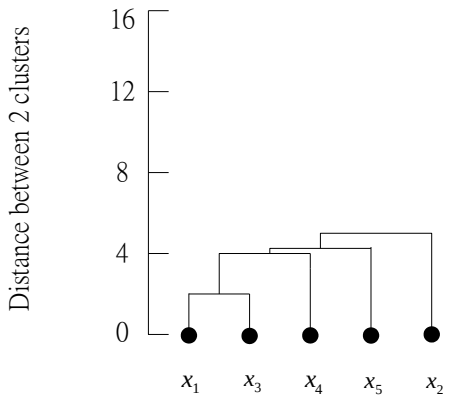


Figure 4.3. An example for using single-linkage algorithm

4.3.2.1.2.Complete Link (Farthest-Neighbor) Method

A second agglomerative method, referred to as complete linkage analysis, maximum method, or furthest neighbor analysis. (**Horn, 1943**) proposed Complete Link Method**.** In the single link method, dissimilarities were replaced using minimum values. For the complete link procedure, maximum values are calculated instead. Letting $r \in R$ and $s \in S$, where R

and *S* are two clusters, distances between clusters *R* and *S* are calculated using the rule

$$d_{(R)(S)} = \max \{d_{rs},\ r \in R \text{ and } s \in S\} \qquad (4.3.2)$$

Example 4.3.2.1.2 To illustrate rule (4.3.2.), we consider the same dissimilarity matrix discussed in Example 4.3.2.1.1

$$\mathrm{D} = d_{rs} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 9 & 2 & 4 & 6 \\ 9 & - & 11 & 5 & 15 \\ 2 & 11 & - & 6 & 4 \\ 4 & 5 & 6 & - & 10 \\ 6 & 15 & 4 & 10 & - \end{bmatrix} \end{array}.$$

1. Merge x_1 and x_3 ($d_{\min} = d(x_1, x_3) = 2$) *represents the most similar objects. Using (4.3.2.), we replace minimum values with maximum values*

$d_{(x_1x_3)(x_2)} = \max \{ d_{x_1x_2}, d_{x_3x_2}\} = \max \{9,11\} = 11$

$d_{(x_1x_3)(x_4)} = \max \{ d_{x_1x_4}, d_{x_3x_4}\} = \max \{4,6\} = 6$

$d_{(x_1x_3)(x_5)} = \max \{ d_{x_1x_5}, d_{x_3x_5}\} = \max \{6,4\} = 6$

so that the new dissimilarity matrix is

$$\mathrm{D}_1 = d_{rs} = \begin{array}{c} \\ \{x_1, x_3\} \\ x_2 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccc} \{x_1, x_3\} & x_2 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 11 & 6 & 6 \\ 11 & - & 5 & 15 \\ 6 & 5 & - & 10 \\ 6 & 15 & 10 & - \end{bmatrix} \end{array}$$

2.Merge x_2 and x_4 ($d_{\min} = d(x_2, x_4) = 5$), and the distance matrix is updated as follows:

$$\begin{array}{c} \\ \{x_1, x_3\} \\ \{x_2, x_4\} \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccc} \{x_1, x_3\} & \{x_2, x_4\} & x_5 \end{array} \\ \begin{bmatrix} - & 11 & 6 \\ 11 & - & 15 \\ 6 & 15 & - \end{bmatrix} \end{array}$$

3.Merge $\{x_1, x_3\}$ and x_5 ($d_{\min} = d(\{x_1, x_3\}, x_5) = 6$), and the distance matrix is updated as follows:

$$\begin{array}{c} \\ \{x_1, x_3, x_5\} \\ \{x_2, x_4\} \end{array} \begin{array}{c} \begin{array}{cc} \{x_1, x_3, x_5\} & \{x_2, x_4\} \end{array} \\ \begin{bmatrix} - & 15 \\ 15 & - \end{bmatrix} \end{array}$$

4. Finally, all samples are merged into one cluster $\{x_1, x_2, x_3, x_4, x_5\}$ (see Figure 4.4).

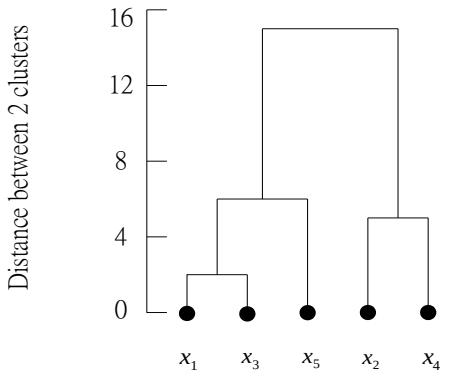


Figure 4.4. An example for using complete-linkage algorithm

4.3.2.1.3.Average Link Method

In the average link method, the distance between two clusters is defined as an average of dissimilarity measures. **Sokal and Michener (1958)** proposed the average linkage cluster method. When comparing two clusters of objects *R* and *S*, the single link and complete link methods of combining clusters depend only upon a single pair of objects within each cluster. Instead of using a minimum or maximum measure, the

average link method calculates the distance between two clusters using the average of the dissimilarities in each cluster **[Kandil (2011)].**

$$d_{(R)(S)} = \frac{\sum_r \sum_s d_{rs}}{n_R n_S} \qquad 4.3.3$$

where $r \in R, s \in S$, and n_R and n_S represent the number of objects in each cluster. Hence, the dissimilarities in Step 3 are replaced by an average of $n_R n_S$ dissimilarities between all pairs of elements $r \in R$ and $s \in S$.

Example 4.3.2.1.3. To illustrate rule (4.3.3.), we consider the same dissimilarity matrix discussed in Example 4.3.2.

$$\mathrm{D}= d_{rs}= \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 9 & 2 & 4 & 6 \\ 9 & - & 11 & 5 & 15 \\ 2 & 11 & - & 6 & 4 \\ 4 & 5 & 6 & - & 10 \\ 6 & 15 & 4 & 10 & - \end{bmatrix} \end{array} .$$

1- Merge x_1 and x_3 ($d_{\min} = d(x_1, x_3) = 2$), and the distance matrix D is updated as follows:

$d_{(x_1x_3)(x_2)=}$ aver $\{ d_{x_1x_2}, d_{x_3x_2}\}$=aver {9,11}=10

$d_{(x_1x_3)(x_4)=}$ aver $\{ d_{x_1x_4}, d_{x_3x_4}\}$=aver {4,6}=5

$d_{(x_1x_3)(x_5)=}$ aver $\{ d_{x_1x_5}, d_{x_3x_5}\}$=aver {6,4}=5

so that the new dissimilarity matrix is

$$\begin{array}{c} \\ \{x_1, x_3\} \\ x_2 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cccc} \{x_1, x_3\} & x_2 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 10 & 5 & 5 \\ 10 & - & 5 & 15 \\ 5 & 5 & - & 10 \\ 5 & 15 & 10 & - \end{bmatrix} \end{array}$$

2-Merge x_2 and x_4 ($d_{\min} = d(x_2, x_4) = 5$), and the distance matrix is updated as follows:

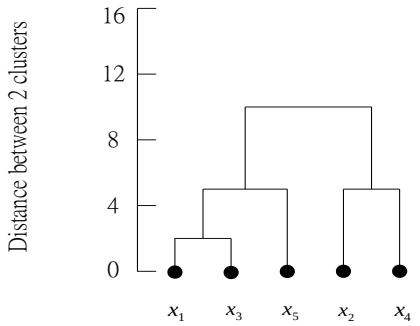
$$\begin{array}{c} \\ \{x_1, x_3\} \\ \{x_2, x_4\} \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccc} \{x_1, x_3\} & \{x_2, x_4\} & x_5 \end{array} \\ \begin{bmatrix} - & 7.5 & 5 \\ 7.5 & - & 12.5 \\ 5 & 12.5 & - \end{bmatrix} \end{array}$$

3-Merge $\{x_1, x_3\}$ and x_5 ($d_{\min} = d(\{x_1, x_3\}, x_5) = 5$), and the distance matrix is updated as follows:

$$\begin{array}{c} \\ \{x_1, x_3, x_5\} \\ \{x_2, x_4\} \end{array} \begin{array}{c} \begin{array}{cc} \{x_1, x_3, x_5\} & \{x_2, x_4\} \end{array} \\ \begin{bmatrix} - & 10 \\ 10 & - \end{bmatrix} \end{array}$$

4-Finally, all samples are merged into one cluster $\{x_1, x_2, x_3, x_4, x_5\}$ (see Figure 4.5).

Figure 4. 5. An example for using average-linkage algorithm



4.3.2.1.4 Centroid Method:

In the *centroid* method, distance is defined as the distance between group centroids. In this method, the distance between two clusters R and S is defined as the Euclidean distance between the mean vectors (often called centroids) of the two clusters:

$$d_{(R)(S)} = d_{\bar{y}_r \bar{y}_s} = \|\bar{y}_r - \bar{y}_s\|^2 \qquad (4.3.4)$$

Where

(i) $\bar{y}_R$ is the mean vectors for the observation vectors in R,

(ii) $\bar{y}_S$ is the mean vectors for the observation vectors in S

$$\bar{y}_R = \frac{\sum_r y_r}{n_r} = \begin{bmatrix} \bar{y}_{r1} \\ \bar{y}_{r2} \\ . \\ . \\ \bar{y}_{rp} \end{bmatrix} \quad \text{and} \quad \bar{y}_S = \frac{\sum_s y_s}{n_s} = \begin{bmatrix} \bar{y}_{s1} \\ \bar{y}_{s2} \\ . \\ . \\ \bar{y}_{sp} \end{bmatrix} \qquad (4.3.5)$$

The two clusters with the smallest distance between centroids are merged at each step. After two clusters R and S are joined, the centroid of the new cluster RS is given by the weighted average

$$\bar{y}_{RS} = \frac{n_R \bar{y}_R + n_S \bar{y}_S}{n_R + n_S} \qquad (4.3.6)$$

4.3.2.1.5. Ward's Method:

Ward's method, also called the *incremental sum of squares method*, uses the within cluster (squared) distances and the between-cluster (squared) distances (**Ward 1963**, Wishart 1969a). If RS is the cluster obtained by combining clusters R and S, then the sum of within-cluster distances (of the items from the cluster mean vectors) are:

$$\text{SSE}_\text{R} = \sum_{i=1}^{n_R} (y_i - \bar{y}_R)' (y_i - \bar{y}_R) \qquad (4.3.7)$$

$$\text{SSE}_\text{S} = \sum_{i=1}^{n_S} (y_i - \bar{y}_S)' (y_i - \bar{y}_S) \qquad (4.3.8)$$

$$\text{SSE}_\text{RS} = \sum_{i=1}^{n_{RS}} (y_i - \bar{y}_{RS})' (y_i - \bar{y}_{RS}) \qquad (4.3.9)$$

Where

(i) $n_{rs} = n_r + n_s$,

(ii) $\bar{y}_{RS} = \frac{n_R \bar{y}_R + n_S \bar{y}_S}{n_R + n_S}$,

(iii) n_r is the numbers of points in R

(iv) n_s is the numbers of points in S

(v) n_{rs} is the numbers of points in RS

Since these sums of distances are equivalent to within-cluster sums of squares, they are denoted by SSE_R , SSE_S, and SSE_{RS}.

Ward's method joins the two clusters R and S that minimize the increase in SSE, defined as

$$I_{RS} = SSE_{RS} - (SSE_R + SSE_S) \qquad (4.3.10)$$

It can be shown that the increase I_{RS} in (4.3.10) has the following two equivalent forms:

$$I_{RS} = n_R(\bar{y}_R - \bar{y}_{RS})'(\bar{y}_R - \bar{y}_{RS}) + n_s(\bar{y}_S - \bar{y}_{RS})'(\bar{y}_S - \bar{y}_{RS}) \qquad (4.3.11)$$

$$= \frac{n_R n_S}{n_R + n_S}(\bar{y}_R - \bar{y}_S)'(\bar{y}_R - \bar{y}_S) \qquad (4.3.12)$$

Where

$$SSE_{RS} = \sum_{i=1}^{n_{RS}}(y_i - \bar{y}_{RS})'(y_i - \bar{y}_{RS})$$

$$SSE_{RS} = \sum_{i=1}^{n_{RS}} y_i' y_i - \sum_{i=1}^{n_{RS}} y_i' \bar{y}_{RS} - \sum_{i=1}^{n_{RS}} \bar{y}'_{RS} y_i + \sum_{i=1}^{n_{RS}} \bar{y}'_{RS} \bar{y}_{RS}$$

Since $\bar{y}_{RS} = \frac{\sum_{i=1}^{n_{RS}} y_i}{n_{RS}}$

$$SSE_{RS} = \sum_{i=1}^{n_{RS}} y_i' y_i - n_{RS}\bar{y}'_{RS}\bar{y}_{RS} - n_{RS}\bar{y}'_{RS}\bar{y}_{RS} + n_{RS}\bar{y}'_{RS}\bar{y}_{RS}$$

$$SSE_{RS} = \sum_{i=1}^{n_{RS}} y_i' y_i - n_{RS}\bar{y}'_{RS}\bar{y}_{RS}.$$

Similarly

$$SSE_R = \sum_{i=1}^{n_R} y_i' y_i - n_R\bar{y}'_R\bar{y}_R.$$

$$SSE_S = \sum_{i=1}^{n_S} y_i' y_i - n_S\bar{y}'_S\bar{y}_S.$$

Thus

$$I_{RS} = SSE_{RS} - (SSE_R + SSE_S)$$

$$I_{RS} = \sum_{i=1}^{n_{RS}} y_i' y_i - n_{RS}\bar{y}'_{RS}\bar{y}_{RS} - \sum_{i=1}^{n_R} y_i' y_i + n_R\bar{y}'_R\bar{y}_R - \sum_{i=1}^{n_S} y_i' y_i + n_S\bar{y}'_S\bar{y}_S$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S - n_{RS}\bar{y}'_{RS}\bar{y}_{RS}$$

Now we Show that when the right side of (4.3.11) is expanded, it reduces to this same expression

$$I_{RS} = n_R(\bar{y}_R - \bar{y}_{RS})'(\bar{y}_R - \bar{y}_{RS}) + n_s(\bar{y}_S - \bar{y}_{RS})'(\bar{y}_S - \bar{y}_{RS})$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R - n_R\bar{y}'_R\bar{y}_{RS} - n_R\bar{y}'_{RS}\bar{y}_R + n_R\bar{y}'_{RS}\bar{y}_{RS} + n_S\bar{y}'_S\bar{y}_S - n_S\bar{y}'_S\bar{y}_{RS} - n_S\bar{y}'_{RS}\bar{y}_R + n_S\bar{y}'_{RS}\bar{y}_{RS}$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S - 2(n_R\bar{y}'_R + n_S\bar{y}'_S)\,\bar{y}_{RS} + (n_R + n_S)\bar{y}'_{RS}\bar{y}_{RS}$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S - 2(n_R + n_S)\bar{y}'_{RS}\,\bar{y}_{RS} + (n_R + n_S)\bar{y}'_{RS}\bar{y}_{RS}$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S - (n_R + n_S)\bar{y}'_{RS}\,\bar{y}_{RS}$$

Substitute $\bar{y}_{RS} = \frac{n_R\bar{y}_R + n_S\bar{y}_S}{n_R + n_S}$

$$(n_R + n_S)\bar{y}'_{RS}\,\bar{y}_{RS} = (n_R + n_S)\,\frac{(n_R\bar{y}_R + n_S\bar{y}_S)'}{(n_R + n_S)}\,\frac{(n_R\bar{y}_R + n_S\bar{y}_S)}{(n_R + n_S)}$$

$$= \frac{n^2{}_R\bar{y}'_R\,\bar{y}_R + n_Rn_S\bar{y}'_R\,\bar{y}_S + n_Rn_S\bar{y}'_S\,\bar{y}_R + n^2{}_S\bar{y}'_S\,\bar{y}_S}{n_R + n_S} \qquad \text{(I)}$$

Multiplying $n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S$ by $\frac{n_R + n_S}{n_R + n_S}$

$$\frac{n_R + n_S}{n_R + n_S}n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S = \frac{n^2{}_R\bar{y}'_R\,\bar{y}_R + n_Sn_R\bar{y}'_R\bar{y}_R + n_Rn_S\bar{y}'_S\bar{y}_S + n^2{}_S\bar{y}'_S\,\bar{y}_S}{n_R + n_S} \text{ (II)}$$

$$I_{RS} = n_R\bar{y}'_R\bar{y}_R + n_S\bar{y}'_S\bar{y}_S - (n_R + n_S)\bar{y}'_{RS}\,\bar{y}_{RS}$$

From I and II

$I_{RS}=$

$$\frac{n^2{}_R\bar{y}'_R\,\bar{y}_R+n_Sn_R\bar{y}'_R\bar{y}_R+n_Rn_S\bar{y}'_S\bar{y}_S+n^2{}_S\bar{y}'_S\,\bar{y}_S-n^2{}_R\bar{y}'_R\,\bar{y}_R-n_Rn_S\bar{y}'_R\,\bar{y}_S-n_Rn_S\bar{y}'_S\,\bar{y}_R-n^2{}_S\bar{y}'_S\,\bar{y}_S}{n_R+n_S}$$

$$I_{RS}=\frac{1}{n_R+n_S}n_Sn_R\bar{y}'_R\bar{y}_R+n_Rn_S\bar{y}'_S\bar{y}_S+-n_Rn_S\bar{y}'_R\,\bar{y}_S-n_Rn_S\bar{y}'_S\,\bar{y}_R$$

$$I_{RS}=\frac{n_Sn_R}{n_R+n_S}(\bar{y}_R\text{-}\bar{y}_S)'(\bar{y}_R\text{-}\bar{y}_S) \qquad (4.3.12),$$

Thus by (4.3.12), minimizing the increase in SSE is equivalent to minimizing the *between-cluster* distances.

If R consists only of y_i ($n_R=1,\bar{y}_R=y_i$) and S consists only of y_i ($n_S=1,\bar{y}_S=y_j$), then

SSE_R and SSE_S are zero, and (4.3.10) and (4.3.12) reduce to

$$L_{RS}=SSE_{Rs}=\frac{1\times1}{1+1}(y_i-y_j)'(y_i-y_j)=\frac{1}{2}d^2(y_i,y_j)$$

$$L_{RS}=SSE_{Rs}=\tfrac{1}{2}(y_i-y_j)'(y_i-y_j)=\tfrac{1}{2}d^2(y_i,y_j) \qquad (4.3.12)$$

4.3.2.2The divisive hierarchical (top-down):

Divisive algorithms begin with just only one cluster that contains all sample data. Then, the single cluster splits into 2 or more clusters that have higher dissimilarity between them until the number of clusters becomes number of samples or as specified by the user. **[Kandil (2011)].** Two clusters are merged when the distance is low and a cluster is split into smaller clusters when the distance is large (when the elements are not close enough).The following algorithm is one kind of divisive algorithms using splinter party method.

Divisive algorithm using splinter party method [Hui-Chuan (2009)]:

1. Start with just only one cluster. That is, all samples in this one cluster.
2. Repeat step 3, 4, 5, 6 until cluster number is the number of samples or what we want.
3. Calculate diameter of each cluster. Diameter is the maximal distance between samples in the cluster. Choose one cluster R having maximal diameter of all clusters to split.
4. Find the most dissimilar sample from cluster R. Let depart from the original cluster R to form a new independent cluster S (now cluster R doesn't include sample). Assign all members of cluster R to M_R.
5. Repeat 6 until members of cluster R and S don't change.
6. Calculate similarities from each member of M_R to cluster R and S, and let the member owning the highest similarities in M_R move to its similar cluster R or S. Update members of R and S.

Exampel 4.3.2.2 we take a simple example to describe the method above. First, the distance matrix D of 5 samples x_1, x_2, x_3, x_4, x_5 is

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} - & 2 & 6 & 10 & 9 \\ 2 & - & 5 & 9 & 8 \\ 6 & 5 & - & 4 & 5 \\ 10 & 9 & 4 & - & 3 \\ 9 & 8 & 5 & 3 & - \end{bmatrix} \end{array} .$$

Our processing steps are as follows:

1- Because there is only one cluster, this cluster has maximal diameter. For a start, we split this cluster.

2-Calculate average distances from one sample to the others. For example, the average distance from x_1 to x_2, x_3, x_4 and x_5 is $(2+6+10+9)/4=6.75$, and the others:

$x_2:(2+5+9+8)/4=6,$
$x_3:(6+5+4+5)/4=5,$
$x_4:(10+9+4+3)/4=6.5,$
$x_5:(9+8+5+3)/4=6.25.$

Sample x_1 has maximal average distance, so extract x_1 from the cluster. Now we have 2 clusters: $\{x_2,x_3,x_4,x_5\}$ and $\{x_1\}$.

1-Find average distances from x_2, x_3, x_4 and x_5 to clusters $\{x_2,x_3,x_4,x_5\}$ and $\{x_1\}$.

	$\{x_2,x_3,x_4,x_5\}$	$\{x_1\}$
x_2	7.33	2
x_3	4.67	6
x_4	5.33	10
x_5	5.33	9

The distance from x_2 to cluster $\{x_1\}$ is minimum, so put x_2 into cluster $\{x_1\}$. Now clusters are updated to $\{x_3,x_4,x_5\}$ and $\{x_1,x_2\}$. Repeat step 6 of the algorithm to check if members of each cluster are updated.

	$\{x_3,x_4,x_5\}$	$\{x_1,x_2\}$
x_2	7.33	2
x_3	4.5	5.5
x_4	3.5	9.8
x_5	4	8.5

The distance from x_2 to cluster $\{x_1,x_2\}$ is also minimum and cluster members don't change again. Go to step 3 of the algorithm. Now there are 2 clusters $\{x_1,x_2\}$ and $\{x_3,x_4,x_5\}$.

The diameter of the cluster $\{x_1, x_2\}$ is:

$diameter(\{x_1, x_2\}) = \max(|x_1 - x_2|) = 2.$

The diameter of cluster $\{x_3, x_4, x_5\}$ is

$diameter(\{x_3, x_4, x_5\}) = \max(|x_3 - x_4|, |x_3 - x_5|, |x_4 - x_5|) = 5.$

1-We choose the cluster $\{x_3, x_4, x_5\}$ to split (has maximal diameter of all clusters). Calculate average distances from one sample to the others in cluster $\{x_3, x_4, x_5\}$.

$x_3 : (4+5)/2 = 4.5$
$x_4 : (4+3)/2 = 3.5$
$x_5 : (5+3)/2 = 4$

So split $\{x_3, x_4, x_5\}$ into $\{x_3\}$ and $\{x_4, x_5\}$. The average distances from x_4 and x_5 to clusters $\{x_4, x_5\}$ and $\{x_3\}$ are:

$$\begin{array}{c} \\ x_4 \\ x_5 \end{array} \begin{array}{c} \{x_4, x_5\} \quad x_3 \\ \begin{bmatrix} 3 & 4 \\ 3 & 5 \end{bmatrix} \end{array}$$

Because minimum distance is 3, cluster members of each cluster don't update. Go to step 3 of the algorithm.

1-Now we have 3 clusters $\{x_1, x_2\}$, $\{x_3\}$, and $\{x_4, x_5\}$. Their diameters are 2, 0, and 3. Because there is only one sample in cluster $\{x_3\}$, don't think about this cluster. We decide split the cluster $\{x_1, x_2\}$.

2-Split $\{x_1, x_2\}$ into $\{x_1\}$ and $\{x_2\}$. Because cluster members of each cluster don't update, go to step 3.

3-Now we have 4 clusters $\{x_1\}, \{x_2\}$, $\{x_3\}$ and $\{x_4, x_5\}$. Only the cluster $\{x_4, x_5\}$ has more than one sample and have maximal diameter, so split $\{x_4, x_5\}$

4-Split $\{x_4, x_5\}$ into $\{x_4\}$ and $\{x_5\}$. Each sample represents one cluster, so stop (see Figure 4.6).

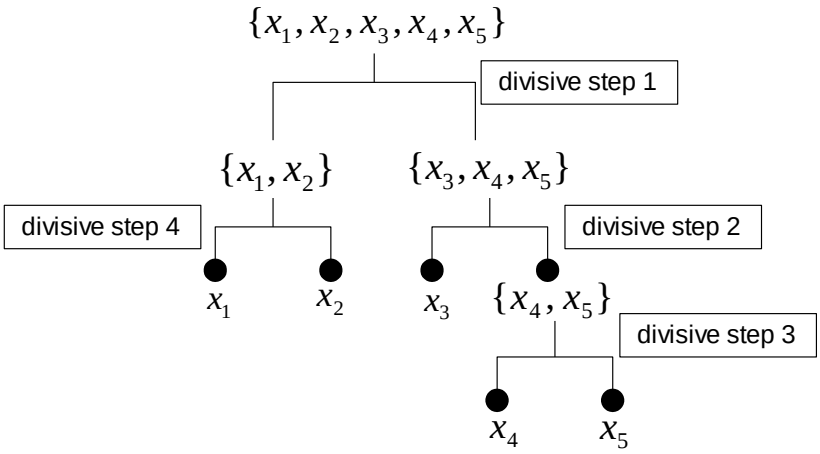


Figure 4.6. An example for hierarchical divisive algorithm

Advantages of the Hierarchical Clustering Methods:

(i) Hierarchical methods are suitable for datasets that possess natural nesting relationships between the clusters. Examples of such datasets include datasets from biology and animal taxonomies **[Dunham (2002)].**

(ii) Moreover, since the distance or similarity is presented through a matrix to these algorithms, the algorithms are able to handle different attribute types **[Berkhin (2002)].**

Limitations of the Hierarchical Clustering Methods:

(i) One of the weaknesses of the hierarchical methods is that, once a cluster is formed, the objects in the clusters may not be relocated to improve the results. As such, unlike the K-means algorithm where

objects are iteratively relocated to improve the result, the hierarchical algorithms lack such possibility.

(ii) The algorithms are also sensitive to outliers **[Xu and Wunsch (2005)]. Dunham (2002)** also noted that, due to the time and space complexity of these algorithms, they may not be suitable for large datasets.

4.3.3 Density-based Methods:

Unlike the hierarchical and partitional cluster analysis algorithms, which consider the distance or similarity between the objects to find the clusters, *density-based* methods are based on the notion of *density*. According to Dunham **[Dunham (2002)],** the term *density* is defined as the minimum number of objects located within a certain distance of one another. Thus, the clusters are represented by the dense areas of the data objects and are usually separated by the areas with low density. In this approach, the clusters may take any arbitrary shape and grow in any direction, as long as the density in the neighboring area exceeds a certain threshold **[Han and Kamber (2006)].** Examples of algorithms from this family are: *DBSCAN* (Density-Based Spatial Clustering Algorithm with Noise) **[Ester and Xu (1996)]** and *DENCLUE* (DENsitybased CLUstEring) **[Laflin (1998)].** As the name implies, the DBSCAN algorithm is suitable for spatial datasets with noise. The algorithm also discovers clusters of arbitrary shape **[Han and Kamber (2006)].**

However, this algorithm is very sensitive to the choice of user-defined parameters (e.g. the radius of the neighborhood) **[Han and Kamber (2006)].** The DENCLUE algorithm is suitable for high dimensional datasets. Similar to the DBSCAN algorithm, this algorithm also discovers arbitrary shaped clusters and handles datasets with large amount of noise **[Han and Kamber (2006)].**

4.3.4 Grid-based Methods:

In the Grid-based cluster analysis **[Han and Kamber (2006)]** methods, the entire data space is first divided into a finite number of cells that form a grid structure. The cluster analysis is then performed on this grid data, instead of the original data points. Since the number of cells in the grid data is usually much less than the number of original data points, the computation and processing time of this algorithm are relatively faster than many other cluster analysis algorithms. The algorithms from this family are mostly suitable for spatial datasets. *STING* (STatistical INformation Grid) **[Wang et al (1997)],** *WaveCluster* **[Sheikholeslami et al (1998)],** and *CLIQUE* **[Agrawal et al (1998)]** are an example of algorithms based on this method. The STING algorithm manipulates the statistical information (e.g. count, maximum, minimum, and standard deviation) of the grid cells to process the queries. The algorithm is query-independent as the statistical information regarding the attributes are pre-computed and stored in each cell. STING is also very efficient. Moreover, when a given dataset is updated, this algorithm is able to perform incremental updates without re-computing all the statistical information **[Han and Kamber (2006)].** However, the user-specific parameters (e.g. the number of grids and number of layers) need to be provided by the users and therefore the selection of parameters may have impact on the end result. The WaveCluster algorithm, in contrast, applies a signal processing technique called wavelet transform, to find the clusters. More information regarding wavelet transform and WaveCluster are presented in **[Sheikholeslami et al (1998)], [Han and Kamber (2006)].** The algorithm is not sensitive to outliers, discovers clusters of arbitrary shapes, and performs well for large datasets. However, one of the drawbacks of this algorithm is that it may only be applied to low-dimensional datasets. On

the other hand, the CLIQUE algorithm, which integrates density-based and grid-based algorithms together, is suitable for large, highly dimensional datasets.

4.3.5 Model-based Methods:

Model-based approaches assume that all the data is generated by a mixture of underlying statistical distributions. For example, the *EM* (Expectation-Maximization) algorithm is a popular model-based approach that performs expectation-maximization analysis based on statistical modeling **[Han and Kamber (2006)].** The *COBWEB* and *SOM* (Self-Organized Map) algorithms also fall into this category, where the former is a conceptual learning algorithm and the latter is a neural network-based algorithm. A detailed discussion of these algorithms is presented in **[Han and Kamber (2006)].**

4.3.6 Clustering High Dimensional Data:

Highly dimensional datasets consist of several hundreds or even thousands of attributes. For instance, objects in a text dataset are usually regarded as a collection of documents and each document consists of hundreds or even thousands of words and terms. Thus, the attributes for this type of datasets are the collection of these words and terms gathered from the documents. In such cases, the previously discussed clustering algorithms may not work well as the data become very sparse with the increase of the number of dimensions. As a result, when the similarity between the data points is calculated, the result is usually a very small value which may not contribute to the computation. Moreover, as **Han and Kamber (2006)** noted, the average density of these points is also likely to be very low. Therefore, new or modified algorithms that handle the problem of high dimensionality are necessary. Two such methods for

clustering high-dimensional datasets are Subspace Clustering and Frequent Pattern-based Clustering. The subspace clustering algorithms such as *CLIQUE* and *PROCLUS*, tend to find the clusters from a subset of dimensions of the original set of attributes. On the other hand, the frequent pattern-based clustering algorithms search for frequently occurring patterns from the dataset and use these patterns to find the clusters **[Han and Kamber (2006)].** With a growing number of domains containing high dimensional data, performing cluster analysis on highly dimensional datasets has become challenging. Therefore, special care is needed to successfully perform cluster analysis on this type of datasets.

4.3.7 Constraint-based Clustering

The *Constraint-based* methods consist of cluster analysis algorithms that heavily rely on user guidance. Users provide various constraints and information to the algorithms so that the clusters may be generated based on the preferences given by the users. **Yin et al. (2005)** proposed one such user-guided clustering algorithm called *CrossClus*. The algorithm is suitable for multi-relational datasets. The algorithm starts with selecting a set of relevant features from multiple relations to construct a single object type, based on the user interest and domain specific knowledge. Next, the K-medoids based algorithm, *CLARANS* is applied to the selected features to find the clusters.

Chapter Five

Application Study and Results

V. Application Study and Results

5.1 Overview of the study

Multidimensional scaling and cluster analysis are widely used in marketing research for positioning of different brands of the companies. It would be desired and beneficial for any company to know how its brand of products is rated among public when compared with other similar competing brands. **(Verma (2013)**.

Multidimensional scaling can create a visual presentation of the subjective dimensions that are not directly shown in the data. By showing these objects visually on a map, it will be easier for public to associate close together objects as similar or close in terms of preference.

Cluster analysis can be used to segregate all the brands of certain product into some clusters, that assist the companies in identifying their current location within the market and who their closest rivals are. This helps the companies to pay attention and focus on their marketing activities of their brands in the same cluster and try to modify it to make it much better.

5.2 Study Data.

The data was collected by some questionnaires which were distributed among different car exhibitions found in the city of Banha, the sample size was a 20 customers. The owners of the car exhibitions were asked to give these questionnaires to their customers. The set of cars used in the questionnaires and in our experiment is presented in Table 5.1.

The Twenty customers were asked to rate the 10 cars by showing the cards bearing the name of a pair of cars. All possible pair of cars were shown, and the customers were asked to rate their preferences of one car over the

other on a scale of 100 points. If the customer perceived that the two cars were completely dissimilar, a score of 0 was given, and if the two cars were exactly similar, a score of 100 was given.

	Object
1	Kia Cerato
2	Chevrolet Aveo
3	Renault Fluence
4	Toyota Corolla
5	Mitsubishi Lancer
6	Geely Emgrand
7	Hyundai Elantra
8	Speranza Tiggo
9	Nissan Sunny
10	Peugeot 208

Table 5.1 Cars' object set

After obtaining the similarity matrix for each consumer, the average of these similarities was calculated for each pair of objects to make the final similarity matrix (the input data).

To summarize, the data produced from the experiment are consisting of:
A collection of 20 *proximity matrices*, one for each consumer. Each proximity matrix is a 10 x 10 symmetric matrix in which cell s_{ij} contains the numerical value of the similarity between cars *i* and *j* as judged by that customer. Only one similarity was obtained for each object pair from each customer. These data are presented in Appendix A.
An *average* similarity *matrix* over all subjects was obtained by averaging the similarity for each object pair over all subjects. This matrix (presented in Table 5.2) is used as SPSS input data.

	kia	Chevrolet	Renault	Toyota	Mitsubishi	Geely	Hyundai	Speranza	Nissan	Peugeot
kia	100.0	34.8	79.2	86.0	76.3	63.3	57.9	62.5	65.6	26.0
Chevrolet	34.8	100.0	54.4	56.0	30.5	40.7	86.0	80.7	23.6	60.9
Renault	79.2	54.4	100.0	70.5	51.2	37.8	77.7	71.6	69.4	70.0
Toyota	86.0	56.0	70.5	100.0	66.3	90.0	50.1	88.6	6.3	89.4
Mitsubishi	76.3	30.5	51.2	66.3	100.0	35.4	76.0	67.5	22.6	63.1
Geely	63.3	40.7	37.8	90.0	35.4	100.0	77.1	54.1	35.1	67.9
Hyundai	57.9	86.0	77.7	50.1	76.0	77.1	100.0	66.1	76.8	59.3
Speranza	62.5	80.7	71.6	88.6	67.5	45.1	66.1	100.0	71.3	33.6
Nissan	65.6	23.6	69.4	66.2	22.6	35.1	76.8	71.3	100.0	59.3
Peugeot	26.0	60.0	70.0	89.4	63.1	67.9	59.3	33.6	59.3	100.0

Table 5.2 Average similarity Matrix for Cars

Using Green, Carmone and Smiths (1989) recommendations, since there are 10 brands 2 dimensions are most appropriate.

The data file was prepared before using SPSS to generate the outputs in multidimensional scaling. The data was exported directly into the output window of SPSS. In the data file the ten variables were defined as ordinal because the scores were representing the dissimilarity ratings. After defining the variable names and their labels, the command sequence (Analyze – Scale - Multidimensional Scaling) was selected on the SPSS program. In Model tab, two dimensional solution was investigated along with the stress value as 0.0367. These two dimensions were the attributes of these brands drawn through knowledge of the market based on the surveys of the customers. Thus, the two dimensions were named as follows:

Dimension 1: Stylish

Dimension 2: Problematic

In terms of the perceptual map shown in figure 1 below, the first dimension seems to correspond with Style ranging from less stylish (on the left) to more stylish (on the right) and dimension 2 seems to correspond with

problems in the car ranging from more problematic (on the top) to less problematic (on the bottom).

The opposites of car brand characteristics have been identified which seem to be linked with the dimensions. The dimensions seem to be strongly based on performance and style of the car. If the dimensions are correct then the following cars on the right side of Dimension 1 in the Euclidean Distance Model should be more stylish and the cars on the bottom of Dimension 2 are less problematic with higher performance and safety.

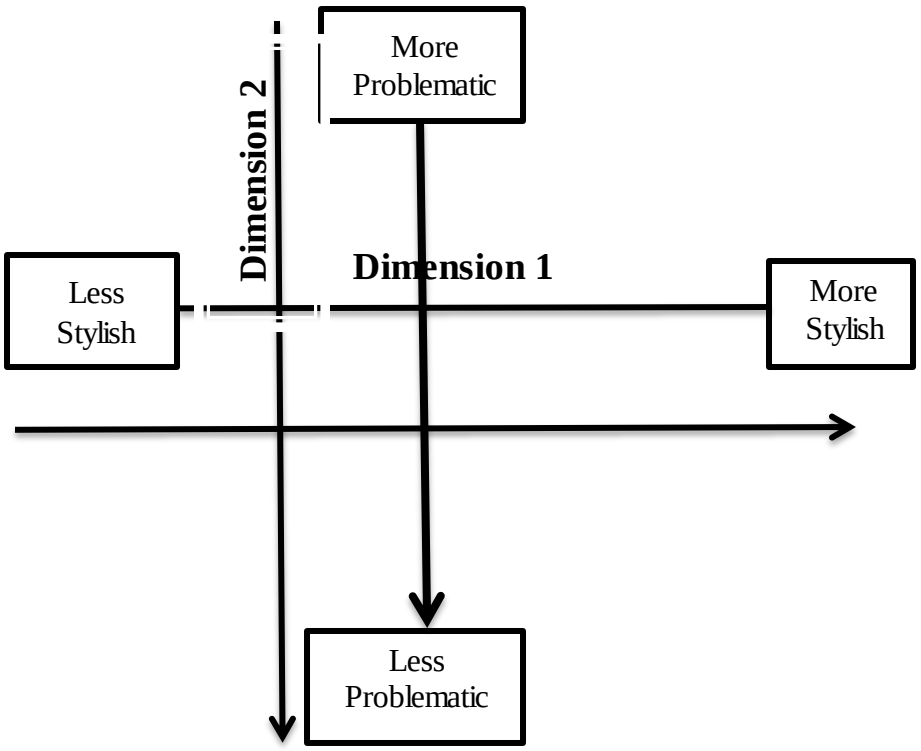


Figure 1 showing the two dimensions used in our study.

5.3 Results:

5.3.1 Basic MDS analysis of cars data

Non-metric solutions were generated using the SPSS program by use the average similarity matrix for cars as input data. The results of SPSS were:

Distance matrix for cars

	Kia	Chevrolet	Renault	Toyota	Mitsubishi	Geely	Hyundai	Speranza	Nissan	Peugeot
Kia	0									
Chevrolet	0.889	0								
Renault	1.364	0.765	0							
Toyota	1.198	0.693	1.413	0						
Mitsubishi	1.094	0.307	0.473	0.942	0					
Geely	1.004	0.681	0.413	1.373	0.52	0				
Hyundai	0.599	0.928	1.633	0.827	1.225	1.375	0			
Speranza	0.703	1.128	1.153	1.697	1.149	0.744	1.289	0		
Nissan	0.858	0.175	0.634	0.866	0.239	0.506	1.008	0.99	0	
Peugeot	0.293	0.961	1.26	1.404	1.091	0.862	0.889	0.412	0.876	0

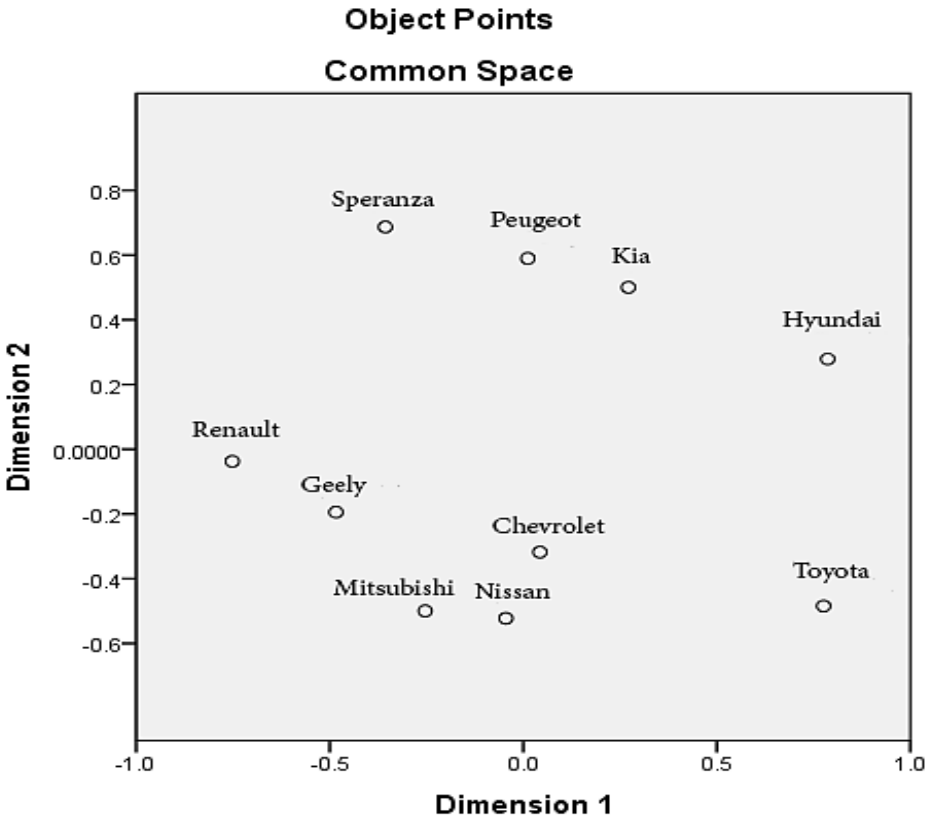
Final Stress value = 0 .0367

The MDS solution was achieved through an iterative procedure, in which an initial solution is established. Further iterations attempt to improve this solution in the context of a stress criterion.

Final coordinates for cars data in 2 dimensional :

	Dimensions	
	1	2
Kia	0.272	0.5
Chevrolet	0.043	-0.318
Renault	-0.752	-0.037
Toyota	0.776	-0.484
Mitsubishi	-0.253	-0.5
Geely	-0.484	-0.194
Hyundai	0.787	0.279
Speranza	-0.356	0.687
Nissan	-0.045	-0.522
Peugeot	0.012	0.59

2 dimensional Object Space for cars Data:



By looking to 2 dimensional Object Space for cars data, it may be concluded that the car brands like Peugeot, Speranza and Kia having more problems than other brands of similar cars while Toyota, Nissan and Mitsubishi have the lowest kind of problems. Brands like Toyota and Hyundai are similar to each other in terms of style and more stylish than the other cars.

5.3.2 Cars cluster analysis:

K-means cluster solutions were generated using the SPSS program by use the average similarity matrix for cars as input data. The results of SPSS were:

As a further way to analyze how consumers perceive the 10 cars brands in the study, Cluster analysis was used based on the stimulus coordinates to put the cars brands in clusters, this will assist the companies in identifying there current location within the market and who their closest rivals are. This may mean the brands should focus closer on the marketing activities of the brands in the same cluster.

From the Euclidean distance model it seems reasonable to identify 4 possible clusters; Cluster analysis will be used through SPSS to check if these 4 clusters are correct or if other clusters are more suitable.

The 4 cluster analysis offers a good solution and this has allowed the profile of the four following groups.

Clusters

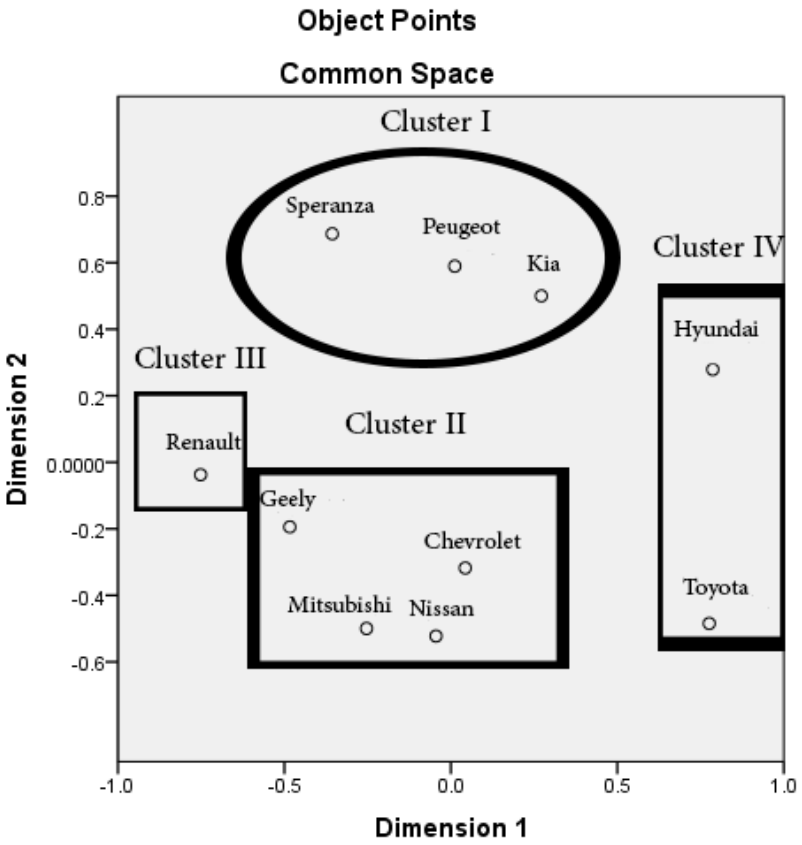
1 Kia, Speranza and peugeot

2 Geely, Chevrolet, Nissan and Mitsubishi

3 Renault

4 Hyundai and Toyota

Derived Stimulus Configuration
Euclidean distance model



Cluster Profiles:

Cluster I:

This cluster contains Kia, Speranza and peugeot. They are perceived to be more problematic, moderate in style. Peugeot and Kia are closer to each other within the group and Speranza is more independent within the cluster. It can be seen in this cluster, how Kia is making its move away from Speranza and Peugeot to be close to cluster 4.

Cluster II:

It's interesting to see that Geely has somehow differentiated itself from other cars (Chevrolet, Nissan and Mitsubishi) within the same cluster. Geely has sufficiently differentiated itself from the expected competition

of other companies by modifying the style. Out of Cluster 2 Nissan and Mitsubishi are the most differentiated, this is because they have less mechanical problems than other cars in the same cluster. Chevrolet and Nissan are in excellent position in the cluster and map as they are moderate stylish and less problematic when compared with the others.

Cluster III:

This cluster contains only Renault. It has firmly established itself as the low cost choice, this seems to be a good position, although seen as a low quality car as it is moderate problematic and less stylish than the other cars. Geely and Speranza seem to be close rivals in other clusters in terms of style. Geely in cluster II seems to be less problematic and more stylish than Renault in cluster III.

Cluster IV:

This cluster contains Hyundai and Toyota, However, it isn't very clear if these brands are in strong direct competition, Toyota seems to be in the best position among cars in terms of reliability with no major problems and more stylish than the others with no close rivals regarding the problems that can appear in the cars along with the time. Although Hyundai is sufficiently differentiated in the eyes of consumers in terms of style, they claim that the car is moderate problematic and require more maintenance.

Conclusion and Recommendations:

MDS requires caution in interpretation but can offer interesting insights into market evaluations of property attributes. Car companies will recognize that MDS provides an important step in identifying people's first impressions of different car brands, of how they 'feel'

about particular car attributes. This evaluation of attributes is important where rationality may not necessarily determine choice. The application of cluster analysis after multidimensional scaling on the same data set inspire more confidence in the accuracy of the results and provided a range of valuable marketing implications, by showing how various attributes are closely linked.

Once the preference points of the customers were identified, they could be easily plotted on a graph along with the different brands. Further investigation of the grouped preference points may lead to identification of some preference segments; this would be more interesting when compared to the clusters which the study identified. This type of study would provide a better basis to consider brand re-positioning and new product introduction since brand repositioning strategies and new product introduction strategies can address existing (known) preference segments.

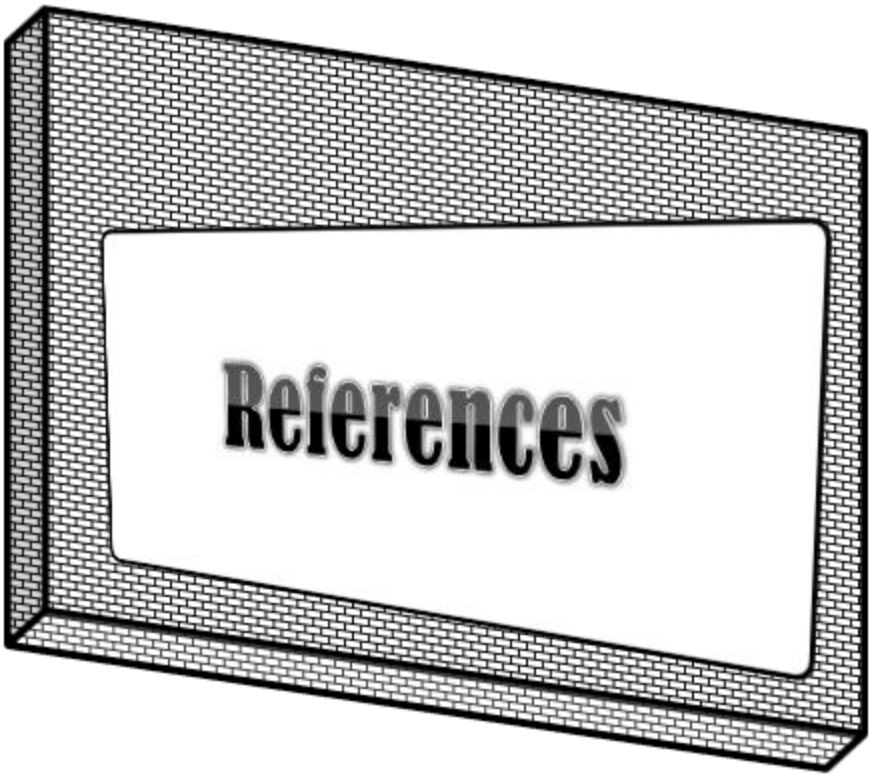
More stylish and less problematic cars are perceived to be expensive ones. Performance and Style attributes are closely linked, it seems that this is how consumers view a brand in terms of its visual appeal and this can dictate a consumer's opinion.

Although the major finding in this study, based on the multidimensional scaling solution, comes from consumer's perceptions of different car brands in two dimensions manner, it may be interesting and worthwhile to investigate three dimensional perceptual maps to gain further insights into the nature of these brands image. Also more in depth analysis could be done, possibly from grouping consumers into different socioeconomic groups to see how

these perceptions change depending on the cluster they are part of. It also could have led to some interesting multidimensional unfolding analysis, to identify ideal points for the different socioeconomic groups, this could have led to valuable marketing implications, suggesting that various car brands could be targeted to specific groups more or less, and that a specific class group could have a stronger predisposition towards a particular car brand and are more likely to be frequent shoppers than subjects from other classes.

5.5 Future Study:

1. We suggest using clustering metric multidimensional scaling to construct a better transportation network between Egypt governorates based on the distance points between them.
2. We suggest using clustering nonmetric multidimensional scaling to establish and construct some economic developmental programs for each Egyptian governorate depending on its economic and social status.
3. We suggest developing a new formulation that is intended for using multidimensional scaling in conjunction with cluster analysis in just one step.


References

REFERENCES

Agrawal, R., Gbhrke, J., Gunopulos, D. and Raghavan, P. (1998) , " Automatic subspace clustering of high dimensional data for data mining applications", ACM SIGMOD Record 27, 2 , 94 - 105.

Aiello, M., Andreozzi, F., Catanzariti, E., ISGRO, F. and Santoro,M.(2002) , " Fast convergence for spectral clustering", In ICIAP '07*:* Proceedings of the 14th International Conference on Image Analysis and Processing (Washington, DC, USA), IEEE Computer Society, pp. 641 - 646.

Anderberg, M . (1973) , " Cluster Analysis for Applications", New York: Academic.

Ando, J., **Ono, Y., and Wright, M. (2001) ,** " Genetic structure of spatial and verbal working memory", Behavioral Genetics, *31,* 615-624.

Bacher, J., Wenzig, K. and Vogler, M. (2004), " SPSS two-step cluster – a first evaluation. In RC33 sixth international conference on social science methodology: Recent developments and applications in social research methodology", Amsterdam, the Netherlands.

Beneyto, M. and Vieta, E. (2008), " Neurocognitive and clinical predictors of functional outcome in patients with schizophrenia and bipolar I disorder at one-year follow up", Journal of Affective Disorders, 109, 286-299.

Bennett,R. and Hays,W. (1960) **,** " Multidimensional unfolding: determining dimensionality of ranked preference data", Psychometrika, 25,27,43.

Berkhin, P. (2002) , " Survey of clustering data mining techniques", Technical Report, Accrue Software, San Jose, CA.

Borg,I. and Groenen,P. (2005) , " Modern Multidimensional Scaling: Theory and Applications", Springer, New York, NY, U.S.A.

Borg, I. and Groenen, P. (1997) , " Modern multidimensional scaling: theory and applications", New York: Springer.

Carroll, J. and Pruzansky, S. (1984b) , "Research methods for multimode data analysis", New York: Praeger.

Carroll, J. and Chang, J. (1969), "How to use INDSCAL, AComputer Program for Canonical Decomposition of N-Way Tables and Individual Differences in Multidimensional scaling", unpublished report, Murray Hill, NJ: Bell Telephone Laboratories.

Carroll, J. and Arabie, P. (1980) , " Multidimensional scaling", Annual Reviezu of Psychology 31,607-649.

Carroll, J. and Chang, J. (1970) , " Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition", Psychornetrika 35,283-319.

Carroll, J., Pruzansky, S. and Kruskal, J. (1980) , " CANDELINC: a general approach to multidimensional analysis of many-way arrays with linear constraints on parameters", Psychometrika, 45, 3–24.

Chang, J. and Carroll, J. (1972) , " How to use PREFMAP and PREFMAP2: Programs which relate preference data to multidimensional scaling solutions", Unpublished manuscript, Bell Telephone Laboratories, Murray Hill.

Clatworthy, J., Buick, D., Hankins, M., Weinman, J., and Home, R. (2005) , " The use and reporting of cluster analysis in health psychology: A review", British Journal of Health Psychology, 10, 329-358.

Coombs, C. (1950), " Psychological scaling without a unit of measurement", Psychological Review, 57, 145-158.

Coombs, C. (1964) , " Theory of Data", New York: Wiley, 1964

Cox,T. and Cox, M. (2001) , " Multidimensional scaling", second edition, New York. Chapman and Hall/ CRC Press. 308p.

De Leeuw, **J**., **Heiser**, **W**., **Meulman**, **J. and Critchley**, **F**. (**Eds**., **1986**) **,** " Multidimensional data analysis", Leiden: DSWO Press.

Dunham ,M. (2002) , " Data Mining: Introductory and Advanced Topics", Prentice Hall PTR, Upper Saddle River, NJ, USA.

Everitt, B. (1980), " Cluster Analysis", 2nd ed, Edward Arnold and Halsted Press.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996) , " A density-based algorithm for discovering clusters in large spatial databases with noise", In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96). pp. 226 – 231

Garten, R. , Davis, C. , Russell, C. and Smith, D. (2009) , "Antigenic and genetic characteristics of swine origin a H1N1 influenza viruses circulating in humans",Science 325(5937): 197-201.

Ghosh, J. (2003) , " Clustering. In The Handbook of Data Mining", Chapter 10, N. Ye, Ed. Lawrence Erlbaum Associates, pp. 247 - 277.

Gordon, A. (1999) , "lassification, second edition", London: Chapman and Hall/ CRC Press. 256p.

Gower, J. (1971**) ,** "A general coefficient of similarity and some of its properties", Biometrics 27: 857–871

Gower, J. (1975), "Generalized Procrustes Analysis", Psychometrika, 40, 33-51.

Gower, J. and Hand, D. (1996) , "Biplots. Monographs on Statistics and Applied Probability", London: Chapman and Hall, 277p.

Gower, J. and Legendre, P. (1986) , "Metric and Euclidean properties of dissimilarity coefficients", Journal of classification. Vol. 3,5-48.

Graef, J. and Spence, I. (1979) , " Using distance information in the design of large multidimensional scaling experi- ments", Psychological Bulletin, 86, 60-66.

Green. E., Carmone, J., and Smith, M. (1989) , " *Multidimensional Scaling : Concepts and Applications*", Allyn and Bacon, London.

Hair, J., Anderson, T., Tatham, R., and Black, C. (1998) , " Multivariate data analysis", Upper Saddle River, NJ: Prentice Hall.

Han, J. and Kamber, M. (2006) , " Data Mining: Concepts and Techniques, 2nd Ed.Morgan Kaufmann Publishers Inc", San Francisco, CA, USA.

Hebert, A., Masson, H. and Denoeux, T. (2006) , "Fuzzy multidimensional scaling", Computational Statistics and Data Analysis, 51(1), 335-359

Heiser, W. (1991) , "A generalized majorization method for least squares multidimensional scaling of pseudo distances that may be negative", Psychometrika, Vol. 56, 7-27.

Heiser, W., and Meulman, J. (1983) , " Constrained multidimensional scaling, including confirmation", Applied Psychological Measurement, 7, 381-404.

Horn, D. (1943) , " A study of personality syndromes", Character and personality, 12, 257- 274.

Hui-Chuan , L. (2009)S, "urvey and Implementation of Clustering Algorithms", an Unpublished master's thesis for master's degree, Hsinchu, Taiwan, Republic of China.

Jain, A., Murty, M., and Flynn, P. (1999) , " Data clustering: A review. ACM Computing Surveys", 264 - 323. www.citeseer.ist.psu.edu/jain99data.html.

Johnson, S. (1967), "Hierarchical clustering schemes", Psychometric, 32, 241-254.

Johnson, S. (1967), "Hierarchical clustering schemes",. *Psychometrika* 32, 241-254.

Kaufman, L. and Rousseeuw, P. (1990), " Finding groups in data: An introduction to cluster analysis", Wiley-Interscience.

Kaufman, L. and Rousseeuw, P. (2005) , " Finding Groups in Data: An Introduction to Cluster Analysis", Wiley-Interscience.

Kruskal, J. (1964a) , " Multidimensional scaling by optimizing goodness of fit to anonmetric hypothesis", Psychometrika, 29, 1-27.

Kruskal, J. (1964b) , "Nonmetric Multidimensional Scaling: A Numerical Method",, *Psychometrika*, 29, 115-129.

Kruskal,J. and Wish, M. (1978) , " Multidimensional Scaling. Sage Publications Inc., Beverly Hills, CA, U.S.A., 1978.

Kuennapas, T, and Janson, A. (1969) , " Multidimensional similarity of letters", Perceptual and Motor Skills 28,3-12.

Laflin, S. (1998),"Laflin's general coefficient", Website http://www.cs.bham.ac. uk/~slb/courses/Taxonomy/Taxonomy02.html.

Lance, G. and Williams, W. (1967) , " A general theory of classificatory sorting strategies: 1 hierarchical systems", Computer journal, 9, 373-380.

Larose, D. (2004) , "Discovering Knowledge in Data: An Introduction to Data Mining", Wiley-Interscience.

McQuitty, L. (1957), " Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies", Educational and psychological Measurement, 17, 207-229.

Meila, M. and Shi, J. (2002) , "Learning segmentation by random walks In Neural ", Information Processing Systems, vol 13. MIT press, Cambridge.

Murtagh, F. and Contreras, P.(2011) , " Methods of Hierarchical Clustering ", Computing Research Repository - CORR , vol. abs/1105.0.

Pedrycz, W. (2005) , " Knowledge-Based Clustering ",From Data to Information Granules. Wiley-Interscience, perl/webwn?s=clustering.

Pruzansky, S., Tversky, A., and Carroll, J. (1982) , "Spatial versus tree representations of proximity data", Psyehometrika, 47, 3-24.

Ramsay, J. (1977), " Maximum likelihood estimation in multidimensional scaling", *Psychometrika* 42,241-266.

Rehman,M. and Mehdi,S. (2013) , " comparison of density-based clustering algorithms",from the World Wide Web: (http://www.researchgate.net/publication/242219043_COMPARISON_OF_DENSITY-BASED_CLUSTERING_ALGORITHMS).

Richardson, M. (1938),"Multidimensional psychophysics ",Psychological Bulletin, 35, 659-660.

Romesburg, C. (1984), "Cluster Analysis for Researchers", Belmont, California:Lifetime Learning Publications.

Sammon, J. (1969), "A non-linear mapping for data structure analysis", LEEE Transactions on computers. Vol. 18,401-409.

Sattath, S. and Tversky, A. (1977), " Additive similarity trees", *Psychometrika* 42,319-345.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. (1998), "WaveCluster: A multiresolution clustering approach for very large spatial databases", (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., pp. 428 - 439.

Shepard, R. (1962a), "Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function. I.," *Psychometrika*, 27, 125-140.

Shepard, R. (1962b) , " Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function. II.", *Psychometrika*, 27, 219-246.

Sibson, R. (1979) , " Studies in the robustness of multidimensional scaling:Perturbational analysis of classical scaling",Journal of the Royal Statistical Society, Series B, 41, 217-229.

Sibson, R. (1978) , " Studies in the robustness of multidimensional scaling: procrustes statistics ",Journal of the Royal Statistical Society, Series B,40, 234-238. Sibson, R.

Snyder Jr, c. Law, H. and Hattie, J . (1984) , " Research methods for multimode data analysis (pp. 372-402) ",New York. Praeger

Sokal, R. and Michener, C. (1958), " A statistical method for evaluating systematic relationships", University of Kansas Science Bulletin, 38, 1409-1438.

Sokal, R., and Sneath, P. (1963) , "Principles of Numerical Taxonomy", San Francisco and London, W.H. Freeman & Co, 359 p.

Takane,Y., Young, F. and Leeuw,J. (1977) , " Nonmetric individual differences multidimensional scaling: an alternating leasts quares method with optimal scaling features. *Psychometrika*, 42(1):7–67.

Tarpey, **T.,** (**2007**)**,** " Linear Transformations and the k-Means Clustering Algorithm: Applications to Clustering Curves", The American Statistician, 61, 34-40.

Teknomo, K. (2007) , " **Similarity Measurement**", Website http://people.revoledu. c o m / k a r d i / t u t o r i a l / S i m i l a r i t y / .

Thaler, N., Allen, D., McMurray, J. and Mayfield, J. (2010) , " Sensitivity of the Test of Memory and Learning to attention and memory deficits in children with ADHD", The Clinical Neuropsychologist, *24*, 246-264.

Theodoridis, S. (2009) , " K.: Pattern Recognition, 4th edn", Academic Press, New York. pp. 602, 605, 606.

Torgerson, W. (1958) , " Theory and methods of scaling", New York: Wiley.

Torgerson. W. (*1952*) , " Multidimensional scaling", *Psychometrika*. 17. 401-419.

Tryon, R. (1939) , " ***Cluster Analysis*** ",Edwards Brothers.

Tversky, A. and Gati, I. (1982) , " Similarity, separability, and the triangleinequality ",*Psychological Reviezu* 89,123-154.

Tversky, A. and Hutchinson, J. (1986) , " Nearest neighbor analysis of psychological spaces", *Psychological Reviezu* 93,3-22.

Tversky, A. (1977) , " Features of similarity", *Psychological Reviezu* 84,327-352.

University, P. WordNet 3.0. (2006) , " Website. http://wordnet.princeton.edu/ USA) ", IEEE Computer Society, pp. 641 - 646.

Velmurugan,T. and Santhanam,T.,(2011) ,"A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach" An experimental approach. Information. Technology",J.ournal, Vol, 10, No .3, pp478-484.].

Wang, W., Yang, J. and Muntz, R. (1997) , " STING: A statistical information grid approach to spatial data mining ",In VLDB '97:

Proceedings of the 23rd International Conference on Very Large Data Bases (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., pp. 186 - 195.

Ward, J. (1963) , " Hierarchical grouping to optimize an objective function", Journal of the American Statistical Association, 58, 236-244.

Webb, A. (2000) , " Statistical Pattern Recognition", 2nd Edition. John Wiley and Sons, Information Processing Systems, pp. 873 - 879.

Weinberg, R., Smith, J., Jackson, A., and Gould, D. (1984), " Effect of association, dissociation, and positive self-talk strategies on endurance performance", Canadian Journal of Applied Sport Sciences, *9*, 25-32.

Wikipedia (2008), " Mahalanobis distance", wikipedia, the free encyclopedia.

Wishart, D. (1969), " An algorithm for hierarchical classifications", Biometrics, 25, 165-170.

Witten, I. and Frank, E. (2005) , " Data Mining: Practical Machine Learning Tools and Techniques ",2 ed. Morgan Kaufmann.

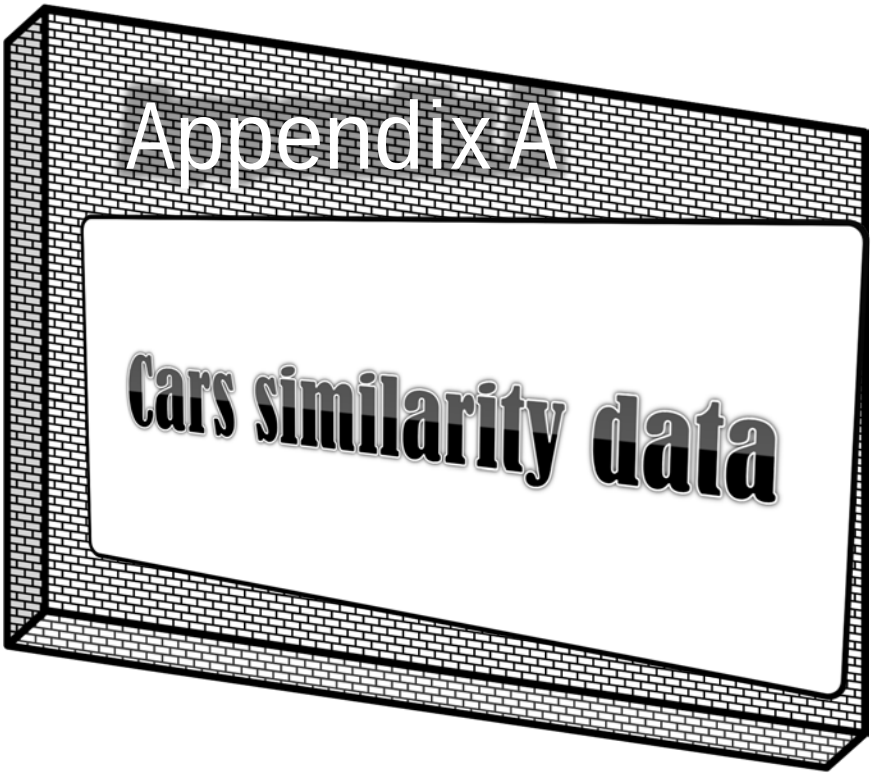
Xu, R. and Wunsch II, D. (2005) , " Survey of clustering algorithms", IEEE Transactions on Neural Networks 16,3, 645 - 678.

Yin, X. and Yu, P. (2005) , " Cross-relational clustering with user's guidance. In KDD '05: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining", (New York, NY, USA), ACM, pp. 344 -353.

Young, G. and Householder, A. (1938) , " Discussion of a set of points in terms of their mutual distances", *Psychometrika* 3,19-22.

المراجـــع العربيـــــة:

1- عبد الفتاح محمد أحمد قنديل (2011)، "التحليل متعدد المتغيرات– النظرية والتطبيق.


Appendix A
Cars similarity data

APPENDIX A: CARS SIMILARITY DATA

The following tables present the similarity matrix for each subject in the cars study.

Cars Similarity Matrix for Subject 1

	1	2	3	4	5	6	7	8	9	10
1										
2		23								
3		99	51							
4		99	23	78						
5		90	16	22	49					
6		74	55	50	99	13				
7		14	88	77	75	50	70			
8		25	95	48	99	99	79	99		
9		60	36	69	24	21	53	99	99	
10		0	89	72	81	77	71	74	51	71

Cars Similarity Matrix for Subject 2

	1	2	3	4	5	6	7	8	9	10
1										
2		62								
3		77	16							
4		98	14	55						
5		76	22	40	47					
6		84	16	16	81	7				
7		17	80	36	93	60	90			
8		76	93	86	80	94	36	19		
9		74	20	16	38	5	18	6	71	
10		10	72	78	92	92	86	16	2	99

Cars Similarity Matrix for Subject 3

	1	2	3	4	5	6	7	8	9	10
1										
2	85									
3	82	15								
4	97	28	56							
5	51	31	36	43						
6	79	27	7	82	7					
7	13	84	38	87	76	82				
8	82	99	73	68	80	40	20			
9	69	24	30	27	16	12	28	80		
10	15	80	78	90	72	66	17	5	95	

Cars Similarity Matrix for Subject 4

	1	2	3	4	5	6	7	8	9	10
1										
2	49									
3	96	96								
4	97	92	94							
5	68	12	90	93						
6	77	44	88	90	26					
7	97	93	94	25	93	49				
8	54	76	92	94	20	24	93			
9	47	48	92	94	35	18	94	23		
10	21	47	90	92	68	67	87	55	15	

Cars Similarity Matrix for Subject 5

	1	2	3	4	5	6	7	8	9	10
1										
2	9									
3	90	70								
4	87	65	6							
5	87	77	83	83						
6	33	79	25	89	39					
7	86	86	99	22	90	40				
8	81	30	57	88	69	39	97			
9	74	20	94	78	5	81	92	88		
10	23	26	72	94	2	76	81	20	5	

Cars Similarity Matrix for Subject 6

	1	2	3	4	5	6	7	8	9	10
1										
2	10									
3	53	75								
4	99	99	99							
5	87	27	65	99						
6	60	66	72	99	99					
7	96	99	90	10	90	75				
8	98	99	91	98	88	34	99			
9	73	15	90	99	9	56	95	75		
10	54	62	84	99	95	53	85	91	49	

Cars Similarity Matrix for Subject 7

	1	2	3	4	5	6	7	8	9	10
1										
2	69									
3	63	58								
4	76	85	79							
5	52	14	51	81						
6	61	39	35	83	36					
7	80	90	93	6	78	85				
8	28	87	83	94	64	44	9*o*			
9	80	20	92	98	51	23	80	33		
10	78	28	40	99	36	71	82	62	13	

Cars Similarity Matrix for Subject 8

	1	2	3	4	5	6	7	8	9	10
1										
2	16									
3	81	47								
4	56	32	71							
5	87	68	44	71						
6	60	35	21	98	34					
7	84	94	98	57	99	99				
8	50	87	79	73	19	92	45			
9	99	25	53	98	52	17	99	84		
10	16	92	90	83	79	44	24	18	98	

Cars Similarity Matrix for Subject 9

	1	2	3	4	5	6	7	8	9	10
1										
2	14									
3	61	47								
4	79	96	77							
5	72	21	12	73						
6	66	12	28	81	13					
7	66	64	75	41	71	82				
8	51	67	32	93	49	66	86			
9	7	20	67	71	15	56	76	69		
10	19	51	6	88	25	81	50	8	8	

Cars Similarity Matrix for Subject 10

	1	2	3	4	5	6	7	8	9	10
1										
2	11									
3	90	69								
4	72	26	90							
5	93	17	69	24						
6	39	34	36	98	80					
7	26	82	77	85	53	99				
8	80	74	75	99	93	87	13			
9	73	8	91	35	17	17	99	91		
10	24	62	90	76	85	64	77	24	65	

Cars Similarity Matrix for Subject 11

	1	2	3	4	5	6	7	8	9	10
1										
2	21									
3	97	49								
4	99	21	76							
5	88	14	20	47						
6	72	53	48	97	11					
7	12	86	75	73	48	68				
8	23	93	46	97	97	77	97			
9	58	34	67	22	19	51	97	97		
10	0	87	70	79	75	69	72	49	71	

Cars Similarity Matrix for Subject 12

	1	2	3	4	5	6	7	8	9	10
1										
2	64									
3	81	18								
4	98	16	57							
5	78	24	42	49						
6	86	18	18	83	9					
7	19	82	38	95	62	92				
8	78	95	88	82	96	38	21			
9	76	22	18	40	7	20	8	73		
10	10	74	80	94	94	88	18	4	99	

Cars Similarity Matrix for Subject 13

	1	2	3	4	5	6	7	8	9	10
1										
2	95									
3	92	25								
4	97	38	66							
5	61	41	46	53						
6	89	37	17	82	17					
7	23	94	48	97	86	92				
8	92	99	83	78	90	50	30			
9	79	34	40	37	26	22	38	90		
10	25	90	88	90	82	76	27	15	95	

Cars Similarity Matrix for Subject 14

	1	2	3	4	5	6	7	8	9	10
1										
2	39									
3	86	86								
4	97	82	84							
5	58	2	80	83						
6	67	34	78	80	16					
7	87	83	84	15	83	39				
8	44	76	82	94	10	14	83			
9	37	38	82	84	25	8	84	13		
10	11	37	80	92	58	57	77	45	15	

Cars Similarity Matrix for Subject 15

	1	2	3	4	5	6	7	8	9	10
1										
2	10									
3	91	71								
4	88	66	7							
5	88	78	84	84						
6	34	80	26	90	40					
7	87	87	99	23	91	41				
8	82	30	58	89	70	40	98			
9	75	21	95	79	6	82	93	89		
10	24	27	73	95	23	77	82	21	6	

Cars Similarity Matrix for Subject 16

	1	2	3	4	5	6	7	8	9	10
1										
2	9									
3	52	74								
4	98	98	98							
5	86	28	64	98						
6	59	67	71	98	98					
7	95	98	90	9	89	74				
8	97	98	90	98	87	33	98			
9	72	14	89	98	8	55	94	74		
10	53	61	83	98	94	52	84	90	48	

Cars Similarity Matrix for Subject 17

	1	2	3	4	5	6	7	8	9	10
1										
2	61									
3	62	58								
4	77	85	80							
5	53	14	52	81						
6	62	39	36	83	37					
7	81	90	94	6	79	85				
8	29	87	84	94	65	44	91			
9	81	20	93	98	52	23	81	33		
10	79	28	41	99	37	71	83	62	14	

Cars Similarity Matrix for Subject 18

	1	2	3	4	5	6	7	8	9	10
1										
2	15									
3	80	47								
4	55	32	70							
5	86	68	43	71						
6	59	35	20	98	33					
7	83	94	97	57	98	99				
8	49	87	78	73	18	92	44			
9	98	25	52	98	51	17	98	84		
10	15	92	89	83	78	44	23	18	97	

Cars Similarity Matrix for Subject 19

	1	2	3	4	5	6	7	8	9	10
1										
2	24									
3	81	47								
4	89	96	87							
5	82	21	22	73						
6	76	12	38	81	23					
7	76	64	85	41	81	82				
8	61	67	42	93	59	66	96			
9	17	20	77	71	25	56	86	69		
10	29	51	16	88	35	81	60	8	18	

Cars Similarity Matrix for Subject 20

	1	2	3	4	5	6	7	8	9	10
1										
2	1									
3	80	69								
4	62	26	80							
5	83	17	59	24						
6	29	34	26	98	70					
7	16	82	67	85	43	99				
8	70	74	65	99	83	87	3			
9	63	8	81	35	7	17	89	91		
10	14	62	80	76	75	64	67	24	55	

جامعـة بنهـــــا
كليـــــة التجـــــارة
قسم الإحصاء والرياضة
والتأمين

التحليل التجميعي المتعدد الابعاد وتطبيقه

إعـداد
مها عبدالله ابراهيم موسي
معيدة بقسم الإحصاء والرياضة والتأمين

إشـراف

الأستاذ الدكتور
عبد الفتاح محمد قنديل
أستاذ الإحصاء وعميد كلية التجارة السابق
جـــامعـــة بنهـــــا

الدكتور
زهدي محمد نوفل
أستاذ مساعد الإحصاء بكلية التجارة
جـــامعـــة بنهـــــا

رسالة مقدمة إلى قسم الإحصاء ، الرياضة والتأمين، بكلية التجارة

جامعة بنها – كجزء مكمل لمتطلبات الحصول على درجة الماجستير في الإحصاء

2015

الملخص العربي

يعتبر كلا من التحليل المتعدد الأبعاد والتحليل التجميعي من التقنيات الرقمية التي تساعد الباحث في التحقق من بنية البيانات في مختلف المجالات. باستخدام التحليل المتعدد الأبعاد يستطيع الباحث تحويل كمية كبيرة من بيانات التشابه و الأختلاف الي رسم هندسي. وباستخدام التحليل التجميعي يتم تجميع العناصر المتشابهة معا في نفس المجموعة.

يفضل استخدام التحليل التجميعي مع التحليل المتعدد الأبعاد حيث ان

1- يعمل التحليل التجميعي علي تقديم صورة أوضح للتشابه بين للبنات أذا كان ذلك غير واضح بالرسم الهندسي للتحليل المتعدد الأبعاد.
2- في العديد من مشاكل التحليل التجميعي يكون ليس لدينا بيانات رقميةعن المتغيرات ويكون فقط لدينا بنات عن التشابه والأختلاف بين المتغيرات. وبينات التشابه والأختلاف اما أن تكون رقمية أو غير رقميه . ولتحويل البيانات الغير رقمية الي بيانات رقمية فيتم أستخدام التحليل المتعدد الأبعاد.

يتوافر للباحث العديد من أساليب التحليل المتعدد الأبعاد والتحليل التجميعي و يعتمد الأختيار لهذه الأساليب بصفة أساسية علي نوع البيانات تحت الدراسة.

في هذه الرسالة تم تقديم الأنواع المختلفة لكلا من أساليب التحليل المتعدد الأبعاد والتحليل التجميعي مقرنة كلا منها بأمثلة رياضية محلولة للتوضيح.

حيث أنه يعتمد كلا من التحليل التجميعي و التحليل المتعدد الأبعاد بصفة اساسية علي علي بينات التشابه و الأختلاف فقد قدمنا الأنواع المختلفة لمقايس التشابه والأختلاف مقرنة كلا منها بأمثلة رياضية محلولة للتوضيح.

في هذا البحث تم تقديم تطبيق علي التحليل التجميعي و التحليل المتعدد الأبعاد علي بيانات فعلية تم تجميعها عن طريق أستمارة أستبيان التي وزعت علي معارض السيارات بمدينة بنها. وكان حجم العينة 20 شخص. أحتوت أستمارة الأستبيان علي كل أسماء الأزواج الممكنة لعشرة سيارات. تم الطلب من هذه الاشخاص بأعطاء رقم للتشابه والاختلاف لكل زوج سيارات من من هذه الأزاواج. ترواح هذا الرقم بين صفر ومائه. فأذا رأي الشخص أن زوج السيارات مختلفين الي أعلي درجة أعطي الرقم صفر. و أذا رأي الشخص أن زوج السيارات متشابه الي أعلي درجة أعطي الرقم مائه.

تم استخدام برنامج (SPSS) لتطبيق التحليل المتعدد الأبعاد لتحويل بيانات التشابه لسوق السيارات ال رسم هندسي. ثم تم استخدام برنامج (SPSS) لتطبيق التحليل التجميعي اتكوين مجموعات متشابه فيما بينها لأنواع السيارات المختلفة محل الدراسة. وبعد الحصول علي النتائج تم تفسير النتائج وأوصينا شركات السيارات تحت الدراسة بأستخدام هذه النتائج للتعرف علي أنواع السيارات الأكثر تشابه لها أي الأكثر منافسة لها.

ولتحقيق غرض الدراسة تم تقسيمها على النحو التالى:

- **الفصل الاول:** فى هذا الفصل تم عرض اهمية الدراسة و الدراسات السابقة التى تناولت الموضوع محل الدراسة.
- **الفصل الثاني:** في هذا الفصل تم عرض مفهوم مقايس التشابه والاختلاف وانواعها المختلفة مع امثله رياضية محلولة لتوضيح كل منها.

- **الفصل الثالث التحليل المتعدد الابعاد:** في هذا الفصل تم عرض مفهوم التحليل المتعدد الابعاد وانواعه مع وجود امثلة رياضية محلولة لتوضيح كل منها.

- **الفصل الرابع** : في هذا الفصل تم عرض مفهوم التحليل التجميعي و طرقه المختلفة مع وجود امثله رياضية محلولة لتوضيح كل منها.
- **الفصل الخامس:** في هذا الفصل تم عرض تطبيق التحليل المتعدد الابعاد والتحليل التجميعي علي بيانات فعلية تم تجميعها باستمارة استبيان تم توزيعها علي محلات بيع وتاجير السيارات لاعطائها للمستهلك وكان حجم العينة 20 شخص وتم اخذ المتوسط الحسابي لهذه البيانات (بيانات التشابه والاختلاف المجمعة من ال20 شخص) وادخالها الي برنامج ال(SPS)وتم الحصول علي النتائج كما هو موضح بالفصل الخامس.