Design and Analysis of Clustering Algorithms for Numerical, Categorical and Mixed Data

A thesis submitted to Cardiff University

for the degree of

Doctor of Philosophy

by

Maria Del Mar Suarez Alvarez

Manufacturing Engineering Centre

Cardiff University

United Kingdom

February 2010

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

In recent times, several machine learning techniques have been applied successfully to discover useful knowledge from data. Cluster analysis that aims at finding similar subgroups from a large heterogeneous collection of records, is one of the most useful and popular of the available techniques of data mining.

The purpose of this research is to design and analyse clustering algorithms for numerical, categorical and mixed data sets. Most clustering algorithms are limited to either numerical or categorical attributes. Datasets with mixed types of attributes are common in real life and so to design and analyse clustering algorithms for mixed data sets is quite timely. Determining the optimal solution to the clustering problem is NP-hard. Therefore, it is necessary to find solutions that are regarded as "good enough" quickly.

Similarity is a fundamental concept for the definition of a cluster. It is very common to calculate the similarity or dissimilarity between two features using a distance measure. Attributes with large ranges will implicitly assign larger contributions to the metrics than the application to attributes with small ranges. There are only a few papers especially devoted to normalisation methods. Usually data is scaled to unit range. This does not secure equal average contributions of all features to the similarity measure. For that reason, a main part of this thesis is devoted to normalisation.

The first part of the thesis concentrates on the development of a mathematically rigorous approach to normalisation of the feature vectors for mixed data sets based on a unified statistical approach. The most common cases of metrics, namely the Euclidean metrics are used as a measure for continuous numerical features, while the matching dissimilarity measure is used to deal with categorical attributes. The introduced normalised metrics secure that the average contributions of all attributes to the measures are equal to each other from statistical point of view.

The second part of the thesis concentrates on the application of the unified statistical approach to the general case of the Minkowski metrics and the development of a novel algorithm for hard clustering using the Minkowski distances with an appropriate objective function. The algorithm may be used in these cases, while the k-prototypes is not applicable.

The third part of the thesis introduces the RANKPRO (the Random Search with k-prototypes algorithm). It combines the advantages of the Bees and k-prototypes algorithms and outperforms the latter algorithm. The RANKPRO balances two objectives: first it explores the search space effectively due to random selection of new solutions, and on the other hand it improves promising solutions fast due to employment of several steps of the k-prototypes algorithm.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor D.T. Pham, for his invaluable guidance during the course of this study.

I would also like to thank my family for their love and encouragement.

I am very grateful to Dr. Yuri Prostov for his valuable comments on the thesis and for discussing the theoretical and practical aspects of clustering algorithms, related problems and possible ways to their solution.

I am very grateful to Mr. M. Prostov for his valuable discussions on statistical approaches to clustering procedures.

I am also grateful to all the members of the MEC Machine Learning Group for providing useful technical discussions.

This work is supported by the Manufacturing Engineering Centre, Cardiff University.

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed.....M Suarez.....(Maria del Mar Suarez Alvarez - Candidate)

Date.....September 2009.....

Statement 1

This thesis is being submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD).

Signed.....M Suarez.....(Maria del Mar Suarez Alvarez - Candidate)

Date.....September 2009.....

Statement 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed.....M Suarez.....(Maria del Mar Suarez Alvarez- Candidate)

Date.....September 2009.....

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed.....M Suarez.....(Maria del Mar Suarez Alvarez - Candidate)

Date.....September 2009

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

BA	Bees Algorithm.
CRS	Controlled Random Search.
EM	Expectation Maximisation Algorithm.
GA	Genetic Algorithms.
LBS	Local beam search.
OF-based clustering	Objective function-based clustering.
RANKPRO	The Random Search with k-Prototypes Algorithm.
SBS	Stochastic beam search.
SI	Swarm intelligence.

SYMBOLS

$\mathbf{A}_i$	Records of a data set.
$\mathbf{X}_i$,	Numerical part of a record of a data set.
$\mathbf{Y}_i$,	Categorical part of a record of a data
set.	
X_{1j}, X_{2j}	Independent random variables.
N	Number of records in a data set or the number of elements in a sample.
p	Number of numerical attributes in a record.
l	Number of categorical attributes in a record.
L_p	Linear vector space (Minkowski space).
ρ_E	Euclidean metric.
ρ_{cat}	Categorical metric.
ρ_{max}	Tchebysheff (Chebyshev) or maximum norm metric.
$\rho^*_{p_M}$	Normalised Minkowski metric.
ρ_H	Huang's mixed metric.
γ	Weight in the Huang mixed metric.
J	Objective function.
J_{j,p_M}	Hathaway's objective function.
u_{im}	Element of the partition matrix.
k	Number of clusters.

$\mathbf{Q}_m$	Prototype (centre).
C_m	Cluster m.
$\lvert C_m \rvert$	Number of elements in the cluster C_m.
$\mathbf{X} = (X_1, X_2, \ldots, X_N)$	Random vector formed by a sequence of random variables.
$\mathbf{x} = (x_1, x_2, \ldots, x_N)$	The actually observed sample value.
x_{ij}^*	Normalised attribute value in the data set.
$x_{max,j}$	Maximum value of an attribute.
$x_{min,j}$	Minimum value of an attribute.
μ_j and σ_j	Mean and standard deviation of a random variable.
$\bar{X}_j$ and S_j	Estimators for mean and standard deviation of a sample.
$d_j(x_{ij}, x_{i'j})$	Hastie's attribute dissimilarity.
$D(x_i, x_{i'})$	Hastie's overall measure of dissimilarity.
w_j	Weight assigned to the j-th attribute.
$\bar{D}$	Hastie's average object dissimilarity measure.
$\overline{d_j}$	Hastie's average dissimilarity of the j-th attribute.
E	Expectation of a variable.
$\hat{E}$	Estimator of the expectation.
s_j^2	Estimator of the sample variance.

α_j Inverse of expectation of contribution of the j-th numerical attribute to a metric.

β_j Inverse of expectation of contribution of the j-th categorical attribute to a metric.

$\{p_{j1}, p_{j2}, \ldots, p_{jq_j}\}$ Probabilities of possible states for a categorical attribute.

$\omega(y_{1j}, y_{2j})$ Distance between two categorical attributes.

Acc_{NW} The Ng and Wong accuracy.

$\mathbf{C} = \{C_1, \ldots, C_k\}$, $\mathbf{D} = \{D_1, \ldots, D_k\}$ Two partitions of a set.

R the Rand index.

$n_{m,j}$ Number of records with the attribute a_j that belongs to the m-th cluster.

$n_{m,\varphi(m)}$ Number of records of the m-th cluster whose state of the attribute A is the same as the assigned $a_{\varphi(m)}$.

$Acc(\varphi)$ Clustering accuracy for an assignment φ.

Acc Clustering accuracy.

$\Gamma(x)$ Euler gamma function.

$\| X \|_{p_M}$ Minkowski norm.

$\varphi(z_1, z_2)$ Function of two real valued arguments (Proposition 4).

C_N, V_1, V_2 Sets of chords used in the proof of Lemma 4.

D Dispersion.

$\partial\Phi_{mj}(t)$ Subgradient of $\Phi_{mj}(t)$.

n Number of scout bees (BA).

m	Number of best sites (BA).
e	Number of elite sites (BA).
d_{ngh}	Size of neighbourhood around any of the best sites (BA).
r_e	Number of recruited bees within the neighbourhood for the elite sites (BA).
r_g	Number of recruited bees around other selected sites (BA).
n	Number of the approximate solutions (RANKPRO).
e	Number of kept elite solutions (RANKPRO).
$r = n - e$	Number of solutions used for random search (RANKPRO).
t_{exec}	Time specified for the RANKPRO algorithm.
n_r	Given number of simulations for the RANKPRO algorithm.
$J_{av}(t_{exec})$	Average value of the objective function obtained in n_r simulations.
n_{iter}	Number of iterations of the k-prototypes algorithm (RANKPRO).

Chapter 1

Introduction

This chapter introduces the motivation and objectives of the research, and a general description of adopted methods and approaches. The chapter also outlines the general structure of the thesis.

1.1 Motivation

There is an increasing amount of data being collected everyday but only the part that can be used for extracting knowledge becomes valuable. Data Mining (DM) may be defined as a process of extracting useful knowledge in the form of relations and structure from large amount of data. The derived knowledge can then be applied to achieve economic, operational or other benefits.

In this thesis DM is considered as a synonym to the knowledge discovery process or knowledge discovery in databases. This process consists of a set of processing steps that should be followed to discover relations and structure in data. DM needs to develop appropriate tools to efficiently and effectively extract previously unknown information

from raw collections of data. In this thesis we deal with objective function-based clustering that is also called partition based clustering.

Partitioning is a natural way of studying complex problems in a number of areas like pattern recognition, classification and clustering. In a number of fields of machine intelligence, an object is represented by a vector variable (the feature vector). In application to data sets organised as flat files, the rows represent records, the columns represent features that are called attributes, and hence the feature vector can be defined as a set of attributes. Each attribute can take on a finite or infinite (continuous) number of possible values. In many traditional applications, it is assumed usually that all the features are the same type. Clustering of numerical data sets are the most studied problem. However, real-life data sets are often mixed, i.e. they consist of both numerical and categorical types. Currently methods for analysis of data in mixed feature space are still an issue. Hence, design and analysis of clustering algorithms for numerical, categorical and mixed data sets are very timely.

In this thesis we will deal with normalisation. Strictly speaking normalisation has to be applied to all records of data sets before clustering. Indeed, if the data is not normalised then the average contribution of each feature to the similarity measure depends on the units of measurements of the feature and, therefore, the contribution of the features are scale dependent. If the units of a measurement are changed then the contribution of a feature to the similarity measure can change dramatically. This is why normalisation of data sets is widely used in a number of fields of machine intelligence.

In the overwhelming majority of published normalisation procedures, data have been scaled to unit range. However, after this kind of data set normalisation, the average contributions of all features to the similarity measure may be not equal to each other.
It has been often suggested also to truncate the out-of-range components assuming that it is just eliminating the outliers. However, truncating the out-of-range components could lead to loss of information from the data set.

In spite of the importance of data normalisation, there have been only few papers specifically devoted to normalisation methods for data sets. It has been correctly realised that a normalisation procedure for numerical data sets, has to be a transformation of the attribute to a random variable with zero mean and unit variance. Indeed, this scaling provides equal contributions of variables to the Euclidean similarity measure. However, one needs to apply normalisation not only to numerical attributes but also to categorical attributes.

A natural way for normalisation of all numerical, categorical and mixed data sets is to employ a statistical approach. However, early papers on statistical approaches were not targeted to clustering of mixed data sets and normalisation of metrics. It was stated that methods for analysis of data in mixed feature space are still an issue. For example, one can expect that the mean of the distance between two categorical attributes that may have only two states (e.g. male - female or white - black) is not the same as the mean of the distance between two categorical attributes that may have twenty different states. Some authors have used the averages of distance measures for normalisation. However,

nothing was known about statistical consistency of the proposed estimators. In addition, the estimators were biased and these approaches were not applicable to some metrics. Hence, mathematically rigorous treatment of the normalisation procedure is needed and explicit presentation of normalised mixed metrics has to be provided.

After normalisation of data, appropriate algorithms for efficient and effective clustering of data sets with mixed numerical and categorical values have to be developed. Currently the most popular is the *k*-prototypes algorithm for clustering of mixed data sets. This algorithm is a generalisation of the *k*-means algorithm. The latter is applicable only to numerical data sets. These algorithms have the same common drawback, namely the search process of new solutions converges often not to a global minimum but to a local minimum. Hence, new algorithms have to balance two objectives: to explore the search space effectively and to utilise the most promising solutions during the work of the algorithm.

1.2 Research Objectives

The aim of this research is to design and analyse new clustering algorithms for numerical, categorical and mixed data sets. Most clustering algorithms are limited to either numerical or categorical attributes. Datasets with mixed types of attributes are common in real life and so to design and analyse clustering algorithms for mixed data sets is quite timely.

The specific objectives are:

1. To develop a mathematically rigorous approach to normalisation of feature vectors for mixed data sets based on a unified statistical approach.
2. To analyse the clustering algorithms with proposed new normalised metrics in the case of the matching dissimilarity measure being used to deal with categorical attributes, and the general Minkowski metrics being used as a measure for continuous numerical features, including the particular cases $p_M = 2$ (the Euclidean metric).
3. To develop a new algorithm to be used in the cases where $p_M \neq 2$, since the k-prototypes cannot be used in those cases. This clustering algorithm was earlier suggested only for fuzzy clustering. It will be developed and applied for hard clustering using Minkowski norm distances.
4. To develop a new unsupervised clustering algorithm for numerical, categorical and mixed data sets that will have less probability for premature convergence than the *k*-prototypes algorithm. The algorithm has to balance two objectives: to explore the whole search space effectively, and to improve promising solutions fast. The new algorithm has to combine the advantages of both the Bees and the *k*-prototypes algorithms and to outperform the algorithms.

1.3 Methods and approaches

For the four objectives targeted in this thesis, several methods and approaches will be employed. They are summarised as follows:

1. A unified statistical approach to both numerical and categorical attributes is applied for normalisation of the feature vectors for mixed data sets in both cases; the Euclidean and the general Minkowski metrics. Normalised Minkowski and Euclidean metrics and metrics for mixed data sets are introduced in an explicit way. The introduced generalised statistical procedure assures that the means of the different normalised attributes are equal to each other and therefore, these variables give equal contributions to the similarity measures.
2. In the case where $p_M = 2$, the k-prototypes clustering algorithm will be implemented and applied to data sets from the UCI repository with and without normalisation of attributes and the accuracy of clustering results will be compared by both a new approach for calculating the accuracy and the traditional Rand index.
3. A unified statistical approach to general cases of the Minkowski distances and the development of a novel algorithm for hard clustering using the Minkowski distances with an appropriate objective function. Implemented

codes are applied to two data sets from the UCI repository with and without normalisation of attributes for various values of the Minkowski power p_M.

4. A new clustering algorithm called RANKPRO: the Random Search with *k*-prototypes algorithm will be presented. The algorithm combines the advantages of the Bees and *k*-prototypes algorithms. The algorithm balances two objectives: it explores the search space effectively due to random selection of new solutions, and improves promising solutions fast due to employment of the *k*-prototypes algorithm. The RANKPRO algorithm will be applied to various data sets, including data sets with mixed numerical and categorical values and its performance will be compared with the performance of the *k*-prototypes algorithm.

1.4 Outline of the thesis

The thesis is organised in six chapters. The topics addressed in each chapter are as follows:

Chapter 2: In this Chapter notations and definitions of some concepts related to clustering, similarity measures for numerical, categorical and mixed data sets, objective functions, and statistical estimators, are recalled. The chapter ends with a literature review of the most recent applications of object-function based clustering for mixed data sets.

Chapter 3: In this Chapter a unified statistical approach to both numerical and categorical attributes is applied in order to normalise the feature vectors for mixed data sets. The most common cases of metrics, namely the Euclidean metrics are used as a measure for continuous numerical features, while the matching dissimilarity measure is used to deal with categorical attributes. New normalised metrics are introduced such that the average contributions of all attributes to the measures are equal to each other from statistical point of view. Advantages of the introduced normalised metrics are demonstrated on examples of their applications to various data sets.

Chapter 4: In this chapter, a new statistical approach introduced in Chapter 3 is developed further and applied in the case of the Minkowski metrics being used as a measure for continuous numerical features, while to deal with categorical attributes again the matching dissimilarity measure is used. Various mathematical problems related to the normalisation of mixed metrics are resolved. The introduced metrics are applied to some data sets when it is more advantageous to apply the general Minkowski metrics (including the Tchebysheff and city-block metrics) instead of a particular case $p_M = 2$ (the Euclidean metrics). Since the k -prototypes cannot be used in the cases where $p_M \neq 2$, a new algorithm to be used in those cases will be developed. This clustering algorithm was earlier suggested only for fuzzy clustering. It will be developed and applied for hard clustering using Minkowski norm distances.

Chapter 5: In this Chapter a new clustering algorithm called RANKPRO: the Random Search with k-Prototypes Algorithm is presented. The algorithm combines the

advantages of a recently introduced by Pham et al. (2006b) population-based search algorithm called the Bees Algorithm (BA), and *k*-prototypes algorithm proposed by Huang (1997b) as an extension of the *k*-means algorithms to cluster large data sets with mixed numerical and categorical values. The RANKPRO algorithm balances two objectives: it explores the search space effectively due to random selection of new solutions, and improves promising solutions fast due to employment of the *k*-prototypes algorithm. The efficiency of the new algorithm is demonstrated by clustering several numerical, categorical and mixed data sets.

Chapter 6: In this Chapter conclusions and the main contributions of this thesis are presented. Finally, suggestions for future research in this field are provided.

Chapter 2

Preliminaries and Literature Review

In this Chapter notations and definitions of some concepts related to data models, clustering, similarity measures for numerical, categorical and mixed data sets, and objective functions, are recalled. Some mathematical and statistical notions used in clustering analysis are also reminded. The chapter ends with a literature review of the most recent applications of objective - function based clustering for mixed data sets.

2.1 Data and data types

It is well known (see e.g. Jain and Dubes, 1988, Cios et al., 2007) that data can have diverse formats and can be stored trough a variety of different storage models. In a number of fields of data mining an object is represented by a vector variable, namely the feature vector **A** (Jain et al. 1999). In application to databases, the features are called attributes, and hence **A** can be defined as a set of attributes $\mathbf{A} = \left(A_1, A_2, ..., A_{p+l}\right)$. The collection of objects described by the same features is called a data set. Data sets may be stored as flat files and in other formats using databases and data warehouses. Flat (rectangular) files are the most common way to store the data sets and further we will deal only with flat files. The rows represent objects (also known as records, individuals, patterns, data points) and the columns represent features.

Each attribute can take on a finite or infinite (continuous) number of possible values. In many traditional applications, it is assumed usually that all the features are the same type. However, real-life data sets are often mixed, i.e. they consist of both numerical and categorical types. It is known that the measurement scale of a categorical variable consists of a set of categories. Only two data types of attributes are considered here, namely numerical and categorical because other types of attributes can be transformed to these two types. For mixed data, the vector of features $\mathbf{A}$ can be split into $\mathbf{A} = \left(\mathbf{A}^n, \mathbf{A}^c\right)$, namely the vector of numerical features $\mathbf{A}^n = \left(A_1^n, ..., A_p^n\right)$ and the vector of categorical features $\mathbf{A}^c = \left(A_1^c, ..., A_l^c\right)$.

2.1.1 Original Stevens' classification of variables.

It is generally accepted that the "levels of measurement", or scales of measure are expressions that typically refer to the classification of scale types developed by the psychologist S.S. Stevens. Stevens (1946) argued that measurements can be classified into four different types of scales: nominal, ordinal, interval and ratio.

Stevens's classification said that nominal is synonym of categorical. There has been, and continues to be, debate about the merits of Stevens's classification, particularly in the cases of the nominal and ordinal classifications (Michell, 1986).

The Table 2.1 presents a slightly modified classification of variables and appropriate statistical notions and mathematical operations that should be used for analysis of each scale type of variables

Scale Type	Permissible Statistics	Admissible Scale Transformation	Mathematical structure
nominal (also denoted as categorical or discrete)	mode, chi square	Equality (=)	standard set structure (unordered)
ordinal	median, percentile	Order (<)	totally ordered set
interval	mean, standard deviation, correlation, regression, analysis of variance	Subtraction (−) and weighted average	affine line
ratio	All statistics permitted for interval scales plus the following: geometric mean, harmonic mean, coefficient of variation, logarithms	Addition (+) and multiplication (×)	field

Table 2.1: A modified Stevens' classification of variables (scale types), and appropriate statistical notions, mathematical operations and structure.

2.1.2 Accepted classification of variables.

In this thesis we accept the term categorical as a general term that can be split into levels: nominal and ordinal. If categorical variables have ordered scales they are called ordinal variables, while the variables having no ordered scales are called nominal variables. Hence, for nominal variables, the order of listing the categories is irrelevant, and the statistical analysis should not depend on that ordering (Agresti 1996). We consider also binary variables as categorical.

Further, in this thesis we accept the scale types: interval and ratio are numerical variables. For numerical or quantitative features, the feature domain Dom (A_j) can be represented on the real line, i.e. they are continuous variables. For categorical features (sometimes these features are also called qualitative), the domain is a finite set of different states. Evidently, categorical features may be represented by numerical codes of possible different states of the feature. A data set can be represented as a matrix of size $N \times (p+l)$ where N is the number of records, and $(p+l)$ is the total number of attributes, i.e. the i-th row of the matrix represents the i-th record of the data set $(1 \le i \le N)$. This row is a vector $(x_{i1},\ldots,x_{ip},y_{i1},\ldots,y_{il})$, whose values $x_{i1},\ldots,x_{ip}$ are numerical, while the values $y_{i1},\ldots,y_{il}$ are categorical.

One can see from the above Table that the central tendency of a categorical attribute can be represented by its mode, but the mean cannot be defined. This observation was used

by Huang (1997, 1998) in his generalisation of a very popular clustering algorithm, the k-means algorithm. If the k-means algorithm can be applied only to numerical data sets, the k-modes algorithm can be applied to categorical data sets. These algorithms will be discussed later.

2.2 Constructing data models

As it has been noted in Chapter 1, the aim of Data Mining is to extract knowledge from data. Methods of machine analysis of data can be roughly divided into two fundamental groups: supervised and unsupervised learning.

In supervised learning, characteristics to records of data sets are given. The characteristics can be expressed either in the form of some discrete labels or as some values of auxiliary continuous variables. In the former case, we deal with a classification problem; while in the later case we deal with a regression, or an approximation, or continuous prediction problem (see e.g. Cios et al., 2007). Supervised learning includes various approaches such as statistical methods, including Bayesian methods (Pham and Ruz, 2009); neural networks; decision trees, rule algorithms, and their hybrids. Any supervised learning method has to be provided with a training data set that represents information about some domain of the data set. In classification problems, the objective of supervised learning is to construct a function (classifier) that generates for each record (individual) a class label as its output. Using a training data set rules are produced; these

rules are used to predict the labels of new unseen examples (i.e., examples not in the training set).

Unsupervised learning assumes that the data knowledge process does not involve any supervision and it discovers a structure in data automatically. Unsupervised learning includes various approaches such as association rules and clustering. Clustering aims at finding smaller, more homogeneous groups from a large heterogeneous collection of items (Anderberg, 1973, Berry and Linoff, 1997). Computer-assisted analysis must partition objects into groups, and must provide an interpretation of this partition (Berry and Linoff, 1997).

As it is well known, clustering is an inductive process (Bezdek and Pal, 1992, Estivill-Castro, 2002). This means that using particular observations of data, isolated facts are explained first by some empirical generalisations (working hypotheses) and then by a general theory. In application to clustering of data sets, this means that any partition produced by an algorithm or a human is a hypothesis to suggest (or explain) groupings in the data. The mathematical formulation of the inductive principle is called clustering criterion (see e.g. Kim et al., 1988; Doherty et al., 1988, Estivill-Castro and Murray, 1998, Halkidi et al., 2000; 2001). It discriminates one grouping hypothesis over another one for the same data set. The models are the structures used to represent clusters, while the induction principle selects a "best fit" model for a given data set. Several induction principles corresponding to specific clustering algorithms will be discussed later.

By breaking the object into smaller homogeneous parts that can be each analysed and explained separately, one can understand very sophisticated phenomena. The selected hypothesis becomes a model for the data, and can potentially constitute a mechanism to classify unseen instances of the data. This is the reason why clustering algorithms have been studied so extensively. In particular, efficient clustering is a fundamental task in data science, where the goal is to discover similarities within a large data set.

2.3 Some mathematical notions used in clustering analysis

The cluster analysis in general and the objective function-based cluster analysis in particular are mathematically based disciplines where one needs to work with various mathematical notions like norm, metric, distance, and others. Hence the definitions of these mathematical concepts and the proper use of the concepts are crucial for cluster analysis. Indeed, the aim of clustering is to group the closest data points together. Hence, clustering relays on calculating distances between records. Thus, to measure quantitatively the distinction between elements of the data sets, i.e. to formulate similarity or dissimilarity criteria, one needs to use the concept of the distance and other above mentioned concepts.

2.3.1 Concepts of metric and distance.

Let us consider a set M. A metric on a set M is a function which defines a positive real number (distance) between any two elements x and y of the set. For all x, y, z in M, a metric should satisfy the following conditions:

Identity of indiscernibles: $\rho(x,y) = 0$ if and only if $x = y$.

Non-negativity: $\rho(x,y) \geq 0$.

Symmetry: $\rho(x,y) = \rho(y,x)$.

The triangle inequality: $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$.

An example of a trivial metric is the discrete metric, i.e. if $x = y$ then $\rho(x,y) = 0$. Otherwise, $\rho(x,y) = 1$. However, the most popular example is the Euclidean distance; the distance between distinct points is positive and the distance from x to y is the same as the distance from y to x. The latter metric is translation and rotation invariant.

Other examples of metrics will be given later. We will consider mainly Minkowski metrics of degree p_M that include the Euclidean metric as a particular case ($p_M = 2$).

2.3.2 Concept of norm.

Let us consider a real vector space R^n, i.e. its elements $x \in R^n$ are vectors with real-valued entries. A *norm* of a vector x is denoted by $\|x\|$. This is a function that assigns a strictly positive real number to all vectors in the vector space, other than the zero vector. A norm should satisfy the following conditions:

1. $\|x\| > 0$ if $x \neq 0$,and $\|x\| = 0$ if and only if $x = 0$.

2. A norm is a linear function, i.e. multiplying a vector by a real number α changes its norm linearly

$\|\alpha x\| = |\alpha| \cdot \|x\|$.

3. A norm satisfies the triangle inequality for any two elements x and y.

$\|x + y\| \leq \|x\| + \|y\|$

In the case of norm beign a distance, this inequality means that the distance from point A through B to C is never shorter than going directly from A to C.

The above mentioned definitions allow the researcher to dismiss some models suggested for clustering. For example, Wu and Yang (2002) introduced an alternative to c-means clustering algorithm and they employed the following function:

$d(x, y) = 1 - \exp(-\beta \|x - y\|^2)$.

They called this function "distance" and claimed that it is a metric. However, one can see that d does not satisfy the triangle inequality and therefore this function is not a metric.

There is the following relation between norms and metrics:
Every norm determines a metric and some metrics determine a norm.
Norms are used in Chapters 3 and 4.

2.3.3 Concepts of random variables, mathematical expectation, mean, mode and median.

Throughout this thesis we will employ statistical treatment of data sets. In the framework of our approach each record (the row) of a data set will be regarded as a random sample of a population under consideration, i.e. a data set is treated as a set of N observations (samples), while each sample (record) is considered as a realisation of possible values of the feature vector $\mathbf{A}$. Of course, the basic concepts can be found elsewhere (see, e.g. Spiegel, 1975). Hence, only some concepts of probability theory and statistics that will be actively used in the thesis will be recalled. As usual, capital letters X and Y will be used to denote random variables and lower-case letters, x and y to denote the specific values that those variables may take.

For a continuous random variable X that has a density function $f(x)$, the mathematical expectation of $E(X)$ is defined as

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx.$$

Another term for the mathematical expectation is the mean that is denoted by μ_X or by μ. It represents the average of the values of the random variable.

The median of the random variable X corresponds to an ordinate which separates the area under the density function graph into two parts having equal areas, i.e. the median is that value x for which

$$P(X \leq x) = P(X \geq x) = \frac{1}{2}.$$

The mode is that value x which occurs most often or, in other words, has the greatest probability of occurring. At this value $f(x)$ has its maximum.

2.3.4 Concepts of statistic and estimators.

For statistical treatment of feature vectors, one needs to know the probability distributions of their attributes. Probability distributions are normally unknown because one has only a random sample. It is known that estimation is a way of extracting valuable information about the distribution of probability that generated it from a sample.

An observable function of the random data variable is called a statistic. If there is an unknown real parameter θ taking values in a real parameter space then a real-valued statistic that is used to estimate the parameter is called an estimator of this parameter. An estimator can be treated as a guess of the true value θ_{tr} of the parameter θ. It is expected that estimates are close to the true value θ_{tr}. However, since an estimator is a random variable and it is characterised itself by its probability distribution, one cannot say with certainty that an estimate is close to the true value of a parameter of the distribution. It is only possible to hope that the central region of the distribution of the estimator is close to the true value of the parameter. To express this hope in a mathematical way, the concept of unbiased estimators is introduced. The properties of estimators will be considered below.

2.3.5 Desirable properties of estimators.

For any given parameter, different estimators are possible. Hence, it is generally accepted that estimators have to satisfy the following main desirable properties: an estimator has to be unbiased, consistent and efficient.

Let us consider a statistic of size N. An estimator is said to be unbiased if

$E[\theta]_N = \theta_{tr}$ for any size N.

Here E means the expectation of a variable. Roughly speaking, the above definition means that the distribution mean of the estimator is equal to the true value of the parameter for any size of the statistic. An estimator whose expectation is not equal to the true value is said to be biased.

An estimator is a consistent estimator of the parameter, if as sample size increases, the estimator gets closer and closer to the value of the parameter being estimated. In other words, if one has a sequence of values of the estimator as a function of the sample size, then as the size expands *ad infinitum*, this sequence converges in probability to the true value of the parameter being estimated. Otherwise the estimator is said to be inconsistent.

The term of efficient estimator is used when there exist two or more unbiased estimators of the parameter. For example, the sample mean and the sample median are both unbiased estimators of the distribution mean. For a given sample size N, it is possible to define the relative efficiency of one estimator with respect to another one as the ratio of their variances. Only in some cases an unbiased efficient estimator exists, that has the lowest variance among unbiased estimators. Since we will not consider more than one unbiased estimator for a parameter, the property of efficiency of estimators will not be discussed further.

Estimators are used in Chapters 3 and 4; see for example 3.4.1, 3.4.3, 3.4.4, 4.2 and 4.3.

2.3.6 Similarity measures.

It is known that clustering analysis is the organisation of a collection of records into clusters where the elements within a cluster have a certain degree of similarity, and hence the similarity is a fundamental concept for definition of a cluster (see, e.g. Jain et al., 1999). Any measure of the degree of closeness (likeness) is called similarity measure (Looney 1997). It is very common to calculate the similarity or dissimilarity between two features using a distance measure. In clustering analysis of numerical data sets, the similarity or dissimilarity between two feature vectors $\mathbf{X}_1 = (x_{1j},...,x_{1p})$ and $\mathbf{X}_2 = (x_{2j},...,x_{2p})$ is often calculated using a square distance measure. Indeed, it is very natural to use the Euclidean metric (distance) ρ_E (or L_2 metric)

$$\rho_E(\mathbf{X}_1,\mathbf{X}_2) = \|\mathbf{X}_1 - \mathbf{X}_2\|_2 = \left(\sum_{j=1}^{p}(x_{1j} - x_{2j})^2\right)^{1/2}$$

as a measure for continuous numerical features because this metric is in everyday use. For example, the most popular clustering algorithm for numerical data sets is the *k*-means algorithm that uses the Euclidean distance.

It is evident that the Euclidean distance is a particular case ($p_M = 2$) of the following Minkowski distance ρ_{p_M} (or L_p metric)

$$\rho_{p_M}(\mathbf{X}_1,\mathbf{X}_2) = \| \mathbf{X}_1 - \mathbf{X}_2 \|_{p_M} = \left(\sum_{j=1}^{p}|x_{1j} - x_{2j}|^{p_M}\right)^{1/p_M}$$

where p_M is a positive number, $1 \le p_M < +\infty$.

Another particular case of the Minkowski distance is the city block (Manhattan) distance (or L_1 metric)

$$\rho_1(\mathbf{X}_1, \mathbf{X}_2) = \| \mathbf{X}_1 - \mathbf{X}_2 \|_1 = \sum_{j=1}^{p} |x_{1j} - x_{2j}|$$

The Tchebysheff (Chebyshev) or maximum norm metric. It gives the maximum of absolute difference between the feature vectors.

$$\rho_{max}(\mathbf{X}_1, \mathbf{X}_2) = \| \mathbf{X}_1 - \mathbf{X}_2 \|_{max} = \max_j \left| x_{1j} - x_{2j} \right|.$$

This metric can be also obtained from the Minkowski distance if the following limit is taken $p_M \to \infty$.

One can see that other distances like the Hamming, Mahalanobis, Hausdorff and so on, are also used in clustering analysis. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. This distance can be treated as a particular case of the city block (Manhattan) distance when all features are binary (Jain and Dubes, 1988). The Mahalanobis distance is based on correlations between variables and it is used mainly for solving supervised learning problems. A non-formal explanation of the Hausdorff distance is the following: according to this distance two sets are close to each other if every point of either set is close to some point of another set.

Each metric imposes its own geometry. The Euclidean distance leads to spherical shapes of equidistant regions. Points with a constant Mahalanobis distance to the centre are located on a hyperellipsoid that envelops the centre of the object points (Varmuza and Filzmoser, 2009). The Hamming distance imposes diamond-like geometry, while the Tchebysheff distance forms hyper squares (Cios et al., 2007).

It is claimed (Berkhin, 2002) that lower values of the power p_M of the usual Minkowski distance correspond to more robust estimations in applications to numerical data (therefore, less affected by outliers).

It is more difficult to introduce similarity measures for categorical data. Clustering mixed (numeric and categorical) data is a rather difficult problem. Indeed, when all attributes are of the same kind then the inter- and intra-cluster similarity can be defined according to one similarity measure between records, while for mixed data usually one needs to employ two different similarity measures.

2.3.7 Proximity and similarity indices.

Let us consider a finite set of observations $u_i \in U$. The index of similarity $S(u_i, u_j)$ is a real valued function defined on $U \times U$ that satisfies the following conditions:

Non-negativity:

$S(u_i, u_j) \geq 0$ for any $u_i, u_j \in U$.

Normalisation (Identity of indiscernibles):

$S(u_i, u_i) = 1$ for any $u_i \in U$.

Symmetry:

$S(u_i, u_j) = S(u_j, u_i)$ for any $u_i, u_j \in U$.

Contrary to distances that are normally used in application to numerical data , the indices of similarity are often applied to all kinds of variables, including categorical variables (Duran and Odell, 1974, Giudici, 2003).

Goodall (1966) (see also Jain and Dubes, 1988) proposed an index of similarity using probabilistic approach. It was suggested that the index has a uniform distribution when the data are random. Gower's similarity coefficient (Gower, 1971) is another popular measure of proximity for mixed data types.

Using the above mentioned similarity coefficients and indices, and other dissimilarity measures (Gowda and Diday, 1991), the standard hierarchical clustering methods can handle data with numerical and categorical values. However, the quadratic computational cost makes them unacceptable for clustering large data sets (Anderberg, 1973, Jain and Dubes, 1988).

The proximity index $d(u_i,u_j)$ between two observations $u_i,u_j \in U$ is a real valued function defined on $U \times U$ that satisfies the following conditions:

The inequality that is used to measure similarity:

$d(u_i,u_i) \geq \max_j d(u_i,u_j)$ for any $u_i,u_j \in U$.

Non-negativity:

$d(u_i,u_j) \geq 0$ for any $u_i,u_j \in U$.

Symmetry:

$d(u_i,u_j) = d(u_j,u_i)$ for any $u_i,u_j \in U$.

If identity of indiscernibles is used to measure proximity between identical observations: $d(u_i,u_i) = 1$ for any $u_i \in U$ then $0 \leq d(u_i,u_j) \leq 1$ for any observations $u_i,u_j \in U$. Note that this definition of the proximity index is slightly different from the definition given by Jain and Dubes (1988).

The proximity index $d(u_i,u_j)$ between two categorical variables $u_i,u_j \in U$ can be used as indicator of mismatch or as a distance function in the categorical space. In this case the index takes just two values

$$d(u_i,u_j)=\begin{cases}1, & \text{if } u_i=u_j\\ 0, & \text{if } u_i\neq u_j\end{cases}$$

for any $u_i,u_j\in U$. Huang (1998) used the notation $\delta(u_i,u_j)$ as the indicator of mismatch (simple matching measure)

$$\delta(u_i,u_j)=\begin{cases}0, & \text{if } u_i=u_j\\ 1, & \text{if } u_i\neq u_j\end{cases}$$

However, the above notation can be confused with the common notation of the Kronecker delta δ_{ij}, while the latter delta $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ if $i\neq j$. Therefore, we will use the notation ω for matching measure. Hence, the distance between two categorical feature vectors $\mathbf{Y}_1=(y_{11},\ldots,y_{1l})$ and $\mathbf{Y}_2=(y_{21},\ldots,y_{2l})$ is defined as:

$$\rho_{cat}(\mathbf{Y}_1,\mathbf{Y}_2)=\omega(y_{11},y_{21})+\ldots+\omega(y_{1l},y_{2l})$$

where

$$\omega(y_{1j},y_{2j})=\begin{cases}0 & \text{for } y_{1j}=y_{2j}\\ 1 & \text{for } y_{1j}\neq y_{2j}\end{cases}.$$

2.4 Minkowski distance or L^p space

The **Minkowski distance** of order p, based on the Minkowski norm (L^p) is defined as:

$$\rho_p(\mathbf{x}_1,\mathbf{x}_2) = \| \mathbf{x}_1 - \mathbf{x}_2 \|_p = \left(\sum_{j=1}^{n} |x_{1j} - x_{2j}|^p \right)^{1/p}$$

where $\mathbf{x}_1 = (x_{11},\ldots,x_{1n})$ and $\mathbf{x}_2 = (x_{21},\ldots,x_{2n})$.

p does not need to be an integer, but it cannot be less than 1, because otherwise the triangle inequality does not hold.

Further we will consider a data set represented as a matrix of size $N \times (p+l)$. Here N is the number of records, p is the number of numerical attributes and l is the number of categorical attributes. Because we consider p as the number of numerical attributes, we have to change n in the above mentioned definition of the Minkowski norm for p , and also we will use p_M instead of p as the Minkowski power and the formula will be:

$$\rho_{p_M}(\mathbf{x}_1,\mathbf{x}_2) = \| \mathbf{x}_1 - \mathbf{x}_2 \|_{p_M} = \left(\sum_{j=1}^{p} |x_{1j} - x_{2j}|^{p_M} \right)^{1/p_M}$$

As we mentioned before, a norm satisfies the triangle inequality for any two elements x and y.

$$\|x + y\| \le \|x\| + \|y\|$$

The triangle inequality in L^p spaces is:

$$\|f + g\|_{p_M} \le \|f\|_{p_M} + \|g\|_{p_M}$$

where f and g are elements of $\mathrm{L}^p(S)$.

2.5 Typical steps in clustering activity

Cluster analysis is the organisation of a collection of records into clusters based on similarity (Jain and Dubes, 1988). Typical clustering activity involves the following steps (Jain and Dubes, 1988):

(a) **Representation of records** (optionally including feature extraction and/or selection): record representation refers to the number of classes, the number of records, and the number, type, and scale of the features available to the clustering algorithm. Some of this information may not be controllable by the researcher. We should also try to avoid correlated variables that could lower the performance of some methods. Feature selection is the process of identifying the most effective subset of the original features to use in clustering. Feature extraction is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to obtain an appropriate set of features to use in clustering.

(b) **Definition of a record proximity measure appropriate to the data domain.** Record proximity is usually measured by a distance function defined on pairs of data points.

(c) **Clustering or grouping.** Clustering as we will see later can be divided in three main categories: objective function-based (partition-based), hierarchical clustering and model-based clustering.

(d) **Data abstraction** (if needed). This is the process of extracting a simple and compact representation of a data set.

(e) **Assessment of output** (if needed). Cluster validity analysis is the assessment of a clustering procedure's output.

Feature selection is very important in a number of new applications with very large input spaces. This is because these applications critically need space dimensionality reduction for efficiency and efficacy of the predictors. In particular, these applications include bioinformatics (DNA microarrays, mass-spectrometric data, etc.), combinatorial chemistry (e.g. high throughput screening of drug candidates), text processing (e.g. spam filtering), decision making (e.g. oil drilling), pattern recognition (e.g. handwriting recognition), speech processing, and vision.

There are various methods for supervised feature selections (see, e.g. Cios et al. (2007)). For example, minimum redundancy feature selection, filtering approach of feature selection, wrapper approach of feature selection. The supervised methods assume that class label information for each data record is given. For unsupervised feature selection several methods have been developed.

There are many transformations for feature extraction; some of these methods do not alter the space dimensionality (e.g. normalisation), while others enlarge it (non-linear

expansions, feature discretisation), reduce it (space embedding methods) or can act in either direction (extraction of local features). In this thesis we will study normalisation methods.

2.6 Main types of clustering

Cluster analysis or clustering can be divided in three main types: objective function-based clustering (partition-based), hierarchical clustering and model-based clustering (Estivill-Castro, 2002, Cios et al., 2007).

2.6.1 Hierarchical clustering.

This kind of clustering is based on creating a hierarchical decomposition of the set of data points using some criterion or models. However, it is based not on continuous mathematical models like probability distributions, but on discrete, structural models. As a result, hierarchical clustering produces a representation of data in a form of a graph (dendrogram). There are two different approaches: the bottom-up, also known as agglomerative approach, and top-down also known as divisive approach (Cios et al., 2007). The former approach treats each record as a single-element cluster and then successively merges the closest clusters. At each pass, the two closest clusters are merged. The process repeats until the current number of clusters is equal to k, or a predefined threshold value is reached. The later approach works in the opposite direction.

The entire set is initially treated as a single cluster, and it is kept splitting into smaller clusters. Almost all hierarchical clustering algorithms are agglomerative, as divisive methods present a huge computational task. This kind of clustering will not be used in the thesis.

2.6.2 Model-based clustering.

Let us describe the model-based clustering following Cios et al. (2007). In model-based clustering methods, each observation is obtained from a mixture of c sources of data with given prior probabilities $p_1, p_2, \ldots, p_c$, component-specific conditional probability density function and its parameters. It is assumed in this kind of clustering that there is a certain probability model of the data, i.e. there is a set of equations which describes the behaviour of the data under consideration in terms of random variables and the associated probability distributions of the variables. These probability distributions define the clusters. Each object is generated by one and only one of these distributions; hence belongs to one and only one cluster.

The parameters of the model have to be estimated. A popular method used for fitting a statistical model to data is the maximum likelihood estimation. This method picks the values of the model parameters that maximize the probability (likelihood) of the sample data, i.e. these values make the data "more likely" than any other values would make them. The maximum likelihood approach is used under assumption that each data item

was independently drawn form the statistically distributed data mixture. The principle is used to find the distribution parameters and, hence, one may obtain the proportions in the clusters, their location and scatter, and for each individual item, the probability that it belongs to the i-th cluster.

Since it is assumed that the data are a result of a mixture of c sources of data, the above structure is called mixture density model. These sources might be considered as clusters with given prior probabilities $p_1, p_2, \ldots, p_c$ that are also called the mixing parameters. Each component of this mixture is described by some conditional probability density function, $p(\mathbf{x}|\theta_i)$ characterised by a vector of parameters θ_i. Under these assumptions, the model is additive and comes in the form of mixture densities:

$$p(x|\theta_1, \theta_2, \ldots, \theta_c) = \sum_{i=1}^{c} p(x|\theta_i) p_i .$$

To build the model, one has to estimate the parameters of the contributing probability density functions. To do so we have to assume that $p(\mathbf{x}, \theta)$ is identifiable which means that if $\theta \neq \theta'$ then there exists a $\mathbf{x}$ such that $p(\mathbf{x}|\theta) \neq p(\mathbf{x}|\theta')$. As it has been mentioned above, the standard approach used to discover the clusters is to carry out maximum likelihood estimation. Most of the work in this area has assumed that the individual components of the mixture density are Gaussian.

The induction principle of the Maximum Likelihood approach says "choose the model that maximizes the probability of the data being generated by such model" (Kalbfleisch, 1985). This iterative algorithm converges to local optima and is the well-known expectation maximisation (EM) method (Dempster et al., 1977). EM algorithms do not require the specification of distance measures and therefore, it admits both categorical and continuous attributes.

Although the statistical approaches will be intensively used in the thesis, these approaches will be applied to objective function-based clustering. Thus, the above described model-based clustering will not be used in the thesis.

2.6.3 Objective function-based clustering.

A very general category of clustering is concerned with building partitions (clusters) of data sets on the basis of some performance index known also as an objective function.

Here we need to distinguish hard and fuzzy cluster methods. It is known that a partitioning method constructs k groups. If these groups together satisfy the following requirements of a partition (Kaufman and. Rousseeuw, 1990):

1. each group must contain at least one object, and
2. each object must belong to exactly one group,

then this is hard clustering. Thus, in hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering, data elements can belong to more than one cluster, and associated with each element is a set of membership levels (Jain et al., 1999).

It is known that an objective function, known also as cost function, is a function associated with an optimisation problem where the best element from some set of available alternatives is chosen to minimize or maximize the function. The value of this function determines how good the chosen solution is. There are various clustering algorithms for objective function-based clustering because it is practically unfeasible to find a global optimum for the objective function by considering all possible combinations of elements (exhaustive search). Indeed, to present k clusters of the total n elements, we need to consider all $N(n,k)$ (Stirling's number) possible partitions:

$$N(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-i}\binom{k}{i}i^N$$

where the notation

$$\binom{k}{i}$$

denotes the binomimal coefficient

$$\binom{k}{i} = \frac{k!}{i!(k-i)!}.$$

$N(n,k)$ is one of Stirling's numbers (see, e.g. Jensen, 1969). With increasing n this soon becomes intractable, so that inevitably, partitioning algorithms do not consider all partitions and can normally find only local optima.

Objective function-based clustering means: there is an objective function whose value depends on the chosen partition and how small this value is determines how good the particular clustering is. The main design challenge of clustering lies in formulating an objective function that is capable of reflecting the nature of the problem so that its minimisation reveals a meaningful structure in the data set (Pedrycz, 2005).

2.6.4 Hybrids of supervised and unsupervised learning.

There are also other kinds of clustering that can be considered as hybrids of ideas of supervised and unsupervised learning. In particular, conceptual clustering algorithms, semi-supervised learning algorithms like ISODATA, and analysis of effectiveness of clustering algorithms using labelled data sets.

Conceptual clustering algorithms consist of two tasks: (i) to find clusters in a given data set, and (ii) to produce a conceptual description for each found cluster (Cios et al., 2007). The former task is an unsupervised machine learning task, while the latter task is a characterisation problem that belongs to supervised machine learning tasks. The

conceptual clustering algorithms may cluster data with categorical values (Fisher, 1987, Lebowitz, 1987, Michalski and Stepp, 1983). The ability to produce conceptual descriptions of clusters is important to data mining because the conceptual descriptions provide assistance in interpreting clustering results. The conceptual clustering algorithms are based on a search for objects which carry the same or similar concepts. Therefore, their efficiency relies on good search strategies. For problems in data mining, which often involve many concepts and very large object spaces, the concept-based search methods can become a potential handicap for these algorithms to deal with extremely large data sets.

Quite often the advantage of labelled data, whose labels are extracted by the use of association rules as the supervised information, is combined with the use of unsupervised learning methods, like in the algorithm ISODATA to establish semi-supervised learning algorithms. ISODATA: Iterative Self-Organizing Data Analysis Techniques Algorithm may be considered as a variation of the k -means clustering algorithm. It allows the number of clusters to be automatically adjusted by splitting clusters with large standard deviations or merging similar clusters. Since it uses the training set of the data it is a hybrid of supervised and unsupervised methods.

We need to note that a hybrid of ideas of supervised and unsupervised learning is also used to check the effectiveness of clustering algorithms. For example, Liu and Huang (2003) considered a variant of a genetic algorithm and evaluated the fitness of each

chromosome with a combination of fuzzy within cluster variance of unlabelled data and misclassification error of labelled data.

It is known that optimisation techniques use various methods, strategies and algorithms. Evolutionary approaches belong to these techniques. In particular, they include genetic algorithms (GA) and Swarm intelligence (SI) that mimic heuristically biological evolution.

2.6.4.1 Genetic algorithms (Evolutionary approaches for clustering)

Genetic algorithms mimic the principle of the survival of the fittest individual in the process of selection. GAs deal with a population of abstract representations (called chromosomes or the genotype) of candidate solutions (called individuals, creatures, or phenotypes). The space of all candidate solutions is called the search space. GAs make use of evolutionary operators and a population of solutions to obtain the globally optimal partition of the data. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population based on their fitness (traditionally the objective function in GA applications is called the fitness function), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced,

or a satisfactory fitness level has been reached for the population (see e.g. Michalewicz 1996).

Pseudo-code of a genetic algorithm

1. Choose initial population
2. Evaluate the fitness of each individual in the population
3. Repeat:
   (i) Select best-ranking individuals to reproduce
   (ii) Apply genetic operations (crossover and mutation) and give birth to offspring
   (iii) Evaluate the individual fitnesses of the offspring
   (iv) Replace worst ranked part of population with offspring
4. Until terminating condition is met.

In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. This means that it does not "know how" to sacrifice short-term fitness to gain longer-term fitness. The GA search tries to balance two objectives: utilising the best solutions and exploring the search space. GAs are discussed in details by many authors (see, e.g. Michalewicz, 1996, and Mitchell, 1996). Some specific features of GA based clustering are discussed in Chapter 5.

2.6.4.2 Swarm intelligence (Evolutionary approaches for clustering)

Similarly to GA, Swarm intelligence (SI) is a type of artificial intelligence that mimics the collective behaviour of animals. For example, SI includes the Ant Colony Optimization (Dorigo et al., 1996) and Bees Algorithm (Pham et al. 2006b). The latter is a new technique that was introduced to mimics nature's evolutionary principles that drive the search of bees towards an optimal solution. In application to problems of optimisation, a bee means a point of the domain (the search space) of the objective function, while the fitness of the bee means the value of the objective function at this point. It was shown (Pham et al. 2006b) that using the BA for some optimisation problems is more effective than using the GA based techniques (Goldberg, 1989). The main ideas of the Bees algorithm (Pham et al. 2006b) are discussed in detail in Chapter 5.

2.7 Objective - function based clustering algorithms and its applications

2.7.1 Objective - function based clustering for mixed data sets.

Data analysis with mixed data may follow three main strategies: Variables partitioning; Variables converting, and Compatibility measures (Anderberg, 1973, Gibert and Cortes, 1997). Variables partitioning consists on partitioning the variables upon their type, then

reducing the analysis to the dominant type (determined owing to the group with a greater number of variables, or the group containing the more relevant variables, or the background knowledge on the domain and so on). Variables converting method converts all the variables to a unique type, trying to conserve as much original information as possible. It does not necessarily produce meaningful results in the case of categorical domains beign not ordered. This method is traditionally used in mathematical statistics (Neal and Hinton 1999, Pregibon and Elder, 1996). Finally, compatibility measures method consists on the use of compatible measures which cover any combination of variable types, making a homogeneous treatment of all the variables. Its idea is to allow clustering on a domain simultaneously described by numerical and categorical variables without transforming the variables themselves. The last method was used by many authors, e.g. Ralambondrainy (1995), Gupta et al. (1999) and Huang (1998), for clustering records of mixed data. In fact, they extended the distance-based k-means algorithm to handle categorical data in addition to numerical data.

2.7.2 The k-means, k-modes and k-prototypes algorithms

The k-means, k-modes and k-prototypes algorithms are based on the most intuitive and frequently used objective function - the squared error criterion. The function tends to work well with isolated and compact clusters. The induction principle of the k-means based approaches says "pick the model (set of k centres) that minimizes the total squared error".

Step1. (Initialization). Having determined the number of groups, k prototypes (sometimes called seeds), are defined. The seeds constitute the centres (measures of position, usually means) of the clusters in the initial partition.

Step2. (Assignment of points to clusters) Each data point is assigned to the cluster with the closest centre. For each element of the data set, the distances are calculated between the element and the prototype of the cluster to which it has been assigned.

Step 3. (Update of all cluster centres). Recalculate the centres of the clusters. The objective function is calculated using these distances and it has to have a minimum value, otherwise the elements will be moved to other clusters.

Step 4. (Stopping criterion). If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of records to new cluster centres, or minimal decrease in squared error.

The k-means clustering method (Anderberg, 1973, MacQueen, 1967) is efficient for processing large data sets. Therefore, it is best suited for data mining. However, the k-means algorithm only works on numeric data, because it minimises a cost function by changing the means of clusters. Hence, one cannot use it in applications to categorical data or mixed data.

To deal with mixed data, the distance metric may be redefined as a sum of two measures, one for the categorical attributes and one for the numerical attributes. The hard part of combining metrics like this is that an appropriate weighting of the measures needs to be derived for the overall measure to be useful. Huang (1997, 1998) introduced two extensions of the k-means algorithm, namely the algorithms, called k-modes and k-prototypes, respectively. The former algorithm was targeted to deal with categorical attributes, while the latter was introduced to cluster large data sets with mixed numerical and categorical values. To deal with categorical data Huang replaced means of clusters used in the k-means algorithm by modes, and used a frequency-based method to update modes in the clustering process to minimise the clustering objective function (cost function) (Huang 1998). In the k-prototypes algorithm he defined a "dissimilarity measure" that takes into account both numerical and categorical attributes. In fact, he considered a metric ρ_H, where ρ_H^2 is the sum of the square of the Euclidean numerical metric and a weighted categorical metric (the matching dissimilarity measure).

The advantages and drawbacks of these algorithms may be described as follows. The k-means algorithm has been widely adopted as a general purpose algorithm because it is easy to implement. It also has practically no limitation on the size of data sets because its time complexity (the time complexity of an algorithm refers to the time it takes to run) is $O(n)$, where n is the number of data points. It also does not explicitly restrict the dimensionality of the data. Disadvantages of the algorithms are that the algorithms require the clusters to be spherical, that the data be free of noise (those conditions hardly

occur in practical situations) and that the algorithms are sensitive to the selection of the initial partition.

2.7.3 Most recent applications of clustering for categorical and mixed data sets

It is accepted in the overwhelming majority of papers devoted to OF-based clustering that the k -means algorithm performs very well in application to numerical data. Currently many authors see the main problem in OF-based cluster analysis in development of new algorithms for clustering categorical and mixed data.

Peters and Zaki (2004) introduced the Click algorithm, which searches clusters in categorical data sets. They treat informally clusters as especially dense interval regions within a data set. A region can be considered dense if the actual support is higher than the expected support of a given interval region. It was claimed that the Click algorithm outperforms previous approaches by a factor of two to three. However, Andreopoulos et al. (2009) have noted that there is a problem related to applying density-based clustering to categorical biomedical data. In their treatment a categorical dataset with l attributes is viewed as an l-dimensional "cube", offering a spatial density basis for clustering. Since the "cube" of attribute values has no ordering defined, the search for dense subspaces is rather slow. So they employed the Hamming distance and introduced the HIERDENC algorithm for "hierarchical density based clustering of categorical data". Applications of

the algorithm results in layered clusters where a central subspace often has a higher density.

As it has been mentioned above, Huang (1997, 1998) introduced the k-modes and k-prototypes algorithms as extensions of the k-means algorithm. These algorithms are very popular. Zhang et al. (2006) claimed that their statistical procedure for clustering categorical data based on Hamming distance vectors outperforms the k-modes algorithm. However, the method was not applied to mixed data sets.

Ahmad and Dey (2007) presented the "k-mean clustering algorithm for mixed numeric and categorical data". As an example, they considered a categorical attribute A_i that may have two values a and b. In order to find the distance between a and b, they considered the overall distribution of a and b in the data set along with their co-occurrence with values of other attributes. For the given data set, they considered another categorical attribute A_j and denote by w a subset of values of A_j and by z the complementary set of values occurring for this attribute. Then they denoted by $P_i(w/a)$ the conditional probability that an element having value a for A_i, has a value belonging to w for A_j and $P_i(z/b)$ denotes the conditional probability that an element having value b for A_i, has a value belonging to z for A_j. According to their definition, distance between the pair of values a and b of A_i with respect to attribute A_j and a particular subset **w** is defined as follows $\delta_w^i(a,b) = P_i(w/a) + P_i(z/b)$. This definition is

not symmetric with respect to a and b. Hence it is not a metric. Besides, one has to note that the paper contains some undefined items and this makes it practically impossible to use the model for practical realisation.

A number of very interesting approaches were derived from ideas introduced by Bezdek and his co-workers (see, e.g. Bobrowski and Bezdek, 1991; Hathaway and Bezdek, 1995). Bobrowski and Bezdek (1991) introduced an extension of the hard and fuzzy c-means clustering algorithms to the cases of l_1 and l_∞ norms. Their approach was developed further by Miyamoto and Agusta (1995, 1998), Hathaway et al. (2000), Takata et al. (2001), Koga et al. (2001), Endo et al. (2006) and others. In these papers it was introduced a very promising idea to generalise the standard $\sum \rho_2^2$ objective function to the functions $\sum \rho_{p_M}^{p_M}$, where ρ_{p_M} is the Mikowski distance and p_M is the power of the Minkowski norm. However, these generalisations were applied only to fuzzy clustering algorithms. In Chapter 4 this idea is extended to the case of hard clustering and applied to mixed data sets.

Chan et al (2004) and Huang et al. (2005) introduced a weighting k-means type clustering algorithm that can calculate attribute weights automatically. The algorithm calculates a new weight for each attribute based on the variance of the within cluster distances. The algorithm was applied to both synthetic and real data. It was claimed that the algorithm outperformed the standard k-means type algorithms in recovering clusters in data. To estimate the accuracy of clustering both the clustering accuracy and the Rand

index were employed. One has to note that clustering was performed without normalisation of variables, while it is known that the raw data need to be normalised (Aksoy and Haralick, 2001; Larose, 2005).

Normalisation of attributes was discussed in a number of papers (see, e.g., Aksoy and Haralick, 2001; Hastie et al., 2001; Larose, 2005; Pham et al., 2006a). It was realised that normalisation should give all attributes equal influence on characterising overall dissimilarity between pairs of objects (Hastie et al., 2001; Pham et al., 2006a). However, Hastie et al. (2001) after introducing a correct interpretation of the normalisation procedure, gave an example where standardisation obscured the two well-separated groups. They argued that *variables that are more relevant in separating the groups should be assigned a higher influence in defining object dissimilarity. Giving all attributes equal influence in this case will tend to obscure the groups to the point where a clustering algorithm cannot uncover them.* In fact, this argument is very similar to the above arguments of Chan et al (2004) and Huang et al. (2005) who applied weighting of attributes without normalisation. We agree that in particular examples clustering without normalisation may give good results. However, this is the case of luck because this means that by chance the attributes have proper weights. We believe that it is too naïve to relay on luck in unsupervised learning when there is no a priory information about importance of attributes for clustering. We agree that if one knows a priory that some attributes have bigger contributions to similarity measures than the rest of the attributes then this can be taken into account by appropriate weighting of the attributes. However, it looks quite natural to apply the normalisation procedure first and only after the means

of contributions of all attributes have been equalised, then to apply the weighting procedure to more important attributes.

2.8 Summary

The Chapter has recalled a number of notations and definitions of concepts related to clustering, similarity measures for numerical, categorical and mixed data sets, objective functions, and statistical estimators. The Chapter ends with a literature review of the most recent applications of objective - function based clustering for mixed data sets.

Further we deal only with objective function-based clustering of flat file data sets where a data set can be represented as a matrix of size $N \times (p+l)$. Here N is the number of records, p is the number of numerical attributes and l is the number of categorical attributes.

Chapter 3

Clustering mixed data sets (Euclidean metric) by using the *k*-prototypes algorithm

In this Chapter a unified statistical approach to both numerical and categorical attributes is applied in order to normalise the feature vectors for mixed data sets. The proposed approach is extended to the case of mixed metrics, i.e. when different metrics are used for numerical and categorical data. The most common case of metrics, namely the Euclidean metric is used as a measure for continuous numerical features, while the matching dissimilarity measure is used to deal with categorical attributes. Normalised metrics are introduced such that the average contributions of all attributes to the measures are equal to each other from statistical point of view. Advantages of the introduced normalised metrics are demonstrated on examples of their applications to various data sets. Methods for comparing the accuracy of the clustering algorithms are discussed in detail and explained on examples. Results on benchmark data sets are presented together with a comparison with other approaches.

3.1 Background

It has been defined in Chapter 2 that Data Mining (DM) is a process of extracting relations and patterns from data and hence DM is a tool for transforming raw collections of data into information. As it was noted by Larose (2005), "*In the real world, dirty data sets need cleaning; raw data need to be normalized; outliers need to be checked*". The normalisation procedure relies on the use of various mathematical concepts, and hence, it is important to develop appropriate mathematical tools for this procedure.

We call data set a collection of objects described by the same features. As we have seen in Chapter 2, a data set can be represented as a matrix of size $N \times (p+l)$ where N is the number of records, p is the number of numerical attributes and l is the number of categorical attributes. The i-th row of the matrix represents the i-th record of the data set and it is a vector $(x_{i1},\ldots,x_{ip},y_{i1},\ldots,y_{il})$. The values $x_{i1},\ldots,x_{ip}$ are numerical while the values $y_{i1},\ldots,y_{il}$ are categorical. In clustering analysis of numerical data sets, it is very common to calculate the similarity or dissimilarity between two feature vectors $\mathbf{x}_1 = (x_{11},\ldots,x_{1p})$ and $\mathbf{x}_2 = (x_{21},\ldots,x_{2p})$ using a square distance measure. Indeed, it is very natural to use the Euclidean metric ρ_E (or L_2 metric)

$$\rho_E(\mathbf{x}_1,\mathbf{x}_2) = \| \mathbf{x}_1 - \mathbf{x}_2 \|_2 = \left(\sum_{j=1}^{p} (x_{1j} - x_{2j})^2 \right)^{1/2} \tag{3.1}$$

as a measure for continuous numerical features because this metric is in everyday use. In addition, the *k*-means algorithm uses the Euclidean metric (3.1) to measure distances between records and combines the use of the metric with employment of the objective function that is defined as the sum of squares of $\rho_E(\mathbf{x}_i, \mathbf{x}_j)$. This combination has some specific mathematical features that will be discussed in Section 3.2, and it gives some advantages to the *k*-means algorithm.

For categorical data and for mixed (numeric and categorical) data, there is no such a natural similarity measure as the Euclidean metric. Therefore, two different similarity measures are often combined for clustering of mixed data (see, e.g. Gibert and Cortes, 1997, Huang 1997). One of possible combinations is the combination of the most common cases of metrics, namely the Euclidean metric that is used to measure distances between continuous numerical attributes, and the matching dissimilarity measure that is used to measure distances between categorical attributes. This combination is used in the *k*-prototypes algorithm that is the most popular algorithm for clustering mixed data sets (Huang 1998). The same combination of metrics is considered in this Chapter. The application of the proposed procedure to the general case of Minkowski metrics is discussed in Chapter 4.

In spite of the importance of data normalisation, there are only few papers especially devoted to normalisation methods for data sets. Milligan and Cooper (1988) discussed various normalisation methods that have to be applied to numerical data before conducting a cluster analysis. Aksoy and Haralick (2001) gave a review of

normalisation techniques that may be applied to numerical data sets. The goal of the normalisation procedures reviewed was the normalisation of each feature component to the [0, 1] range. However, after this kind of data set normalisation, the average contributions of all features to the similarity measure may be not equal to each other.

The idea to use a weighted Euclidean distance that may take into account the scatter of samples within a cluster (see e.g. Chen, 1973), was recently generalised to Minkowski distance by Pham et al. (2006b). Mirkin (1996, 1997, 1998) discussed normalisation of mixed features based on their contributions to the quadratic data scatter. Mirkin (1998) stated that methods for analysis of data in mixed feature space are still an issue.

In this thesis we argue that the average contribution of the j-th feature component to the total measure has to be equal to its mean and therefore, the goal of a normalisation procedure is the equalisation of the attribute contributions. In this chapter a unified statistical approach is applied to both numerical and categorical attributes in order to normalise the feature vectors for mixed data sets. After the proposed normalisation, the means of all dimensionless attributes will be the same and hence, contributions of the features to similarity measures are approximately equalised.

This chapter is organised as follows: Section 3.2 discusses some specific features of k-means. Section 3.3 presents a description of commonly used normalisation techniques. In Section 3.4, the proposed statistical approach to normalisation of

feature vectors is presented. Methods of estimation of accuracy of the clustering algorithms are discussed in detail in Section 3.5. Numerical results on benchmark data sets are presented in Section 3.6 and the conclusion of this Chapter is given in Section 3.7.

3.2 Some specific features of the k-means algorithm

Sometimes it is argued that the k -means algorithm (MacQueen 1967) is so successful in application to numerical data just because it involves Euclidean distances and the corresponding spherical geometry (see e.g., Cios et al., 2007). Whilst those are good reasons, there is another more important argument to explain the popularity of the k -means algorithm. Let us discuss some known special properties of the k -means algorithm for partition of data set into k clusters. It uses as the objective function J not the sum of Euclidean distances but the sum of squares of the metric. If the above explanation reflected all specific properties of the algorithm then one were able to use as the objective function the sum of Euclidean distances with the same success. Thus, the k-means algorithm minimises the objective function J

$$J = \sum_{m=1}^{k} \sum_{i=1}^{N} u_{im} \rho_E^2(\mathbf{X}_i, \mathbf{Q}_m), \tag{3.2}$$

$$u_{im} \in \{0,1\}, \ 1 \le i \le N, \ 1 \le m \le k,$$

$$\sum_{m=1}^{k} u_{im} = 1, \forall i, \ \text{and} \ \sum_{i=1}^{N} u_{im} > 0 \ \ \forall m. \tag{3.3}$$

where u_{im} is an element of the partition matrix. The condition $u_{im} = 1$ means that the record $\mathbf{X}_i$ is assigned to cluster m with prototype (centre) $\mathbf{Q}_m$.

Let us write (3.2) as

$$J = \sum_{m=1}^{k} \sum_{i=1}^{N} u_{im} \sum_{j=1}^{p} \left(\mathbf{X}_{ij} - \mathbf{Q}_{mj} \right)^2 \tag{3.4}$$

Using (3.3), one can rewrite (3.4) in the following form

$$J = \sum_{m=1}^{k} \sum_{i \in C_m} \rho_E^2 (\mathbf{X}_i, \mathbf{Q}_m) \tag{3.5}$$

The second sum is taken by elements that belong to the cluster C_m. The objective function that is calculated using these distances has to have a minimum value. Hence, the problem is

$$J = \sum_{m=1}^{k} \sum_{i \in C_m} \sum_{j=1}^{p} \left(\mathbf{X}_{ij} - \mathbf{Q}_{mj} \right)^2 \to \min \tag{3.6}$$

However, (3.6) has the minimum value if for any m we have

$$\sum_{i \in C_m} \sum_{j=1}^{p} \left(\mathbf{X}_{ij} - \mathbf{Q}_{mj} \right)^2 \to \min$$

or changing the order of summation, we can write

$$\sum_{j=1}^{p}\sum_{i\in C_m}\left(\mathbf{X}_{ij}-\mathbf{Q}_{mj}\right)^2 \to \min$$

Let us write the condition of an extremum for a smooth function for any fixed attribute j. Hence, we obtain

$$\frac{\partial}{\partial Q_{mj}}\sum_{i\in C_m}\left(\mathbf{X}_{ij}-\mathbf{Q}_{mj}\right)^2 = -2\sum_{i\in C_m}\left(\mathbf{X}_{ij}-\mathbf{Q}_{mj}\right)=0$$

Since the attribute number j is fixed, all $\mathbf{Q}_{mj}$ in the above expression are the same because we take the sum within the cluster C_m. Hence, we can represent it as

$$\sum_{i\in C_m}\mathbf{X}_{ij}-|C_m|\mathbf{Q}_{mj}=0$$

where $|C_m|$ denotes the number of elements in the cluster C_m.

Eventually, we obtain the expression for recalculating new centres of the clusters

$$\mathbf{Q}_{mj}=\frac{1}{|C_m|}\sum_{i\in C_m}\mathbf{X}_{ij} \qquad (3.7)$$

Note that if one writes the condition of an extremum for another objective function, e.g. the sum of the Minkowski distances including the sum of Euclidean distances then after taking the derivative, one does not obtain as simple expression as (3.7).

The possibility to derive the very simple expression for recalculating new cluster centres is the main reason of the popularity of the k-means algorithm.

As it has been mentioned in Chapter 2, Huang (1997, 1998) introduced two extensions of the k-means algorithm, namely the algorithms, called k-modes and k-prototypes, respectively. The former algorithm was targeted to deal with categorical attributes, while the latter was introduced to cluster large data sets with mixed numerical and categorical values. In this Chapter 3 the k-prototypes is used to cluster data sets with mixed numerical and categorical values.

3.3 Normalisation of feature vectors

As it has been mentioned in Chapter 2, normalisation is a particular kind of feature extraction method. Normalisation of data sets is widely used in a number of fields of machine intelligence. Sometimes the term standardisation is used as a synonym to normalisation. This kind of feature extraction is important because if the data is not normalised then the contribution of each feature to the similarity measure depends on the units of measurements and, therefore, the contribution of the features to the measure are scale dependent.

3.3.1 Normalisation of numerical data sets

A direct application of geometric measures (e.g. city block or Euclidean distances) to attributes with large ranges will implicitly assign larger contributions to the metrics than the application to attributes with small ranges. In addition, the attributes should be dimensionless; for example, we can not compare attributes in metres (m) with attributes in Newtons (N). Indeed, the numerical values of the ranges of dimensional attributes depend on the units of measurements and therefore, the choice of the units of measurements may greatly affect the results of clustering. If it is known a priory that some attributes are irrelevant to the problem under consideration then they can be removed from the feature vector.

In the general case of normalisation of data sets, when there is no a priory information about preferences of some attributes, one has to assume that all attributes are equally important. In this case, the distance or dissimilarity functions of clustering algorithms involve all attributes of the data set. As Chan et al. (2004) noted, this is applicable if all or most attributes are important to every cluster. However, clustering results become less accurate if a significant number of attributes are not important to some clusters. Hence, if all attributes are equally important to measure similarity between feature vectors then one should not use distance measures like the Euclidean distance (3.1) without normalisation of data (see, e.g. (Gibert and Cortes, 1997; Aksoy and Haralick, 2001). Further one need to apply normalisation not only to numerical attributes but also to categorical attributes.

New normalised metrics are introduced such that the average contributions of all attributes to the measures are equal to each other from statistical point of view. Although this idea has been recently discussed in the literature (Hastie et al., 2001), they said nothing about statistical consistency of the proposed estimators. In addition, they used biased estimators.

Min–Max Normalisation.

This approach normalises the data by dividing the attribute value x_{ij} by its range using scaling with a shift

$$x_{ij}^{*} = \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}} \tag{3.8}$$

Here x_{ij}^{*} is the normalised attribute value in the data set, $x_{max,j}$ and $x_{min,j}$ are the maximum and the minimum values of attribute A_j, respectively. This is the most cited method of normalising data sets. Sometimes it is referred to as Min–Max normalisation (Larose, 2005). Doherty et al. (2004) applied this kind of normalisation to the Minkowski metric.

Evidently, the results scaled by (3.8) do not depend on the original units of data measurements, and this linear scaling will transform the data to the range $[0,1]$. However, this normalisation procedure does not achieve equalisation of the attribute means. Hence, the application of the transformation (3.8) for normalisation of real world data sets and consequent clustering using either Euclidean or Minkowski norm,

do not give equal contributions of variables to the similarity measures because the means of the different normalised attributes are not necessary equal to each other.

Z-Score Standardisation.

This is a very popular normalisation technique that normalises the variables by taking the difference between its value and its mean value and scaling this difference by the standard deviation of the variable (Jain and Dubes, 1988; Larose, 2005)

$$x_{ij}^{\bullet} = \frac{x_{ij} - \bar{X}_j}{S_j(X)} \tag{3.9}$$

It will be shown below that this approach is consistent with our approach when the Euclidean metric is used.

For numerical datasets when the Euclidean metric is used, the most common normalisation procedure is the Z-score standardisation, i.e. to transform the attribute A_j^n to a random variable with zero mean and unit variance by

$$x_{ij}^{\bullet} = (x_{ij} - \mu_j) / \sigma_j \tag{3.10}$$

where μ_j and σ_j are the mean and standard deviation for values of the j-th attribute A_j^n respectively. As it will be shown, this scaling provides equal contributions of variables to the Euclidean similarity measure.

It was also suggested often to truncate the out-of-range components assuming that it is just eliminating the outliers (Aksoy and Haralick, 2001). However, truncating the out-of-range components could lead to loss of information from the dataset.

It was noted that providing all attributes are normally distributed, the probability of the attribute value normalised by (3.10) is in the [−1,1] range equals to 68%. If one applies an additional shift and rescaling as

$$x_{ij}^{*} = 0.5[(x_{ij} - \mu_j)/(3\sigma_j) + 1] \tag{3.11}$$

then this guarantees 99% of the values to be in the [0,1] range (Aksoy and Haralick, 2001). However, any shifting of the whole attribute column does not affect the distance metric (3.1). Hence, such an additional shifting has no practical applications to clustering of data sets.

3.3.2 Normalisation of categorical data sets

Normalisation of categorical and mixed datasets was practically not discussed in the literature. For example, the k-prototypes algorithm was applied to a non-normalised metric by Huang (1998). Larose (2005) suggested to apply either the min–max normalisation or Z-score standardisation techniques to numerical attributes and the matching dissimilarity measure without normalisation when mixed categorical and continuous variables are studied. He noted that perhaps, *the min–max normalisation may be preferred* in this case.

As it has been mentioned above, normalised metrics are introduced in this thesis such that regardless of the type of attributes their average contributions to the measures are equal to each other from statistical point of view. Although this idea has been recently discussed in the literature (Hastie et al., 2001), nothing was said about statistical consistency of the proposed estimators. In addition, they used biased estimators.

3.4 Statistical approach to normalisation of feature vectors

With geometric similarity measures, usually no assumption is made about the probability distribution of the attributes and similarity (dissimilarity) is based on the distances between feature vectors in the feature space (Aksoy and Haralick, 2001). Each record (row) of a dataset may be regarded as a random sample of a population under consideration, i.e. one has a dataset of N observations (samples) and each sample (record) is a realisation of possible values of the feature vector $\mathbf{A}$.

3.4.1 Estimators

For statistical treatment of feature vectors, one needs to know the probability distributions of their attributes. For a numerical attribute A_j^n, the probability

distribution identifies the probability of the attribute value falling within a particular interval within the range of possible values. For a categorical attribute A_j^c, the probability distribution identifies the probability of certain states occurring.

Suppose that $(X_1, X_2, \ldots, X_N)$ is a random sample of size N from a distribution of a real-valued random variable X with mean μ and standard deviation σ. As it has been mentioned in Chapter 2, an estimator is a function of the observable sample data (statistic) that is used to estimate an unknown population parameter (which is called the estimand). It is known (Spigel 1975, Giudici 2003) that a sample of N observations of a random variable X is a sequence of random variables $(X_1, X_2, \ldots, X_N)$ that are distributed identically as X. One can assume that the sample is a simple random sample when the random variables $(X_1, X_2, \ldots, X_N)$ are independent and therefore they constitute a sequence of independent and identically distributed random variables. Then $\mathbf{X}$ denotes the random vector formed by a sequence of random variables $\mathbf{X} = (X_1, X_2, \ldots, X_N)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ indicates the actually observed sample value.

Practically in all books on statistics one can find that the sample mean for the j-th feature

$$\bar{X}_j = \frac{1}{N}\sum_{i=1}^{N} X_{ij} \tag{3.12}$$

is an unbiased estimator of the unknown population mean μ, while the sample variance

$$S^2 = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2 \tag{3.13}$$

is a biased estimator of the population variance. The unbiased estimator of the population variance σ_j^2 for the j-th feature is

$$S_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(X_{ij} - \bar{X}_j\right)^2 \tag{3.14}$$

It is known that the above estimators (3.12) and (3.14) of the sample mean and the sample variance are consistent (Giudici 2003). One can use the above basic definition to estimate the average and variance for the j-th attribute of the data set. Sometimes it has been suggested to use the biased estimator (3.13) instead of the unbiased estimator (3.14) for the variance for the j-th attribute (Jain and Dubes, 1988).

It is assumed usually in the literature that each numerical feature has a normal (Gaussian) distribution with mean μ_j and standard deviation σ_j. However, in the general case, distribution functions are not known in advance and another function may be a better model for the attributes than the Gaussian distribution.

3.4.2 Earlier attempts of normalisation

The normalisation procedure can be implemented in different ways. For example, Aksoy and Haralick (2001) reviewed five normalisation methods for numerical data,

namely linear scaling to unit range, linear scaling to unit variance, transformation to a uniform [0,1] random variable, rank normalisation, and normalisation by fitting distributions. All these approaches intended to normalise each feature component to the [0,1] range. However, mainly these methods were equivalent to the above described Min–max normalisation and Z-score standardisation techniques. Note that in the textbooks by Jain and Dubes (1988) and by Larose (2005) only these two techniques were mentioned.

Hastie et al. (2001) described the following procedure for combining the p-individual attribute dissimilarities $d_j(x_{ij}, x_{i'j}), j = 1,2,\ldots,p$ into a single overall measure of dissimilarity $D(x_i, x_{i'})$ by means of a weighted average (convex combination)

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j}); \quad \sum_{j=1}^{p} w_j = 1 \tag{3.15}$$

where w_j is a weight assigned to the j-th attribute regulating the relative influence of the variable on the dissimilarity. The weight depends upon its relative contribution to the average object dissimilarity measure $\bar{D}$ over all pairs of records

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot \overline{d_j} \tag{3.16}$$

with the average dissimilarity of the j-th attribute

$$\overline{d_j} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j}) \tag{3.17}$$

Hence, the relative influence of the j-th attribute is $w_j \cdot \overline{d_j}$.

There are several questions and issues related to the above description of the normalisation procedure. Hastie et al. (2001) noted that setting $w_j \sim 1/\overline{d_j}$ for all attributes, irrespective of type, would give all attributes equal influence on characterising overall dissimilarity between pairs of objects. However, one has to realise that the above estimator is biased. Further, the question concerning the consistency of the proposed estimators was not discussed. They consider as example only the same case as in (Chen, 1973), namely the weighted Euclidean distance. On the other hand, there are metrics where the above approach is not valid. For example, it will be discussed in the next Chapter that if one considers the Tchebysheff metric for numerical attributes then (3.15)-(3.17) are not applicable. How can one normalise this metric?

Finally, if one studies a mixed metric that is a sum of two different metrics (for example, one metric is used for numerical data, while another metric is used for categorical data) then the above approach, i.e. formulae (3.15)-(3.17), is not applicable. Definitely, there is a need to discuss the application of the above idea in detail.

3.4.3 A new statistical approach to normalisation of attributes

To obtain a new normalised Euclidean metric, one should calculate the mean contribution of each j-th attribute to the metric $E|X_{1j} - X_{2j}|^2$ (here E means the

expectation of a variable) and to divide the attribute in all records by this mean (if the mean is equal to zero then this attribute should be removed from the feature vector). Hence, the normalised Euclidean metric can be introduced in the following way

$$\rho_E^*(\mathbf{x}_1,\mathbf{x}_2)=\left(\sum_{j=1}^{p}\alpha_j\,|\,x_{1j}-x_{2j}\,|^2\right)^{1/2}, \tag{3.18}$$

where $\alpha_j = 1/E\,|\,X_{1j}-X_{2j}\,|^2, X_{1j}$ and X_{2j} are independent random variables whose values are distributed in accordance with the distribution of the j-th attribute.

Since X_{1j} and X_{2j} are independent random variables having the same distribution, we obtain for the Euclidean metric,

$$E\,|\,X_{1j}-X_{2j}\,|^2 = EX_{1j}^2 - 2EX_{1j}EX_{2j} + EX_{2j}^2 = 2(EX_{1j}^2-(EX_{1j})^2) = 2\sigma_j^2,$$

where σ_j is the standard deviation of the j-th attribute. Thus, the normalised Euclidean metric has the following form

$$\rho_E^*(\mathbf{x}_1,\mathbf{x}_2)=\left(\sum_{j=1}^{p}\frac{(x_{1j}-x_{2j})^2}{2\sigma_j^2}\right)^{1/2}, \tag{3.19}$$

According to (3.14), it is possible to use the following unbiased estimator of the sample variance to estimate σ_j^2 in (3.19),

$$s_j^2 = \frac{1}{N-1}\sum_{r=1}^{N}(x_{rj} - \bar{x}_j)^2,$$

where $\bar{x}_{ij} = \frac{1}{N}\sum_{r=1}^{N} x_{rj}$ is the sample mean for the j-th attribute (see (3.12)).

From (3.19) we obtain the known form of normalisation of features:

$$x_{i1}^* = \frac{x_{i1} - \mu_1}{\sigma_1}, \ldots, x_{ip}^* = \frac{x_{ip} - \mu_p}{\sigma_p},$$

where μ_j is the mean of the j-th attribute.

3.4.4 Data sets with mixed attributes

For data sets with categorical attributes, it is possible to introduce different metrics (see, e.g. Gibert and Cortes, 1997; Huang, 1998; Ralambondrainy, 1995). One of the most cited variants of metrics (see, e.g. Huang, 1998) is studied here, namely the distance between two categorical feature vectors $\mathbf{y}_1 = (y_{11}, \ldots, y_{1l})$ and $\mathbf{y}_2 = (y_{21}, \ldots, y_{2l})$ is defined as

$$\rho_{cat}(\mathbf{y}_1, \mathbf{y}_2) = \omega(y_{11}, y_{21}) + \ldots + \omega(y_{1l}, y_{2l}) \qquad (3.20)$$

were

$$\omega(y_{1j}, y_{2j}) = \begin{cases} 0 & \text{for} \quad y_{1j} = y_{2j} \\ 1 & \text{for} \quad y_{1j} \neq y_{2j} \end{cases}$$

Evidently, the square of the metric (3.20) is

$$\rho_{cat}^2(\mathbf{y}_1, \mathbf{y}_2) = \omega^2(y_{11}, y_{21}) + \ldots + \omega^2(y_{1l}, y_{2l}) \tag{3.21}$$

Combining ρ_E and ρ_{cat} for mixed data, one obtains that the square distance between two mixed feature vectors $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ is

$$\rho^2((\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)) = \rho_E^2(\mathbf{x}_1, \mathbf{x}_2) + \rho_{cat}^2(\mathbf{y}_1, \mathbf{y}_2) \tag{3.22}$$

where $\rho_E^2(\mathbf{x}_1, \mathbf{x}_2)$ is defined by (3.1) and $\rho_{cat}^2(\mathbf{y}_1, \mathbf{y}_2)$ is defined by (3.21).

The same idea as it has been applied to numerical features, will be applied here to categorical ones, namely we will divide the contribution of each attribute to the distance measure by the contribution mean. Hence, the normalised mixed metric is defined similarly to (3.22)

$$\rho^*((\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)) = \left(\sum_{j=1}^{p} \alpha_j (x_{1j} - x_{2j})^2 + \sum_{j=1}^{l} \beta_j \omega^2(y_{1j}, y_{2j}) \right)^{1/2} \tag{3.23}$$

where $\alpha_j = 1/E(X_{1j} - X_{2j})^2$, $\beta_j = 1/E\omega^2(Y_{1j}, Y_{2j})$ and Y_{1j} , Y_{2j} are independent random variables whose values are distributed in accordance with the distribution of the A_j^c -th attribute. If the attribute A_j^c can take q_j values $\{y_{j1}, y_{j2}, \ldots, y_{jq_j}\}$ and the probabilities $\{p_{j1}, p_{j2}, \ldots, p_{jq_j}\}$ of these values are known then

$$E\omega^2(Y_{1j}, Y_{2j}) = E\omega(Y_{1j}, Y_{2j}) =$$
$$\sum_{\substack{r,s=1 \\ r \neq s}}^{q_j} 1 \cdot p_{jr} p_{js} = \sum_{r,s=1}^{q_j} p_{jr} p_{js} - (p_{j1}^2 + \ldots + p_{jq_j}^2)$$

or

$$E\omega^2(Y_{1j}, Y_{2j}) = (p_{j1} + \ldots + p_{jq_j})^2 - (p_{j1}^2 + \ldots + p_{jq_j}^2) = 1 - (p_{j1}^2 + \ldots + p_{jq_j}^2).$$

Thus, it follows from (3.19) that $\alpha_j = 1/2\sigma_j^2$ and from the above equality that

$$\beta_j = 1/(1 - (p_{j1}^2 + \ldots + p_{jq_j}^2)). \tag{3.24}$$

If the distribution of the attributes is unknown then to calculate α_j one can use the estimation (3.19), and to estimate $E\omega(Y_{1j}, Y_{2j})$ one can use the sampling mean

$$\hat{E}\omega^2(Y_{1j}, Y_{2j}) = \frac{1}{N^2} \sum_{r,s=1}^{N} \omega(y_{rj}, y_{sj}) \tag{3.25}$$

The estimation (3.25) is a biased estimator of $E\omega^2(Y_{1j}, Y_{2j})$, hence for small data sets it is better to use the following estimation

$$\hat{E}\omega^2(Y_{1j}, Y_{2j}) = \frac{2}{N(N-1)} \sum_{1 \le r < s \le N} \omega(y_{rj}, y_{sj}), \tag{3.26}$$

that is an unbiased estimator.

3.5 Comparing the accuracy of the clustering algorithms

Comparison of accuracy of clustering algorithms is not an easy task. In the case of datasets having labels (class labels), there are two methods commonly used for comparison: (i) calculating of accuracy, and (ii) calculation of Rand index (Rand, 1971) or its modifications (Hubert and Arabie, 1985).

3.5.1 Accuracy of clustering and Rand index

Using the former approach, Ng and Wong (2002) measured the results of application of their clustering algorithm by the clustering accuracy defined as

$$Acc_{NW} = \frac{\sum_{m=1}^{k} r_m}{N} \tag{3.27}$$

where r_m is the number of objects partitioned into the correct cluster m and N is the total number of records in the data set. The formula (3.27) was also used by Chan et al. (2004) to calculate the accuracy of their attributes-weighting algorithm that was tested by clustering an artificial data set. We need to note that to use the accuracy Acc_{NW}, one has to explain in the algorithm what 'correct' cluster is. Indeed, even if partitioning of the data set was absolutely correct Acc_{NW} can be very low or even be equal to zero just because two labels are replaced one by another. To avoid this problem, we have introduced the ideas of the assignment problem (see paragraph 3.5.2 below).

The Rand index or Rand measure is a measure of the similarity between two data clusterings. The classical definition is the following (Rand, 1971):

Let us consider a set S of N elements, and two partitions $\mathbf{C} = \{C_1, \ldots, C_k\}$ and $\mathbf{D} = \{D_1, \ldots, D_k\}$ of the data set. To calculate the Rand index, one needs first to calculate the following numbers: a is the number of pairs of elements in S that are in the same set in $\mathbf{C}$ and in the same set in $\mathbf{D}$; b is the number of pairs of elements in S that are in different sets in $\mathbf{C}$ and in different sets in $\mathbf{D}$; c is the number of pairs of elements in S that are in the same set in $\mathbf{C}$ and in different sets in $\mathbf{D}$; and d is the number of pairs of elements in S that are in different sets in $\mathbf{C}$ and in the same set in $\mathbf{D}$. Then the Rand index, (R), is calculated as

$$R = \frac{a+b}{a+b+c+d} \qquad (3.28)$$

Intuitively, one can think of $a+b$ as the number of agreements between **C** and **D** and $c+d$ as the number of disagreements between **C** and **D**.

The formula (3.28) was used by many researchers. In particular, it was used by Huang et al. (2005) to evaluate the performance of their attributes-weighting clustering algorithm in application to an artificial data set. It was possible to use (3.28) because the cluster labels of the data points in the synthetic data set were known. The Rand index has a value between 0 and 1 and the larger the Rand index, the higher the accuracy of the clustering.

3.5.2 Assignment problem and calculating the accuracy of clustering

Our calculation of the accuracy function has involved the ideas of a particular case of the assignment problem. In the classic formulation of the problem, there are a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task in such a way that the total cost of the assignment is minimised.

This problem is one of the fundamental combinatorial optimization problems. The latter is a branch of optimisation whose domain is optimisation problems where the set of candidate solutions is discrete or can be reduced to a discrete one, and the goal is to find the best possible solution. The space of all candidate solutions is called the search space.

Let us give a formal description of calculation of the accuracy of a clustering algorithm. For this purpose, consider data sets whose inherent structures are known in advance. Let us consider a data set having a categorical attribute **A** that may have k different states $\{a_1, a_2, \ldots, a_k\}$ that may be associated with labels of the clusters, i.e. the inherent structure (labels) of the data set is associated with the states of this attribute.

Our clustering algorithm will map the records to a discrete set of labels (classes). It is proposed to perform the normalisation procedure of the data set as it is described above and then to apply the clustering algorithm. After clustering, each record will belong to a cluster with a corresponding number m. For each m, let us assign a state $a_{\varphi(m)}$ of the attribute $\mathbf{A} = \{a_1, a_2, \ldots, a_k\}$ to the m-th cluster. Evidently, different clusters should have different states of the attribute **A**. Let us denote by $n_{m,j}$ the number of records with the attribute $A = a_j$ that belong to the m-th cluster.

For a given assignment φ, one can estimate the accuracy $Acc(\varphi)$ of the clustering as

$$Acc(\varphi) = \frac{\sum_{m=1}^{k} n_{m,\varphi(m)}}{N} \tag{3.29}$$

where $n_{m,\varphi(m)}$ is the number of records of the m-th cluster whose state of the attribute A is the same as the assigned $a_{\varphi(m)}$. The clustering accuracy is defined as maximum of $Acc(\varphi)$ for all possible assignments φ

$$Acc = \max_{\varphi} Acc(\varphi). \tag{3.30}$$

Evidently, the closer is Acc to 1 the less is the difference between the partitioning of the data after clustering and the partitioning of the data associated with the attribute A. If $Acc = 1$ then both partitioning into classes are the same.

Thus, one needs to solve the assignment problem with an efficiency matrix $n_{m,j}$, $(m, j = 1,\ldots,k)$ in order to find the clustering accuracy Acc.

We can rewrite the formula (3.29) for calculating the accuracy of clustering results measured by the clustering accuracy Acc, as

$$Acc = \frac{\sum_{m=1}^{k} n_{m,j}}{N} \tag{3.31}$$

where $n_{m,j}$ is the number of records within the cluster m having the same label as the generated cluster label a_j, and N is the total number of records in the data set.

To explain the way we calculated the accuracy of clustering, let us consider an example of $k = 3$ clusters having 10, 9, and 8 records respectively (Figure 3.1). We can have in total $3! = 6$ different assignments.

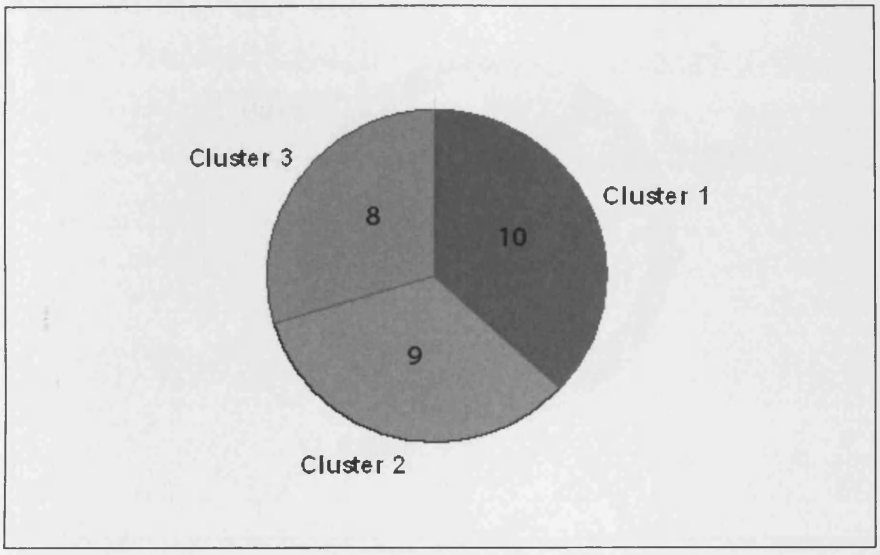


Figure 3.1: An example of a data set with 3 clusters having 10, 9, and 8 records respectively.

Assume there is an attribute that may take 3 different values (class labels): blue, azure and yellow. Let us consider further the following distribution of total $N = 27$ records by the labels. The cluster 1 has 10 records having the following labels: 3 blue, 3 azure and 4 yellow; the cluster 2 has 9 records having the following labels: 5 blue, 2 azure and 2 yellow; and the cluster 3 has 8 records having the following labels: 3 blue, 3 azure and 2 yellow (Figure 3.2).

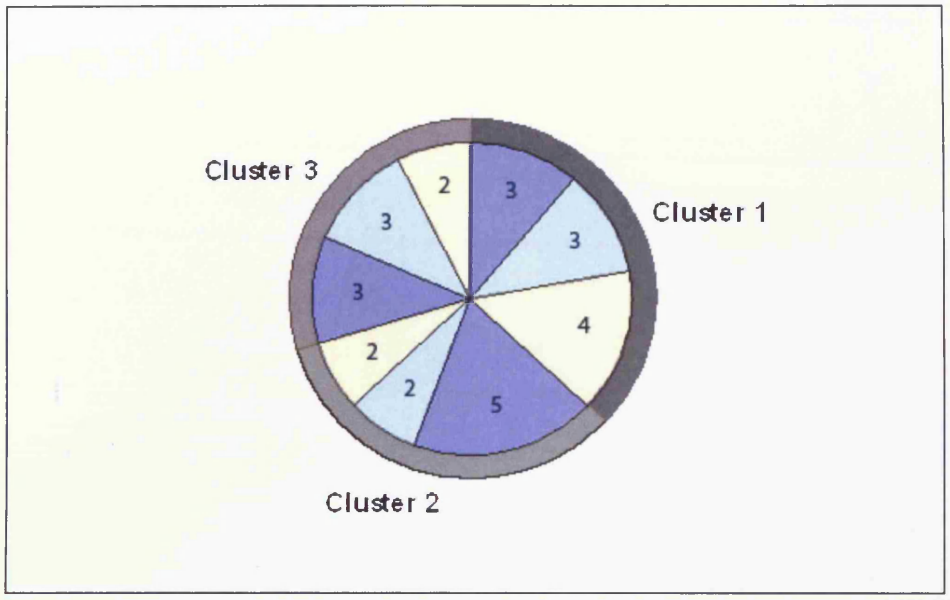


Figure 3.2: An example of a data set having $N = 27$ records, 3 clusters, and the class labels of the records: blue, azure and yellow. The cluster labels are not yet assigned.

Then each cluster may be labelled (assigned) by one of these colours, i.e. there is an attribute **A** whose states are blue (a_1), azure (a_2), and yellow (a_3). The goal is to find an optimal assignment of labels to clusters such that there is a maximum total matching between the cluster labels and the labels of records belonging to each cluster. The assignments corresponding to the example under consideration are presented in Figures 3.3-3.8.

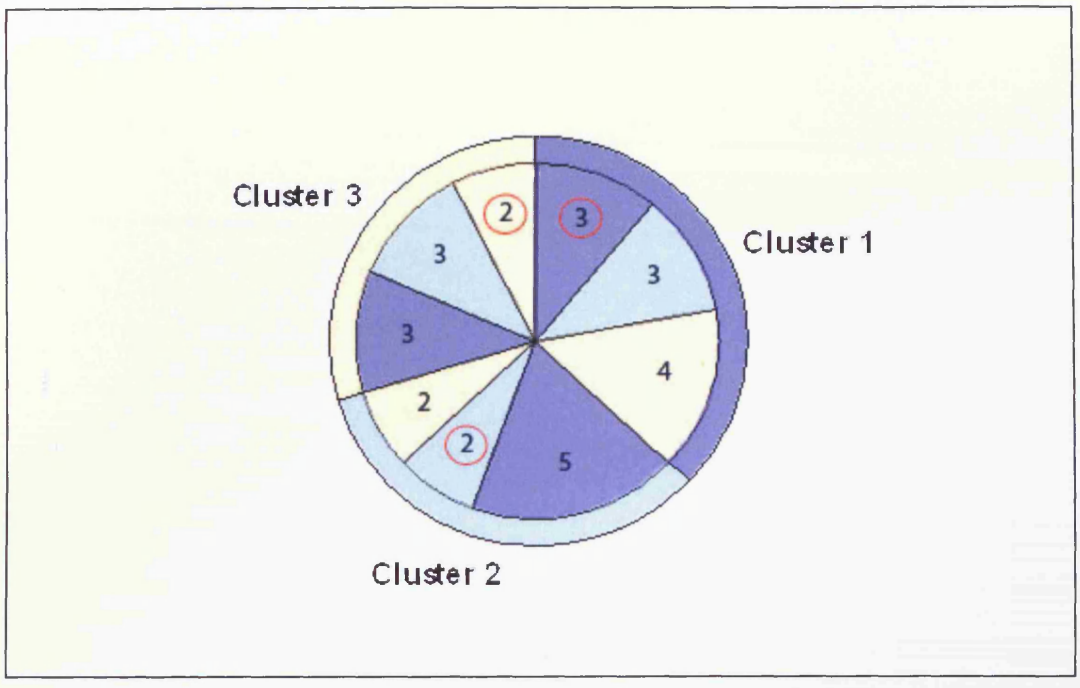


Figure 3.3: The first case of possible assignments of labels to clusters: the label of cluster 1 is blue, the label of cluster 2 is azure, and the label of cluster 3 is yellow. For each cluster, the number of matching labels is in a red circle: $n_{1,blue} = n_{1,1} = 3$, $n_{2,azure} = n_{2,2} = 2$, and $n_{3,yellow} = n_{3,3} = 2$.

If one calculates Acc in accordance with (3.31) in the first case of assignments then the result is

$$Acc_1 = \frac{3+2+2}{27} = \frac{7}{27}.$$

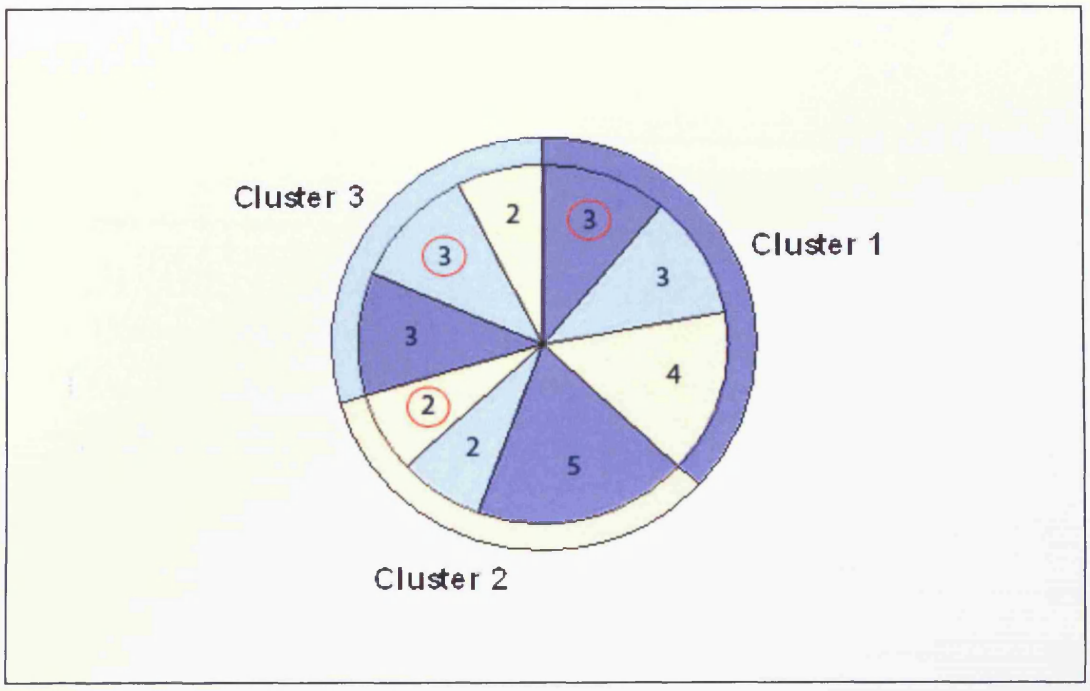


Figure 3.4: The second case of possible assignments of labels to clusters: the label of cluster 1 is blue, the label of cluster 2 is yellow, and the label of cluster 3 is azure. For each cluster, the number of matching labels is in a red circle: $n_{1,blue} = n_{1,1} = 3$, $n_{2,yellow} = n_{2,3} = 2$, and $n_{3,azure} = n_{3,2} = 3$.

If one calculates Acc in accordance with (3.31) in the second case of assignments then the result is

$$Acc_2 = \frac{3+3+2}{27} = \frac{8}{27}.$$

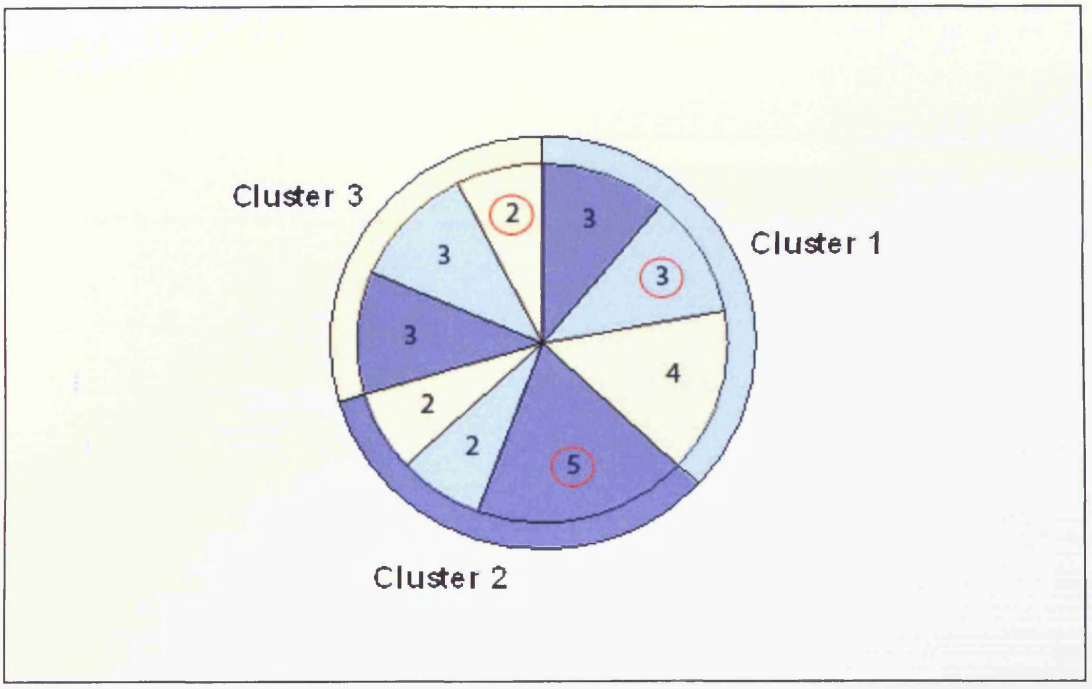


Figure 3.5: The third case of possible assignments of labels to clusters: the label of cluster 1 is azure, the label of cluster 2 is blue, and the label of cluster 3 is yellow. For each cluster, the number of matching labels is in a red circle: $n_{1,azure} = n_{1,2} = 3$, $n_{2,blue} = n_{2,1} = 5$, and $n_{3,yellow} = n_{3,3} = 2$.

If one calculates Acc in accordance with (3.31) in the third case of assignments then the result is

$$Acc_3 = \frac{3+5+2}{27} = \frac{10}{27}$$

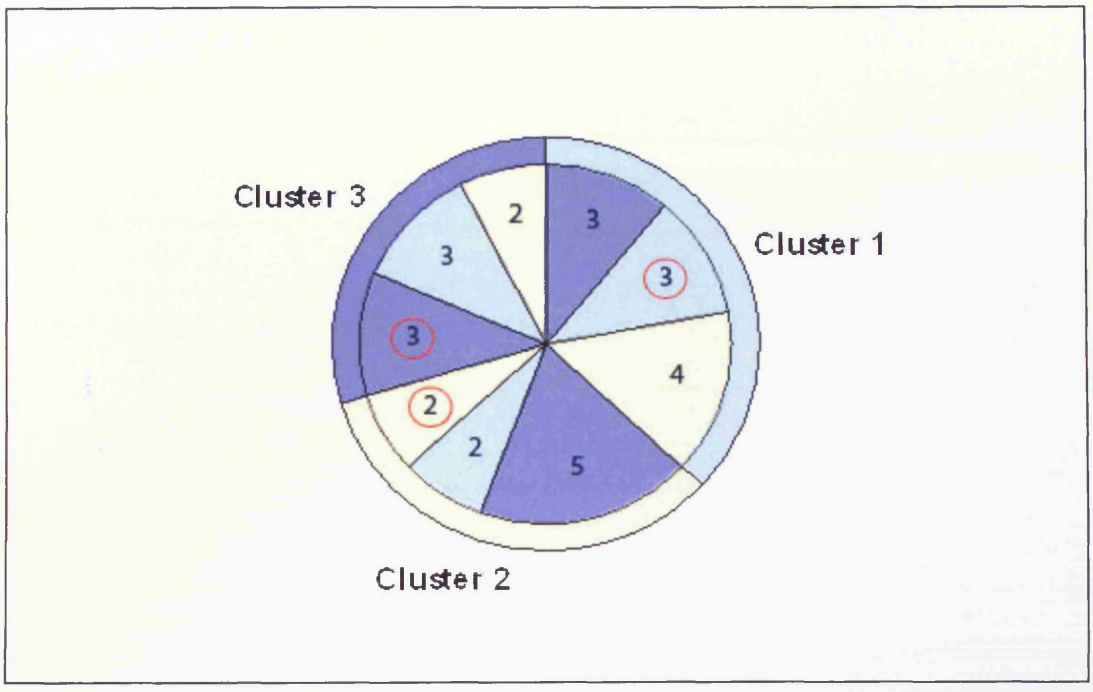


Figure 3.6: The fourth case of possible assignments of labels to clusters: the label of cluster 1 is azure, the label of cluster 2 is yellow, and the label of cluster 3 is blue. For each cluster, the number of matching labels is in a red circle: $n_{1,azure} = n_{1,2} = 3$, $n_{2,yellow} = n_{2,3} = 2$, and $n_{3,blue} = n_{3,1} = 3$.

If one calculates Acc in accordance with (3.31) in the fourth case of assignments then the result is

$$Acc_4 = \frac{3+2+3}{27} = \frac{8}{27}.$$

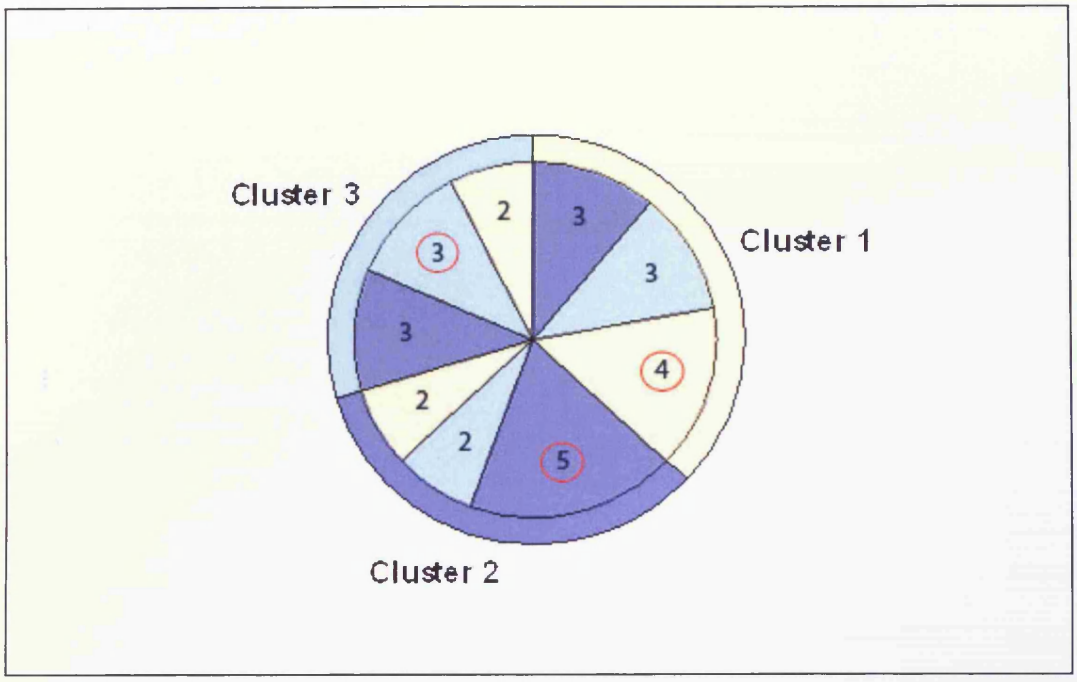


Figure 3.7: The fifth case of possible assignments of labels to clusters: the label of cluster 1 is yellow, the label of cluster 2 is blue, and the label of cluster 3 is azure. For each cluster, the number of matching labels is in a red circle: $n_{1,yellow} = n_{1,3} = 4$, $n_{2,blue} = n_{2,1} = 5$, and $n_{3,azure} = n_{3,2} = 3$.

If one calculates Acc in accordance with (3.31) in the fifth case of assignments then the result is

$$Acc_5 = \frac{4+5+3}{27} = \frac{12}{27}.$$

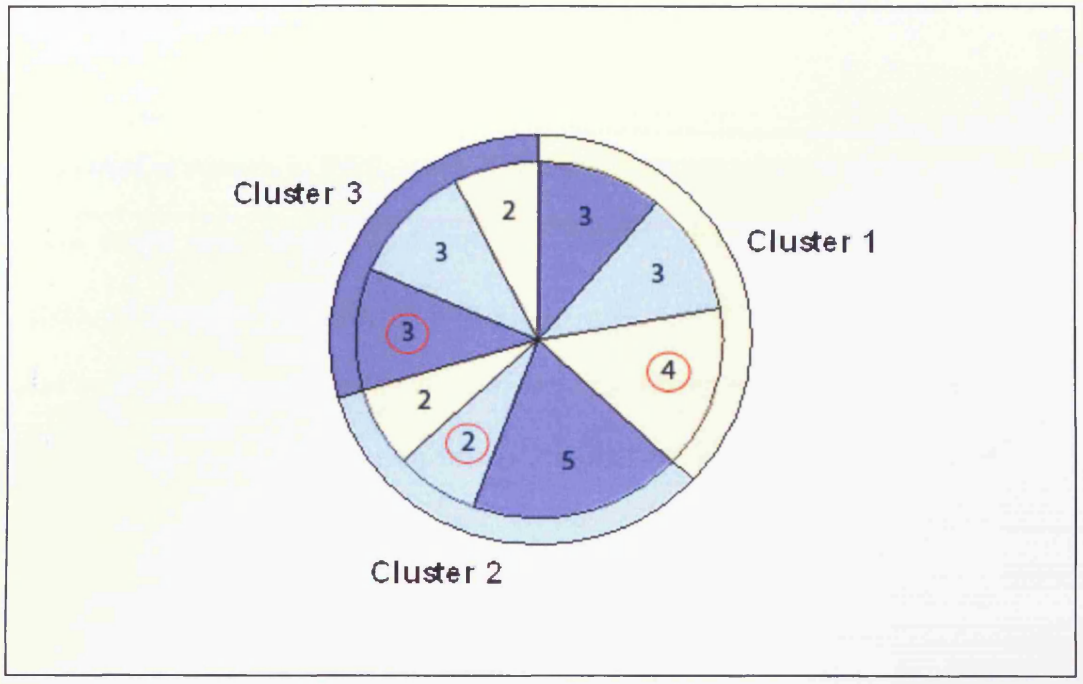


Figure 3.8: The sixth case of possible assignments of labels to clusters: the label of cluster 1 is yellow, the label of cluster 2 is azure, and the label of cluster 3 is blue. For each cluster, the number of matching labels is in a red circle: $n_{1,yellow} = n_{1,3} = 4$, $n_{2,azure} = n_{2,2} = 2$, and $n_{3,blue} = n_{3,1} = 3$.

If one calculates Acc in accordance with (3.31) in the sixth case of assignments then the result is

$$Acc_6 = \frac{4+2+3}{27} = \frac{9}{27}.$$

One can see that the maximum of Acc_j for $j = 1,\ldots,6$ is $Acc_5 = \frac{12}{27}$. This value is taken as the accuracy of the above clustering.

If the total number of records in the data set N is large then calculations of the Rand index R is a rather time consuming procedure. Indeed, one has to consider all possible pairs of records (exhaustive search), i.e. the number of operations is proportional to N^2. In this case, calculation of the assignment based accuracy Acc of clustering is simpler. However, even this procedure is fast only when the total number of records in the data set N is small because in this case one can consider all possible cases as we have considered above. If k is large then to calculate the assignment based accuracy Acc one has to use one of existing algorithms to solve the corresponding assignment problem. In all data sets considered in this thesis we have $k \leq 8$ and therefore we have not employed any of the special algorithms for solving the assignment problem.

3.6 Applications to data sets

The above methods will be applied to several data sets from the UC Irvine repository. All records in those data sets have the class labels and, hence, "true clustering" can be checked.

3.6.1 Soybean Disease Data Set

The soybean data set has 47 records ($N = 47$) with 35 attributes. Each record is attributed to one of the 4 following diseases: Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. The Phytophthora Rot has 17 observations, while other diseases were observed 10 times each. This is a standard categorical data set that was studied a number of times to test clustering algorithms (see, e.g. Huang, 1998; Michalski and Stepp, 1983; Huang, 1997). First the clustering procedure has been applied to the data set without normalisation of the data. Then the clustering procedure with normalisation of all attributes has been applied to the data set. Both procedures with and without normalisation have been applied 100 times to the data set. Table 3.1 presents the results of application of the k -prototypes algorithm without normalisation of the attributes to the soybean data set: the values of the clustering accuracy (Acc), the objective function (J), the Rand index (R), and the number of iterations the algorithm needed to converge, and the attempts showing the best value of the objective function (BF).

Here and henceforth the plus sign in the Tables means that the best value of the objective function has been obtained in the simulations.

Table 3.1: Clustering of the Soybean data set without normalisation of the attributes.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	1	1	199	3	+
2	1	1	199	4	+
3	1	1	199	2	+
4	1	1	199	4	+
5	1	1	199	6	+
6	1	1	199	3	+
7	1	1	199	2	+
8	1	1	199	5	+
9	1	1	199	3	+
10	1	1	199	4	+
11	1	1	199	3	+
12	1	1	199	3	+
13	1	1	199	2	+
14	1	1	199	1	+
15	0.9787	0.9759	199	4	+
16	0.9787	0.9759	201	5	
17	0.9787	0.9759	199	4	+
18	0.9787	0.9759	199	4	+
19	0.9787	0.9759	199	2	+
20	0.9787	0.9759	199	2	+
21	0.9787	0.9759	199	2	+
22	0.9787	0.9759	199	3	+
23	0.9787	0.9759	199	3	+
24	0.9787	0.9759	199	4	+
25	0.9787	0.9759	199	2	+
26	0.9787	0.9759	199	3	+
27	0.9787	0.9759	199	6	+
28	0.9787	0.9759	199	1	+
29	0.9787	0.9759	199	1	+
30	0.9787	0.9759	202	3	
31	0.9787	0.9759	199	3	+
32	0.9574	0.9537	199	1	+
33	0.9574	0.9537	199	4	+
34	0.9574	0.9537	199	5	+
35	0.9574	0.9537	199	2	+
36	0.9574	0.9537	199	6	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
37	0.9574	0.9537	199	2	+
38	0.9574	0.9537	199	2	+
39	0.8936	0.8982	202	8	
40	0.8298	0.8594	211	3	
41	0.766	0.8372	213	3	
42	0.7447	0.8335	215	4	
43	0.7447	0.8335	216	7	
44	0.7234	0.8233	228	1	
45	0.7234	0.8316	216	2	
46	0.7234	0.8316	218	1	
47	0.7234	0.8233	228	3	
48	0.7234	0.8659	246	3	
49	0.7234	0.8316	217	2	
50	0.7234	0.8233	239	2	
51	0.7234	0.8316	217	2	
52	0.7021	0.8261	261	4	
53	0.7021	0.8205	227	2	
54	0.7021	0.8205	227	3	
55	0.7021	0.8205	224	3	
56	0.7021	0.8205	224	3	
57	0.7021	0.8205	238	2	
58	0.6809	0.8196	220	2	
59	0.6809	0.8196	220	5	
60	0.6809	0.8196	220	2	
61	0.6809	0.8196	220	2	
62	0.6809	0.8094	225	2	
63	0.6809	0.8094	225	4	
64	0.6809	0.8196	220	4	
65	0.6809	0.8196	220	3	
66	0.6809	0.8196	220	3	
67	0.6809	0.8094	225	4	
68	0.6809	0.8094	225	4	
69	0.6596	0.8649	260	5	
70	0.6596	0.8187	237	3	
71	0.6596	0.8298	239	4	
72	0.6596	0.8649	260	2	
73	0.6383	0.8464	260	5	
74	0.6383	0.8464	260	3	
75	0.6383	0.8427	252	3	
76	0.6383	0.8187	239	4	
77	0.6383	0.8279	238	5	
78	0.6383	0.8344	238	2	
79	0.6383	0.7993	245	2	
80	0.617	0.8409	252	5	
81	0.617	0.8409	252	3	
82	0.617	0.8335	253	5	
83	0.617	0.8409	253	4	
84	0.617	0.79	244	4	
85	0.617	0.8409	253	3	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
86	0.617	0.8409	252	4	
87	0.617	0.8409	253	4	
88	0.617	0.8335	238	3	
89	0.617	0.8252	258	6	
90	0.617	0.8409	252	3	
91	0.617	0.7484	247	2	
92	0.617	0.8409	254	3	
93	0.5957	0.7493	244	2	
94	0.5957	0.7299	246	2	
95	0.5745	0.7364	241	7	
96	0.5532	0.7086	241	4	
97	0.5532	0.7475	277	2	
98	0.5532	0.7068	245	2	
99	0.4894	0.7484	277	3	
100	0.4468	0.5624	290	2	

Table 3.2 presents the results of application of the k -prototypes algorithm with normalisation of the attributes to the soybean data set. The meanings of the columns presenting the results in Tables 3.2 -3.8 are the same as the meanings described for the Table 3.1.

Table 3.2: Clustering of the Soybean data set with normalisation of the attributes.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	1	1	359.8666	5	+
2	1	1	359.8666	1	+
3	1	1	359.8666	1	+
4	1	1	359.8666	4	+
5	1	1	359.8666	5	+
6	1	1	359.8666	3	+
7	1	1	359.8666	3	+
8	1	1	359.8666	2	+
9	1	1	359.8666	4	+
10	1	1	359.8666	3	+
11	1	1	359.8666	3	+
12	1	1	359.8666	6	+
13	1	1	359.8666	2	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
14	1	1	359.8666	3	+
15	1	1	359.8666	1	+
16	1	1	359.8666	2	+
17	1	1	359.8666	5	+
18	1	1	359.8666	4	+
19	1	1	359.8666	4	+
20	1	1	359.8666	3	+
21	1	1	359.8666	6	+
22	1	1	359.8666	5	+
23	1	1	359.8666	4	+
24	1	1	359.8666	5	+
25	1	1	359.8666	3	+
26	1	1	359.8666	1	+
27	1	1	359.8666	4	+
28	1	1	359.8666	3	+
29	1	1	359.8666	3	+
30	0.9787	0.9759	361.1875	3	
31	0.9787	0.9759	362.3914	3	
32	0.9787	0.9759	361.1875	3	
33	0.9787	0.9759	361.1875	1	
34	0.9787	0.9759	361.1875	3	
35	0.9787	0.9759	361.1875	3	
36	0.9787	0.9759	361.1875	7	
37	0.9787	0.9759	361.1875	3	
38	0.9787	0.9759	362.3914	3	
39	0.9787	0.9759	361.1875	3	
40	0.9787	0.9759	361.1875	5	
41	0.9787	0.9759	361.1875	4	
42	0.9787	0.9759	361.1875	3	
43	0.9787	0.9759	361.1875	3	
44	0.9787	0.9759	362.3914	3	
45	0.9787	0.9759	362.3914	2	
46	0.9787	0.9759	361.1875	3	
47	0.9787	0.9759	361.1875	3	
48	0.9787	0.9759	361.1875	2	
49	0.9787	0.9759	361.1875	5	
50	0.7872	0.8427	399.8791	4	
51	0.7872	0.8427	399.8791	2	
52	0.766	0.8511	462.1653	4	
53	0.766	0.8344	445.4421	3	
54	0.7447	0.8298	422.7914	1	
55	0.7447	0.8298	422.7914	1	
56	0.7447	0.8335	404.7635	1	
57	0.7234	0.8316	402.3598	3	
58	0.7234	0.8233	419.9152	3	
59	0.7234	0.8316	397.2588	2	
60	0.7234	0.8881	497.091	4	
61	0.7234	0.8705	502.1693	2	
62	0.7234	0.8881	497.091	4	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
63	0.7234	0.8316	397.2588	3	
64	0.7021	0.8205	413.3237	4	
65	0.7021	0.8205	418.2429	1	
66	0.7021	0.8205	419.5059	3	
67	0.7021	0.8205	413.3237	3	
68	0.7021	0.8205	414.5062	2	
69	0.7021	0.8853	496.3033	2	
70	0.7021	0.8261	463.5563	3	
71	0.7021	0.8205	422.24	3	
72	0.7021	0.8853	496.3033	3	
73	0.7021	0.8205	414.5062	3	
74	0.7021	0.8205	413.3237	3	
75	0.6809	0.8196	410.2062	2	
76	0.6809	0.8196	418.711	2	
77	0.6809	0.7743	447.8564	4	
78	0.6809	0.8196	410.2062	2	
79	0.6809	0.8196	422.9991	2	
80	0.6809	0.827	499.2963	5	
81	0.6809	0.7909	474.2939	2	
82	0.6809	0.8057	458.2709	2	
83	0.6596	0.802	450.3792	3	
84	0.6596	0.7919	472.0441	2	
85	0.6596	0.8002	510.3877	6	
86	0.6596	0.8501	513.679	1	
87	0.6383	0.7817	467.9814	4	
88	0.6383	0.7983	509.8043	2	
89	0.6383	0.79	455.1688	4	
90	0.6383	0.8427	501.1875	4	
91	0.6383	0.8427	495.3664	4	
92	0.617	0.7558	474.2798	3	
93	0.5957	0.7234	494.6445	2	
94	0.5957	0.6873	495.5843	4	
95	0.5957	0.79	504.0347	3	
96	0.5957	0.8242	493.1026	3	
97	0.5957	0.8252	493.0092	3	
98	0.4681	0.5643	547.0896	2	
99	0.4681	0.5643	548.6295	3	
100	0.4468	0.5624	548.6995	2	

Since the data set is quite small, the "true clustering" ($Acc = 1$) has been obtained quite often in both cases. $Acc = 1$ has been obtained in 14% after clustering without normalisation and in 29% after clustering with normalisation. The average accuracy

in both cases has been 0.782979 and 0.829574 respectively for the former and the latter cases.

3.6.2 Wine Data Set

The wine data set has 178 records ($N = 178$) with 14 attributes. The first attribute indicates the class (cultivar) and it takes three categorical values, while the rest of the attributes are numerical. Clustering has been performed using the first categorical attribute. First the clustering procedure has been applied to the data set without normalisation of the data (see Table 3.3).

Table 3.3: Clustering of the Wine data set without normalisation of the attributes.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.7022	0.7187	2370689.7	11	+
2	0.7022	0.7187	2370689.7	7	+
3	0.7022	0.7187	2370689.7	4	+
4	0.7022	0.7187	2370689.7	8	+
5	0.7022	0.7187	2370689.7	6	+
6	0.7022	0.7187	2370689.7	5	+
7	0.7022	0.7187	2370689.7	6	+
8	0.7022	0.7187	2370689.7	6	+
9	0.7022	0.7187	2370689.7	4	+
10	0.7022	0.7187	2370689.7	11	+
11	0.7022	0.7187	2370689.7	4	+
12	0.7022	0.7187	2370689.7	5	+
13	0.7022	0.7187	2370689.7	7	+
14	0.7022	0.7187	2370689.7	3	+
15	0.7022	0.7187	2370689.7	5	+
16	0.7022	0.7187	2370689.7	3	+
17	0.7022	0.7187	2370689.7	3	+
18	0.7022	0.7187	2370689.7	5	+
19	0.7022	0.7187	2370689.7	6	+
20	0.7022	0.7187	2370689.7	4	+
21	0.7022	0.7187	2370689.7	8	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
22	0.7022	0.7187	2370689.7	7	+
23	0.7022	0.7187	2370689.7	14	+
24	0.7022	0.7187	2370689.7	9	+
25	0.7022	0.7187	2370689.7	3	+
26	0.7022	0.7187	2370689.7	12	+
27	0.7022	0.7187	2370689.7	5	+
28	0.7022	0.7187	2370689.7	7	+
29	0.7022	0.7187	2370689.7	4	+
30	0.7022	0.7187	2370689.7	4	+
31	0.7022	0.7187	2370689.7	9	+
32	0.7022	0.7187	2370689.7	9	+
33	0.7022	0.7187	2370689.7	7	+
34	0.7022	0.7187	2370689.7	10	+
35	0.7022	0.7187	2370689.7	7	+
36	0.7022	0.7187	2370689.7	9	+
37	0.7022	0.7187	2370689.7	3	+
38	0.7022	0.7187	2370689.7	3	+
39	0.7022	0.7187	2370689.7	2	+
40	0.7022	0.7187	2370689.7	5	+
41	0.7022	0.7187	2370689.7	8	+
42	0.7022	0.7187	2370689.7	5	+
43	0.7022	0.7187	2370689.7	7	+
44	0.7022	0.7187	2370689.7	7	+
45	0.7022	0.7187	2370689.7	10	+
46	0.7022	0.7187	2370689.7	5	+
47	0.7022	0.7187	2370689.7	10	+
48	0.7022	0.7187	2370689.7	3	+
49	0.7022	0.7187	2370689.7	5	+
50	0.7022	0.7187	2370689.7	4	+
51	0.7022	0.7187	2370689.7	6	+
52	0.7022	0.7187	2370689.7	8	+
53	0.7022	0.7187	2370689.7	4	+
54	0.7022	0.7187	2370689.7	9	+
55	0.7022	0.7187	2370689.7	4	+
56	0.7022	0.7187	2370689.7	5	+
57	0.7022	0.7187	2370689.7	3	+
58	0.7022	0.7187	2370689.7	5	+
59	0.7022	0.7187	2370689.7	7	+
60	0.7022	0.7187	2370689.7	7	+
61	0.7022	0.7187	2370689.7	5	+
62	0.7022	0.7187	2370689.7	5	+
63	0.7022	0.7187	2370689.7	7	+
64	0.7022	0.7187	2370689.7	7	+
65	0.7022	0.7187	2370689.7	6	+
66	0.7022	0.7187	2370689.7	5	+
67	0.7022	0.7187	2370689.7	6	+
68	0.7022	0.7187	2370689.7	5	+
69	0.7022	0.7187	2370689.7	5	+
70	0.7022	0.7187	2370689.7	4	+

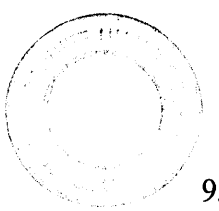
N	Accuracy	Rand Index	Objective function J	Iterations	BF
71	0.7022	0.7187	2370689.7	4	+
72	0.7022	0.7187	2370689.7	9	+
73	0.7022	0.7187	2370689.7	9	+
74	0.7022	0.7187	2370689.7	7	+
75	0.7022	0.7187	2370689.7	5	+
76	0.7022	0.7187	2370689.7	4	+
77	0.7022	0.7187	2370689.7	7	+
78	0.7022	0.7187	2370689.7	11	+
79	0.7022	0.7187	2370689.7	4	+
80	0.7022	0.7187	2370689.7	9	+
81	0.7022	0.7187	2370689.7	6	+
82	0.7022	0.7187	2370689.7	5	+
83	0.7022	0.7187	2370689.7	4	+
84	0.7022	0.7187	2370689.7	7	+
85	0.7022	0.7187	2370689.7	10	+
86	0.7022	0.7187	2370689.7	10	+
87	0.7022	0.7187	2370689.7	6	+
88	0.7022	0.7187	2370689.7	2	+
89	0.5955	0.6898	2631657.1	2	
90	0.5787	0.688	2625223.2	2	
91	0.573	0.6919	2633555.3	11	
92	0.573	0.6919	2633555.3	15	
93	0.573	0.6919	2633555.3	10	
94	0.573	0.6919	2633555.3	13	
95	0.573	0.6919	2633555.3	7	
96	0.573	0.6919	2633555.3	11	
97	0.573	0.6919	2633555.3	13	
98	0.573	0.6919	2633555.3	11	
99	0.573	0.6919	2633555.3	11	
100	0.573	0.6919	2633555.3	11	

Then the clustering procedure with normalisation of all attributes has been applied to the data set. Both procedures with and without normalisation have been applied 100 times to the data set (see Tables 3.3 and 3.4). Although the data set is quite small, the "true clustering" ($Acc = 1$) has not been obtained.

Table 3.4: Clustering of the Wine data set with normalisation of the attributes.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.9719	0.962	635.7884	7	
2	0.9719	0.962	635.7884	10	
3	0.9719	0.962	635.7884	9	
4	0.9719	0.962	635.7884	6	
5	0.9719	0.962	635.7884	5	
6	0.9719	0.962	635.7884	8	
7	0.9719	0.962	635.7884	5	
8	0.9719	0.962	635.7884	7	
9	0.9719	0.962	635.7884	5	
10	0.9719	0.962	635.7884	10	
11	0.9719	0.962	635.7884	4	
12	0.9719	0.962	635.7884	10	
13	0.9719	0.962	635.7884	4	
14	0.9719	0.962	635.7884	5	
15	0.9719	0.962	635.7884	6	
16	0.9719	0.962	635.7884	9	
17	0.9663	0.9543	635.3746	3	+
18	0.9663	0.9543	635.3746	4	+
19	0.9663	0.9543	635.3746	4	+
20	0.9663	0.9543	635.3746	4	+
21	0.9663	0.9543	635.3746	7	+
22	0.9663	0.9543	635.3746	9	+
23	0.9663	0.9543	635.3746	8	+
24	0.9663	0.9543	635.3746	6	+
25	0.9663	0.9543	635.3746	5	+
26	0.9663	0.9543	635.3746	8	+
27	0.9663	0.9543	635.3746	6	+
28	0.9663	0.9543	635.3746	6	+
29	0.9663	0.9543	635.3746	3	+
30	0.9663	0.9543	635.3746	4	+
31	0.9663	0.9543	635.3746	7	+
32	0.9663	0.9543	635.3746	13	+
33	0.9663	0.9543	635.3746	10	+
34	0.9663	0.9543	635.3746	4	+
35	0.9663	0.9543	635.3746	12	+
36	0.9663	0.9543	635.3746	5	+
37	0.9663	0.9543	635.3746	7	+
38	0.9663	0.9543	635.3746	9	+
39	0.9663	0.9543	635.3746	3	+
40	0.9663	0.9543	635.3746	7	+
41	0.9607	0.9467	636.3877	6	
42	0.9607	0.9467	636.3877	4	
43	0.9607	0.9467	636.3877	5	
44	0.9607	0.9467	636.3877	3	
45	0.9607	0.9467	636.3877	5	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
46	0.9607	0.9467	636.3877	6	
47	0.9607	0.9467	636.3877	4	
48	0.9607	0.9467	636.3877	5	
49	0.9607	0.9467	636.3877	5	
50	0.9607	0.9467	636.3877	6	
51	0.9607	0.9467	636.3877	8	
52	0.9607	0.9467	636.3877	5	
53	0.9607	0.9467	636.3877	8	
54	0.9607	0.9467	636.3877	5	
55	0.9607	0.9467	636.3877	8	
56	0.9607	0.9467	636.3877	5	
57	0.9607	0.9467	636.3877	4	
58	0.9607	0.9467	636.3877	6	
59	0.9607	0.9467	636.3877	9	
60	0.9607	0.9467	636.3877	5	
61	0.9551	0.9392	636.2708	6	
62	0.9551	0.9392	636.2708	3	
63	0.9551	0.9392	636.2708	5	
64	0.9551	0.9392	636.2708	5	
65	0.9551	0.9392	636.2708	5	
66	0.9551	0.9392	636.2708	4	
67	0.9551	0.9392	636.2708	4	
68	0.9551	0.9392	636.2708	4	
69	0.9551	0.9392	636.2708	3	
70	0.9551	0.9392	636.2708	5	
71	0.9551	0.9392	636.2708	5	
72	0.9551	0.9392	636.2708	4	
73	0.9551	0.9392	636.2708	5	
74	0.9551	0.9392	636.2708	4	
75	0.9494	0.9311	637.6293	5	
76	0.9494	0.9311	637.6293	8	
77	0.9494	0.9311	637.6293	6	
78	0.9494	0.9311	637.6293	7	
79	0.9494	0.9311	637.6293	7	
80	0.9494	0.9311	637.6293	8	
81	0.9494	0.9311	637.6293	6	
82	0.9494	0.9311	637.6293	10	
83	0.9494	0.9311	637.6293	6	
84	0.9494	0.9311	637.6293	12	
85	0.9494	0.9311	637.6293	5	
86	0.9494	0.9311	637.6293	6	
87	0.9494	0.9311	637.6293	7	
88	0.9494	0.9311	637.6293	5	
89	0.9494	0.9311	637.6293	15	
90	0.9494	0.9311	637.6293	7	
91	0.9494	0.9311	637.6293	9	
92	0.9494	0.9311	637.6293	5	
93	0.9494	0.9311	637.6293	5	
94	0.9494	0.9311	637.6293	7	



N	Accuracy	Rand Index	Objective function J	Iterations	BF
95	0.9494	0.9311	637.6293	3	
96	0.9494	0.9311	637.6293	7	
97	0.9494	0.9311	637.6293	6	
98	0.6236	0.7029	789.6402	8	
99	0.5899	0.684	804.1441	7	
100	0.5281	0.6709	793.4206	4	

The obtained average accuracy value 0.687022 in the case without normalisation of the data has been considerably smaller than the obtained average accuracy value 0.949045 in the case with normalisation of the data. After application of the normalisation procedure to the data set the Rand index has increased from 0.7187 to 0.9543.

3.6.3 Statlog (Heart Diseases) Data Set

The Heart Diseases data set has 270 records ($N = 270$) with 13 attributes (they have been extracted from a larger set of 75). There are no missing values. The last attribute indicates the class (absence (1) or presence (2) of heart disease) and it takes two categorical values. There are 7 categorical attributes and 6 numerical. Clustering has been performed using the last class attribute. First the clustering procedure has been applied to the data set without normalisation of the data (see Table 3.5).

Table 3.5: Clustering of the Heart Diseases data set without normalisation of the attributes

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.5926	0.5154	548278.3735	7	+
2	0.5926	0.5154	548278.3735	8	+
3	0.5926	0.5154	548278.3735	10	+
4	0.5926	0.5154	548278.3735	5	+
5	0.5926	0.5154	548278.3735	10	+
6	0.5926	0.5154	548280.3531	5	
7	0.5926	0.5154	548278.3735	12	+
8	0.5926	0.5154	548278.3735	6	+
9	0.5926	0.5154	548278.3735	5	+
10	0.5926	0.5154	548278.3735	12	+
11	0.5926	0.5154	548278.3735	9	+
12	0.5926	0.5154	548278.3735	8	+
13	0.5926	0.5154	548278.3735	13	+
14	0.5926	0.5154	548278.3735	3	+
15	0.5926	0.5154	548278.3735	14	+
16	0.5926	0.5154	548278.3735	10	+
17	0.5926	0.5154	548278.3735	9	+
18	0.5926	0.5154	548278.3735	7	+
19	0.5926	0.5154	548278.3735	11	+
20	0.5926	0.5154	548278.3735	10	+
21	0.5926	0.5154	548278.3735	12	+
22	0.5926	0.5154	548280.3531	6	
23	0.5926	0.5154	548278.3735	7	+
24	0.5926	0.5154	548278.3735	9	+
25	0.5926	0.5154	548278.3735	9	+
26	0.5926	0.5154	548278.3735	11	+
27	0.5926	0.5154	548278.3735	10	+
28	0.5926	0.5154	548278.3735	8	+
29	0.5926	0.5154	548278.3735	8	+
30	0.5926	0.5154	548278.3735	8	+
31	0.5926	0.5154	548278.3735	7	+
32	0.5926	0.5154	548278.3735	11	+
33	0.5889	0.514	548305.2053	8	
34	0.5889	0.514	548305.2053	10	
35	0.5889	0.514	548305.2053	12	
36	0.5889	0.514	548305.2053	8	
37	0.5889	0.514	548305.2053	6	
38	0.5889	0.514	548305.2053	11	
39	0.5889	0.514	548305.2053	8	
40	0.5889	0.514	548305.2053	12	
41	0.5889	0.514	548305.2053	12	
42	0.5889	0.514	548305.2053	9	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
43	0.5889	0.514	548305.2053	10	
44	0.5889	0.514	548305.2053	10	
45	0.5889	0.514	548305.2053	10	
46	0.5889	0.514	548305.2053	11	
47	0.5889	0.514	548305.2053	10	
48	0.5889	0.514	548305.2053	10	
49	0.5889	0.514	548305.2053	11	
50	0.5889	0.514	548305.2053	10	
51	0.5889	0.514	548305.2053	7	
52	0.5889	0.514	548305.2053	12	
53	0.5889	0.514	548305.2053	10	
54	0.5889	0.514	548305.2053	11	
55	0.5889	0.514	548305.2053	12	
56	0.5889	0.514	548305.2053	11	
57	0.5889	0.514	548305.2053	10	
58	0.5889	0.514	548305.2053	10	
59	0.5889	0.514	548305.2053	12	
60	0.5889	0.514	548305.2053	9	
61	0.5889	0.514	548305.2053	8	
62	0.5889	0.514	548305.2053	11	
63	0.5889	0.514	548305.2053	7	
64	0.5889	0.514	548305.2053	3	
65	0.5889	0.514	548305.2053	10	
66	0.5889	0.514	548305.2053	6	
67	0.5889	0.514	548305.2053	8	
68	0.5889	0.514	548305.2053	9	
69	0.5889	0.514	548305.2053	8	
70	0.5889	0.514	548305.2053	12	
71	0.5889	0.514	548305.2053	10	
72	0.5889	0.514	548305.2053	8	
73	0.5889	0.514	548305.2053	8	
74	0.5889	0.514	548305.2053	11	
75	0.5889	0.514	548305.2053	4	
76	0.5889	0.514	548305.2053	9	
77	0.5889	0.514	548305.2053	12	
78	0.5889	0.514	548305.2053	7	
79	0.5889	0.514	548305.2053	9	
80	0.5889	0.514	548305.2053	8	
81	0.5889	0.514	548305.2053	11	
82	0.5889	0.514	548305.2053	10	
83	0.5889	0.514	548305.2053	11	
84	0.5889	0.514	548305.2053	11	
85	0.5889	0.514	548305.2053	11	
86	0.5889	0.514	548305.2053	8	
87	0.5889	0.514	548305.2053	11	
88	0.5889	0.514	548305.2053	4	
89	0.5889	0.514	548305.2053	11	
90	0.5889	0.514	548305.2053	10	
91	0.5889	0.514	548305.2053	12	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
92	0.5889	0.514	548305.2053	12	
93	0.5889	0.514	548305.2053	11	
94	0.5889	0.514	548305.2053	9	
95	0.5889	0.514	548305.2053	10	
96	0.5889	0.514	548305.2053	12	
97	0.5889	0.514	548305.2053	10	
98	0.5889	0.514	548305.2053	9	
99	0.5889	0.514	548305.2053	11	
100	0.5889	0.514	548305.2053	7	

Then the clustering procedure with normalisation of all attributes has been applied to the data set. Both procedures with and without normalisation have been applied 100 times to the data set (see Tables 3.5 and 3.6).

Table 3.6: Clustering of the Heart Diseases data set with normalisation of the attributes.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.8259	0.7114	1868.3564	4	
2	0.8259	0.7114	1868.3564	7	
3	0.8259	0.7114	1868.3564	6	
4	0.8259	0.7114	1868.3564	5	
5	0.8259	0.7114	1868.3564	5	
6	0.8259	0.7114	1868.3564	5	
7	0.8259	0.7114	1868.3564	6	
8	0.8222	0.7066	1868.3293	6	
9	0.8185	0.7018	1868.3446	4	
10	0.8185	0.7018	1868.3446	5	
11	0.8185	0.7018	1868.3446	5	
12	0.8148	0.6971	1836.1406	3	
13	0.8074	0.6878	1794.5934	4	
14	0.8074	0.6878	1794.5934	7	
15	0.8074	0.6878	1794.5934	6	
16	0.8074	0.6878	1794.5934	6	
17	0.8074	0.6878	1794.5934	3	
18	0.8074	0.6878	1794.5934	4	
19	0.8074	0.6878	1794.5934	6	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
20	0.8074	0.6878	1794.5934	2	
21	0.8074	0.6878	1794.5934	7	
22	0.8074	0.6878	1794.5934	5	
23	0.8074	0.6878	1794.5934	4	
24	0.8074	0.6878	1794.5934	4	
25	0.8074	0.6878	1794.5934	6	
26	0.8074	0.6878	1794.5934	5	
27	0.8074	0.6878	1794.5934	4	
28	0.8074	0.6878	1794.5934	5	
29	0.8074	0.6878	1794.5934	9	
30	0.8074	0.6878	1794.5934	4	
31	0.8074	0.6878	1794.5934	7	
32	0.8074	0.6878	1794.5934	8	
33	0.8074	0.6878	1794.5934	7	
34	0.8074	0.6878	1794.5934	5	
35	0.8074	0.6878	1794.5934	6	
36	0.8074	0.6878	1794.5934	5	
37	0.8074	0.6878	1794.5934	5	
38	0.8074	0.6878	1794.5934	6	
39	0.8074	0.6878	1794.5934	7	
40	0.8074	0.6878	1794.5934	4	
41	0.8074	0.6878	1794.5934	4	
42	0.8074	0.6878	1794.5934	4	
43	0.8074	0.6878	1794.5934	6	
44	0.8074	0.6878	1794.5934	5	
45	0.8074	0.6878	1794.5934	7	
46	0.8074	0.6878	1794.5934	3	
47	0.8074	0.6878	1794.5934	6	
48	0.8074	0.6878	1794.5934	6	
49	0.8074	0.6878	1794.5934	6	
50	0.8074	0.6878	1794.5934	5	
51	0.8074	0.6878	1794.5934	4	
52	0.8074	0.6878	1794.5934	4	
53	0.8074	0.6878	1794.5934	3	
54	0.8074	0.6878	1794.5934	7	
55	0.8074	0.6878	1794.5934	6	
56	0.8037	0.6833	1794.5534	13	+
57	0.8037	0.6833	1794.5534	4	+
58	0.8037	0.6833	1794.5534	8	+
59	0.8037	0.6833	1794.5534	3	+
60	0.8037	0.6833	1794.5534	6	+
61	0.8037	0.6833	1794.5534	13	+
62	0.8037	0.6833	1794.5534	4	+
63	0.8037	0.6833	1794.5534	5	+
64	0.8037	0.6833	1794.5534	8	+
65	0.8037	0.6833	1794.5534	4	+
66	0.8037	0.6833	1794.5534	8	+
67	0.8037	0.6833	1794.5534	7	+
68	0.8037	0.6833	1794.5534	4	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
69	0.8037	0.6833	1794.5534	14	+
70	0.8037	0.6833	1794.5534	4	+
71	0.8037	0.6833	1794.5534	4	+
72	0.8037	0.6833	1794.5534	9	+
73	0.8037	0.6833	1794.5534	8	+
74	0.8037	0.6833	1794.5534	13	+
75	0.8037	0.6833	1794.5534	3	+
76	0.8037	0.6833	1794.5534	5	+
77	0.8037	0.6833	1794.5534	4	+
78	0.8037	0.6833	1794.5534	4	+
79	0.8037	0.6833	1794.5534	5	+
80	0.8037	0.6833	1794.5534	4	+
81	0.7963	0.6744	1859.3437	4	
82	0.7963	0.6744	1859.3437	6	
83	0.7963	0.6744	1859.3437	12	
84	0.7963	0.6744	1859.3437	6	
85	0.7963	0.6744	1859.3437	4	
86	0.7963	0.6744	1859.3437	4	
87	0.7963	0.6744	1859.3437	2	
88	0.7963	0.6744	1859.3437	6	
89	0.7963	0.6744	1859.3437	12	
90	0.7963	0.6744	1859.3437	4	
91	0.7963	0.6744	1859.3437	12	
92	0.7963	0.6744	1859.3437	3	
93	0.5593	0.5052	2056.4629	7	
94	0.5556	0.5043	2056.6104	4	
95	0.5259	0.4995	2020.4735	6	
96	0.5222	0.4991	2142.8896	3	
97	0.5222	0.4991	2020.4696	5	
98	0.5222	0.4991	2020.4696	8	
99	0.5185	0.4988	2144.1491	6	
100	0.5074	0.4983	2153.6535	5	

The "true clustering" ($Acc = 1$) has not been obtained. The obtained average accuracy value 0.590074 in the case without normalisation of the data has been considerably smaller than the obtained average accuracy value 0.784741 in the case with normalisation of the data. After application of the normalisation procedure to the data set the Rand index has increased from 0.5154 to 0.6833. The latter value has been taken not as the best value of the Rand index but in accordance with the

obtained best objective function value (because this is the criterion of the unsupervised objective function-based clustering).

3.6.4 Credit Approval Data Set

The credit approval data set has 690 records ($N = 690$) with 16 attributes. There are 6 numerical attributes, while the rest of the attributes are categorical. Clustering has been performed using the last categorical attribute that takes two values: + and,-, i.e. approval of the credit and rejection of the application. First the clustering procedure has been applied to the data set without normalisation of the data (Table 3.7).

Table 3.7: Clustering of the Credit Approval data set without normalisation.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.5528	0.5048	4897673526	7	+
2	0.5528	0.5048	4897673526	5	+
3	0.5528	0.5048	4897673526	7	+
4	0.5528	0.5048	4897673526	5	+
5	0.5528	0.5048	4897673526	5	+
6	0.5528	0.5048	4897673526	7	+
7	0.5528	0.5048	4897673526	5	+
8	0.5528	0.5048	4897673526	7	+
9	0.5528	0.5048	4897673526	6	+
10	0.5528	0.5048	4897673526	7	+
11	0.5528	0.5048	4897673526	6	+
12	0.5528	0.5048	4897673526	4	+
13	0.5528	0.5048	4897673526	2	+
14	0.5528	0.5048	4897673526	7	+
15	0.5528	0.5048	4897673526	6	+
16	0.5528	0.5048	4897673526	6	+
17	0.5528	0.5048	4897673526	7	+
18	0.5528	0.5048	4897673526	6	+
19	0.5528	0.5048	4897673526	7	+
20	0.5528	0.5048	4897673526	6	+
21	0.5528	0.5048	4897673526	7	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
22	0.5528	0.5048	4897673526	7	+
23	0.5528	0.5048	4897673526	6	+
24	0.5528	0.5048	4897673526	6	+
25	0.5528	0.5048	4897673526	7	+
26	0.5528	0.5048	4897673526	7	+
27	0.5528	0.5048	4897673526	7	+
28	0.5528	0.5048	4897673526	7	+
29	0.5528	0.5048	4897673526	6	+
30	0.5528	0.5048	4897673526	7	+
31	0.5528	0.5048	4897673526	7	+
32	0.5528	0.5048	4897673526	7	+
33	0.5528	0.5048	4897673526	7	+
34	0.5528	0.5048	4897673526	6	+
35	0.5528	0.5048	4897673526	6	+
36	0.5528	0.5048	4897673526	3	+
37	0.5528	0.5048	4897673526	6	+
38	0.5528	0.5048	4897673526	7	+
39	0.5528	0.5048	4897673526	6	+
40	0.5528	0.5048	4897673526	7	+
41	0.5528	0.5048	4897673526	5	+
42	0.5528	0.5048	4897673526	7	+
43	0.5528	0.5048	4897673526	7	+
44	0.5528	0.5048	4897673526	7	+
45	0.5528	0.5048	4897673526	5	+
46	0.5528	0.5048	4897673526	6	+
47	0.5528	0.5048	4897673526	6	+
48	0.5528	0.5048	4897673526	6	+
49	0.5528	0.5048	4897673526	6	+
50	0.5528	0.5048	4897673526	7	+
51	0.5528	0.5048	4897673526	7	+
52	0.5528	0.5048	4897673526	7	+
53	0.5528	0.5048	4897673526	6	+
54	0.5528	0.5048	4897673526	6	+
55	0.5528	0.5048	4897673526	3	+
56	0.5528	0.5048	4897673526	7	+
57	0.5528	0.5048	4897673526	5	+
58	0.5528	0.5048	4897673526	7	+
59	0.5528	0.5048	4897673526	7	+
60	0.5528	0.5048	4897673526	7	+
61	0.5528	0.5048	4897673526	6	+
62	0.5528	0.5048	4897673526	7	+
63	0.5528	0.5048	4897673526	6	+
64	0.5528	0.5048	4897673526	6	+
65	0.5528	0.5048	4897673526	6	+
66	0.5528	0.5048	4897673526	6	+
67	0.5528	0.5048	4897673526	6	+
68	0.5528	0.5048	4897673526	6	+
69	0.5528	0.5048	4897673526	6	+
70	0.5528	0.5048	4897673526	6	+

N	Accuracy	Rand Index	Objective function J	Iterations	BF
71	0.5528	0.5048	4897673526	7	+
72	0.5528	0.5048	4897673526	7	+
73	0.5528	0.5048	4897673526	6	+
74	0.5528	0.5048	4897673526	7	+
75	0.5528	0.5048	4897673526	6	+
76	0.5528	0.5048	4897673526	7	+
77	0.5528	0.5048	4897673526	6	+
78	0.5528	0.5048	4897673526	6	+
79	0.5528	0.5048	4897673526	6	+
80	0.5528	0.5048	4897673526	6	+
81	0.5528	0.5048	4897673526	7	+
82	0.5528	0.5048	4897673526	1	+
83	0.5528	0.5048	4897673526	6	+
84	0.5528	0.5048	4897673526	6	+
85	0.5528	0.5048	4897673526	6	+
86	0.5528	0.5048	4897673526	4	+
87	0.5528	0.5048	4897673526	6	+
88	0.5528	0.5048	4897673526	7	+
89	0.5528	0.5048	4897673526	7	+
90	0.5528	0.5048	4897673526	7	+
91	0.5528	0.5048	4897673526	7	+
92	0.5528	0.5048	4897673526	7	+
93	0.5528	0.5048	4897673526	6	+
94	0.5528	0.5048	4897673526	6	+
95	0.5528	0.5048	4897673526	6	+
96	0.5528	0.5048	4897673526	6	+
97	0.5528	0.5048	4897673526	6	+
98	0.5528	0.5048	4897673526	7	+
99	0.5528	0.5048	4897673526	7	+
100	0.5528	0.5048	4897673526	7	+

Then the clustering procedure with normalisation of all attributes has been applied to the data set (Table 3.8). Both procedures with and without normalisation have been applied 100 times to the data set.

Table 3.8: Clustering of the Credit Approval data set with normalisation.

N	Accuracy	Rand Index	Objective function J	Iterations	BF
1	0.8239	0.7094	5429.256	3	
2	0.8239	0.7094	5429.256	5	
3	0.8239	0.7094	5429.256	4	
4	0.8239	0.7094	5430.4593	5	
5	0.8239	0.7094	5430.4593	4	
6	0.807	0.6881	5346.6759	6	
7	0.807	0.6881	5346.6759	7	
8	0.807	0.6881	5346.6759	3	
9	0.807	0.6881	5346.6759	4	
10	0.807	0.6881	5346.6759	7	
11	0.807	0.6881	5346.6759	9	
12	0.807	0.6881	5346.6759	6	
13	0.807	0.6881	5346.6759	7	
14	0.807	0.6881	5346.6759	7	
15	0.807	0.6881	5346.6759	6	
16	0.807	0.6881	5346.6759	7	
17	0.807	0.6881	5346.6759	4	
18	0.807	0.6881	5346.6759	6	
19	0.807	0.6881	5346.6759	6	
20	0.807	0.6881	5346.6759	4	
21	0.807	0.6881	5346.6759	4	
22	0.807	0.6881	5346.6759	7	
23	0.8025	0.6825	5342.7894	4	
24	0.8025	0.6825	5342.7894	6	
25	0.8025	0.6825	5342.7894	6	
26	0.8025	0.6825	5342.7894	3	
27	0.8025	0.6825	5342.7894	5	
28	0.8025	0.6825	5342.7894	4	
29	0.8025	0.6825	5342.7894	7	
30	0.8025	0.6825	5342.7894	4	
31	0.8025	0.6825	5342.7894	4	
32	0.8025	0.6825	5342.7894	5	
33	0.8025	0.6825	5342.7894	7	
34	0.8025	0.6825	5342.7894	6	
35	0.8025	0.6825	5342.7894	9	
36	0.8025	0.6825	5342.7894	6	
37	0.8025	0.6825	5342.7894	6	
38	0.8025	0.6825	5342.7894	5	
39	0.8025	0.6825	5342.7894	6	
40	0.8025	0.6825	5342.7894	6	
41	0.8025	0.6825	5342.7894	6	
42	0.8025	0.6825	5342.7894	4	
43	0.8025	0.6825	5342.7894	3	
44	0.8025	0.6825	5342.7894	5	
45	0.8025	0.6825	5342.7894	3	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
46	0.8025	0.6825	5342.7894	6	
47	0.8025	0.6825	5342.7894	5	
48	0.8009	0.6806	5342.7538	4	+
49	0.8009	0.6806	5342.7538	4	+
50	0.8009	0.6806	5342.7538	3	+
51	0.8009	0.6806	5342.7538	3	+
52	0.8009	0.6806	5342.7538	5	+
53	0.8009	0.6806	5342.7538	3	+
54	0.7948	0.6733	5438.2405	5	
55	0.7933	0.6715	5438.236	16	
56	0.7933	0.6715	5438.236	7	
57	0.7933	0.6715	5438.236	5	
58	0.7933	0.6715	5438.236	8	
59	0.7933	0.6715	5438.236	9	
60	0.7933	0.6715	5438.236	8	
61	0.7933	0.6715	5438.236	7	
62	0.6861	0.5686	5563.2453	3	
63	0.6784	0.563	5563.2019	4	
64	0.6784	0.563	5564.1337	4	
65	0.6432	0.5403	5536.5778	4	
66	0.6432	0.5403	5536.5778	4	
67	0.6432	0.5403	5536.5778	3	
68	0.5482	0.5039	5901.8234	4	
69	0.5482	0.5039	5562.1779	3	
70	0.5482	0.5039	5562.1779	4	
71	0.5482	0.5039	5562.1779	2	
72	0.5482	0.5039	5562.1779	3	
73	0.5482	0.5039	5562.1779	2	
74	0.5482	0.5039	5562.1779	3	
75	0.5482	0.5039	5562.1779	3	
76	0.5482	0.5039	5562.1779	3	
77	0.5482	0.5039	5562.1779	3	
78	0.5467	0.5036	5561.1694	3	
79	0.5467	0.5036	5561.1694	3	
80	0.5467	0.5036	5561.1694	3	
81	0.5467	0.5036	5561.1694	2	
82	0.5467	0.5036	5561.1694	2	
83	0.5467	0.5036	5561.1694	3	
84	0.5467	0.5036	5561.1694	3	
85	0.5467	0.5036	5561.1694	2	
86	0.5467	0.5036	5561.1694	2	
87	0.5467	0.5036	5561.1694	4	
88	0.5467	0.5036	5561.1694	4	
89	0.5283	0.5008	5973.6835	4	
90	0.5283	0.5008	5973.6835	4	
91	0.5115	0.4995	6077.557	1	
92	0.5115	0.4995	6077.557	3	
93	0.5115	0.4995	6077.557	3	
94	0.5115	0.4995	6077.557	5	

N	Accuracy	Rand Index	Objective function J	Iterations	BF
95	0.5115	0.4995	6077.557	3	
96	0.51	0.4994	6078.3273	3	
97	0.51	0.4994	6078.3273	2	
98	0.51	0.4994	6078.3273	4	
99	0.51	0.4994	6078.3273	1	
100	0.51	0.4994	6078.3273	3	

The results of all 100 runs of the procedure without normalisation were the same and therefore equal to the average accuracy 0.552833 and the value of the cost function is 4897673525.5238. After the runs of the procedure with normalisation, the average accuracy increased to 0.706861 (see Table 3.8). After normalisation, the Rand index increased from 0.5048 to 0.6806. Again the latter value has been taken not as the best value of the Rand index but in accordance with the obtained best objective function value (because this is the criterion of the unsupervised objective function-based clustering).

3.7 Summary

In the overwhelming majority of the earlier approaches to normalisation, scaling was used for numerical attributes when the Euclidean metric was used for measuring dissimilarity between attributes. It was also often assumed that the variables have the normal distribution. These normalisation approaches were applied mainly to assure the values being in the [0, 1] range. However, it has been shown that in general this does not provide equal contributions of the features to the metrics. It was also

suggested often to truncate the out-of-range components and this could lead to loss of information from the data set.

In the general case of feature extraction, when there is no *a priory* information about preferences of some attributes, one has to assume that all attributes are equally important. A direct application of geometric measures to attributes with large ranges will implicitly assign bigger contributions to the metrics than those of attributes with small ranges. If all attributes are equally important to measure similarity between feature vectors then one should not use distance measures like the Euclidean distance (3.1), the matching dissimilarity measure (3.20) and their combination without normalisation of data.

These arguments have been used to support the proposed unified statistical approach that has to be applied to normalise all attributes of the feature vectors of mixed data sets. To obtain a new normalised metric, one should calculate the mean contribution of each attribute to the metric and to divide the attribute in all records by this mean. Estimators are used to calculate the mean contributions.

Evidently, if the mean is equal to zero then this attribute should be removed from the feature vector. The means of contributions of all attributes in all considered cases are the same and hence, contributions of the features to similarity measures are approximately equalized. Such a normalisation is achieved by scaling the numerical attributes, while the categorical attributes are normalised by appropriate choice of their weights.

If one knows a priory that some attributes have bigger contributions to similarity measures than the rest of the attributes then this can be taken into account by appropriate weighting of attributes. It looks quite natural to apply the weighting procedure to metrics that have already been normalised by the above described procedure.

The new normalised metrics has been used for clustering numerical, categorical and mixed data. The k -prototypes algorithm that earlier was applied for a non-normalised metric (Huang, 1998), has been employed. It has been shown that normally the accuracy increased when clustering is performed using normalised metrics. These examples have demonstrated the advantages of the introduced normalised metrics.

Chapter 4

Clustering mixed data sets (Minkowski metric)

In this chapter, a new statistical approach introduced in Chapter 3 is developed further. The new approach is applied to the case of the Minkowski metrics being used as a measure for continuous numerical features, while to deal with categorical attributes again the matching dissimilarity measure is used. Various mathematical problems related to the normalisation of mixed metrics are resolved. The introduced metrics are applied to some data sets where it is more advantageous to apply the general Minkowski metrics (including the Tchebysheff and city-block metrics) instead of a particular case $p_M = 2$ (the Euclidean metric).

4.1. Background

In clustering analysis of numerical data sets, often not only the Euclidean metric (distance) ρ_E (or L_2) but other similarity measures are also used. For example, city block distance (or L_1 metric)

$$\rho_1(\mathbf{x}_1, \mathbf{x}_2) = \| \mathbf{x}_1 - \mathbf{x}_2 \|_1 = \sum_{j=1}^{p} |x_{1j} - x_{2j}| \tag{4.1}$$

the Minkowski distance ρ_{p_M} (or L_p metric)

$$\rho_{p_M}(\mathbf{x}_1,\mathbf{x}_2)=\|\mathbf{x}_1-\mathbf{x}_2\|_{p_M}=\left(\sum_{j=1}^{p}|x_{1j}-x_{2j}|^{p_M}\right)^{1/p_M} \tag{4.2}$$

where p_M is a positive number, $1\le p_M<+\infty$; and the Tchebysheff (Chebyshev) or maximum norm metric

$$\rho_{max}(\mathbf{x}_1,\mathbf{x}_2)=\|\mathbf{x}_1-\mathbf{x}_2\|_{max}=\max_j|x_{1j}-x_{2j}| \tag{4.3}$$

The Euclidean metric (3.1) and city block distance (4.1) are particular cases of the Minkowski metric for $p_M=2$ and $p_M=1$ respectively. The Tchebysheff metric can be obtained from the Minkowski metric as the following limit $p_M\to\infty$. Other metrics are also applied to numerical data sets.

As it has been argued in the previous chapter, if there is no *a priory* information about preferences of some attributes, one has to assume that all attributes are equally important, and hence to assume that the average contribution of the j-th feature component to the total measure is equal to its mean. Therefore, the goal of a normalisation procedure is the equalisation of the attribute contributions. Applying the same unified statistical treatment to both numerical and categorical features of mixed data sets, as it has been used in the previous chapter, new normalised metrics are introduced.

In this Chapter a rigorous statistical approach to data sets is used and various mathematical problems related to the normalisation of mixed metrics are resolved.

Mathematically rigorous treatment of the normalisation procedure is presented and examples of normalised metrics are given in an explicit way. In addition, the proposed approach is extended to the case of mixed metrics, i.e. when different metrics are used for numerical and categorical data respectively.

4.2 Statistical approach to normalisation of the Minkowski metric (numerical attributes)

To obtain a new normalised metric in the general case of the Minkowski metric (4.2), one should calculate the mean contribution of each j-th attribute to the metric $E|X_{1j}-X_{2j}|^{p_M}$ (here E means the expectation of a variable) and to divide the attribute in all records by this mean (if the mean is equal to zero then this attribute should be removed from the feature vector). Hence, the normalised Minkowski metric can be introduced in the following way

$$\rho^{*}_{p_M}(\mathbf{x}_1,\mathbf{x}_2)=\left(\sum_{j=1}^{p}\alpha_j|x_{1j}-x_{2j}|^{p_M}\right)^{1/p_M} \tag{4.4}$$

where $\alpha_j=1/E|X_{1j}-X_{2j}|^{p_M}$, X_{1j} and X_{2j} are independent random variables whose values are distributed in accordance with the distribution of the j-th attribute.

In the general case, the distribution of the j-th attribute is not known in advance, therefore, to estimate the expectation $E|X_{1j}-X_{2j}|^{p_M}$ we can use the sample mean

$$\hat{E}|X_{1j}-X_{2j}|^{p_M}=\frac{1}{N^2}\sum_{r,s=1}^{N}|x_{rj}-x_{sj}|^{p_M} \tag{4.5}$$

The estimation (4.5) is a biased estimator of $E|X_{1j}-X_{2j}|^{p_M}$, hence for small data sets it is better to use the following estimation

$$\hat{E}|X_{1j}-X_{2j}|^{p_M}=\frac{2}{N(N-1)}\sum_{1\le r<s\le N}|x_{rj}-x_{sj}|^{p_M} \tag{4.6}$$

that is an unbiased estimator.

It follows from Proposition B1 (Appendix B) that (4.5) and (4.6) are consistent estimators.

Comment. For $p_M=2$ the above results agree with the results obtained for the Euclidean metric (see Chapter 3).

If the data set is large, i.e. the record number N is large, then it is rather difficult to use the formula (4.5) or it cannot be used at all because the sum has N^2 components and, hence, its calculation is rather time consuming. For example, an average size data set may have $N=10^6$. In this case, the number of operations for calculating just

one sum is $N^2 = 10^{12}$. To calculate $E\left|X_{1j} - X_{2j}\right|^{p_M}$ one can use the following approach.

Often some of the real numerical features are normally distributed. Let us consider now the case when one knows in advance that the values of the j-th attribute are distributed normally. In this case, there is a quite attractive property of ρ_{p_M}. Let $f_{A_j''}(x)$ be a normal distribution with mean μ_j and variance σ_j^2. The probability distribution or density function for a normal distribution has the following formula:

$$f_{A_j''}(x) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right).$$

One can assume that X_{1j} and X_{2j} are independent random variables having the same normal distribution with mean μ_j and variance σ_j^2. Then the random variable $X_{1j} - X_{2j}$ has also a normal distribution with mean 0 and dispersion $2\sigma_j^2$. For this random variable, one can estimate $E\,|\,X_{1j} - X_{2j}\,|^{p_M}$ using the following formula

$$E\,|\,X_{1j} - X_{2j}\,|^{p_M} = \frac{1}{\sqrt{2\pi}\sqrt{2}\sigma_j} \int_{-\infty}^{+\infty} |t|^{p_M} \exp\left(-\frac{t^2}{4\sigma_j^2}\right) dt = \frac{1}{\sqrt{\pi}\sigma_j} \int_0^{\infty} |t|^{p_M} \exp\left(-\frac{t^2}{4\sigma_j^2}\right) dt.$$

Denoting

$$u = \frac{t^2}{4\sigma_j^2},$$

and substituting this expression into the above formula for mathematical expectation $E|X_{1j}-X_{2j}|^{p_M}$, one obtains

$$E|X_{1j}-X_{2j}|^{p_M}=\frac{(2\sigma_j)^{p_M}}{\sqrt{\pi}}\int_0^{\infty}u^{\frac{p_M+1}{2}-1}e^{-u}du=\frac{(2\sigma_j)^{p_M}}{\sqrt{\pi}}\Gamma\left(\frac{p_M+1}{2}\right).$$

where

$$\Gamma(x)=\int_0^{\infty}t^{x-1}e^{-t}dt$$

is the Euler gamma function. Thus, one has

$$E|X_{1j}-X_{2j}|^{p_M}=\frac{(2\sigma_j)^{p_M}}{\sqrt{\pi}}\Gamma\left(\frac{p_M+1}{2}\right).$$

The gamma function can be calculated using the standard algorithms.

4.3 Normalisation of metrics for data sets with mixed attributes

For data sets with categorical attributes, it is possible to introduce different metrics (see, e.g. Gibert and Cortes, 1997; Huang, 1998; Ralambondrainy, 1995). One of the most cited variants of metrics is studied here (see, e.g. Huang, 1998), namely the distance between two categorical feature vectors $\mathbf{y}_1=(y_{11},\ldots,y_{1l})$ and $\mathbf{y}_2=(y_{21},\ldots,y_{2l})$ is defined as:

$$\rho_{cat}(\mathbf{y}_1,\mathbf{y}_2)=\omega(y_{11},y_{21})+\ldots+\omega(y_{1l},y_{2l}) \tag{4.7}$$

were

$$\omega(y_{1j},y_{2j})=\begin{cases}0 & \text{for} \quad y_{1j}=y_{2j}\\ 1 & \text{for} \quad y_{1j}\neq y_{2j}\end{cases}$$

Evidently, the metric (4.7) in degree p_M is

$$\rho_{cat}^{p_M}(\mathbf{y}_1,\mathbf{y}_2)=\omega^{p_M}(y_{11},y_{21})+\ldots+\omega^{p_M}(y_{1l},y_{2l}) \tag{4.8}$$

Let us extend the results obtained for the Euclidean metric to the general case of the Minkowski metric. It follows from the Minkowski inequality that the following function is a metric:

$$\rho((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2))= \\ \left(|x_{11}-x_{21}|^{p_M}+\ldots+|x_{1p}-x_{2p}|^{p_M}+\omega^{p_M}(y_{11},y_{21})+\ldots+\omega^{p_M}(y_{1l},y_{2l})\right)^{1/p_M} \tag{4.9}$$

It will be called the Minkowski mixed p_M -metric.

In fact, we have to prove that the sum of the Minkowski metric for numerical attributes and the matching dissimilarity measure for categorical attributes is a metric. One can see from direct checking that since both metrics are non-negative and symmetric their sum is also non-negative and symmetric. Hence, we need to prove only the triangle inequality for the sum. As it has been mentioned in Chapter 2, the Minkowski inequality is the triangle inequality in L^p spaces:

$$\|f+g\|_{p_M} \le \|f\|_{p_M} + \|g\|_{p_M}$$

The normalisation of the p_M -metric (4.9) is fulfilled in the same way as the normalisation of the Euclidean mixed metric (see Chapter 3)

$$\rho^*((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2)) = \left[\sum_{j=1}^{p}\alpha_j \mid x_{1j}-x_{2j}\mid^{p_M} + \sum_{j=1}^{l}\beta_j \omega^{p_M}(y_{1j},y_{2j})\right]^{1/p_M} \tag{4.10}$$

where $\alpha_j = 1/E\mid X_{1j}-X_{2j}\mid^{p_M}$ and $\beta_j = 1/E\omega^{p_M}(Y_{1j},Y_{2j})$. Note that β_j are calculated in the same way as in (3.24) because $E\omega^{p_M}(Y_{1j},Y_{2j}) = E\omega(Y_{1j},Y_{2j})$.

If the distribution of the attributes is unknown then to calculate α_j one can use the estimation (4.6), and to estimate $E\omega(Y_{1j},Y_{2j})$ one can use the sampling mean

$$\hat{E}\omega^{p_M}(Y_{1j},Y_{2j}) = \frac{1}{N^2}\sum_{r,s=1}^{N}\omega(y_{rj},y_{sj}) \tag{4.11}$$

The estimation (4.11) is a biased estimator of $E\omega^{p_M}(Y_{1j},Y_{2j})$, hence for small data sets it is better to use the following estimation

$$\hat{E}\omega^{p_M}(Y_{1j},Y_{2j}) = \frac{2}{N(N-1)}\sum_{1\le r<s\le N}\omega(y_{rj},y_{sj}), \tag{4.12}$$

which is an unbiased estimator. It will follow from Proposition B1 (see Appendix B) that (4.11) and (4.12) are consistent estimators.

4.4 A general algorithm for normalisation of mixed metrics

Let a mixed metric ρ be a sum of two metrics ρ_1 and ρ_2:

$$\rho((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2)) = \rho_1(\mathbf{x}_1,\mathbf{x}_2) + \rho_2(\mathbf{y}_1,\mathbf{y}_2)$$

The former metric is for numerical attributes and the latter metric is for categorical attributes.

To normalise the mixed metric in this general case, one needs first to normalise metrics ρ_1 and ρ_2, i.e. one needs to find ρ_1^* and ρ_2^*, and then the normalised general mixed metric is:

$$\rho^*((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2)) = \alpha_1\rho_1^*(\mathbf{x}_1,\mathbf{x}_2) + \alpha_2\rho_2^*(\mathbf{y}_1,\mathbf{y}_2).$$

4.5 Clustering algorithms based on Minkowski metrics

4.5.1. Algorithm

As it has been noted, the use of the sum of squares of the Euclidean distances as the objective function has the advantage of having a simple formula for recalculating new values of cluster prototypes (see Section 3.2). This advantage was implemented in the k-means and k-prototypes algorithms. Evidently, these algorithms cannot be used in the case of a general Minkowski metric. Therefore, it was suggested by Miyamoto and Augusta (1996) and Hathaway et. al. (2000) to use instead of (3.2) the following objective function J_{j,p_M} for a generalisation of the fuzzy clustering objective function

$$J_{j,p_M} = \sum_{m=1}^{k}\sum_{i=1}^{N} u_{mi}^{j} \left\| \mathbf{X}_i - \mathbf{Q}_m \right\|_{p_M}^{p_M} \tag{4.13}$$

where $j > 1$ is the exponent of the fuzzy algorithm. We employ a similar to (4.13) objective function in order to use it with the Minkowski distances.

4.5.2. Clustering using Minkowski metrics

Let us define an objective function

$$J_{p_M} = \sum_{m=1}^{k}\sum_{i=1}^{N} u_{im} (\rho_{p_M})^{p_M} (\mathbf{X}_i, \mathbf{Q}_m) \tag{4.14}$$

where $u_{im} \in \{0,1\}$, $1 \le i \le N$, $1 \le m \le k$,

$$\sum_{m=1}^{k} u_{im} = 1, \quad \sum_{i=1}^{N} u_{im} > 0. \tag{4.15}$$

and $p_M \ge 1$.

For $p_M = 2$, the k-means algorithm can be employed for clustering. At each iteration, this algorithm recalculates the prototypes for each of the clusters obtained at the previous iteration, and then the vectors of records are split again in the new clusters depending on what of new prototypes is the closest to a particular record in accordance with the metric. The same approach can be used for clustering using an objective function for an arbitrary $p_M \ge 1$.

Indeed, let us write the objective function (4.14) as

$$J_{p_M} = \sum_{m=1}^{k} \sum_{i \in C_m} (\rho_{p_M})^{p_M} (X_i, Q_m) \tag{4.16}$$

where C_m is the set of all indexes i, $(1 \le i \le N)$ such that the i-th record belongs to the m-th cluster. Now denote Φ_m as

$$\Phi_m = \sum_{i \in C_m} (\rho_{p_M})^{p_M} (X_i, Q_m) \tag{4.17}$$

For every m, let us find a new prototype $\tilde{Q}_m$ such that the sum Φ_m is minimum. Then let us split all record vectors into clusters $\tilde{C}_m$ $(1 \le m \le k)$ in accordance with

the proximity of a prototype $\tilde{Q}_m \in (\tilde{Q}_1,\ldots,\tilde{Q}_k)$ to the record under consideration. Since

$$\sum_{m=1}^{k}\sum_{i\in\tilde{C}_m}(\rho_{p_M})^{p_M}(X_i,\tilde{Q}_m) \le \sum_{m=1}^{k}\sum_{i\in C_m}(\rho_{p_M})^{p_M}(X_i,\tilde{Q}_m)$$

and for every m, we have

$$\sum_{i\in C_m}(\rho_{p_M})^{p_M}(X_i,\tilde{Q}_m) \le \sum_{i\in C_m}(\rho_{p_M})^{p_M}(X_i,Q_m),$$

then we obtain

$$\sum_{m=1}^{k}\sum_{i\in\tilde{C}_m}(\rho_{p_M})^{p_M}(X_i,\tilde{Q}_m) \le \sum_{m=1}^{k}\sum_{i\in C_m}(\rho_{p_M})^{p_M}(X_i,Q_m). \qquad (4.18)$$

It follows from the inequality (4.18) that the value of the objective function at each iteration would not increase and the iterative process converges to a local minimum of the objective function.

Thus, for a successful use of the algorithm, one needs to find effectively new prototypes $\tilde{Q}_m$ such that the sum (4.17) is minimum. As it has been mentioned in Section 3.2, this problem is very simple for $p_M = 2$ in the case of numerical data because

$$\Phi_m = \sum_{i \in C_m} (\rho_2)^2 (X_i, Q_m) = \sum_{i \in C_m} \sum_{j=1}^{p} (X_{ij} - Q_{mj})^2$$

and therefore, the minimum is at

$$\frac{\partial \Phi_m}{\partial Q_{mj}} = -2 \sum_{i \in C_m} (X_{ij} - Q_{mj}) = 0, \quad (j = 1, \ldots, p). \tag{4.19}$$

Solving the system (4.19), we obtain

$$\mathbf{Q}_{mj} = \frac{1}{|C_m|} \sum_{i \in C_m} \mathbf{X}_{ij} \tag{4.20}$$

where $|C_m|$ is the number of records in the m-th cluster. The formula (4.20) is well known for recalculating the prototypes in the classic k-means algorithm.

Now we need to consider the cases $p_M \neq 2$. As it has been mentioned, a similar objective function for fuzzy clustering was considered by Hathaway et .al.(2000). They suggested also an approach for recalculating the prototypes for these cases. However, some very important details of the algorithm were not described. Hence, we need to discuss the algorithm for recalculating the prototypes in detail and apply it for hard clustering.

In the case under consideration, we have

$$\Phi_m = \sum_{i \in C_m} (\rho_{p_M})^{p_M} (X_i, Q_m) = \sum_{i \in C_m} \sum_{j=1}^{p} |X_{ij} - Q_{mj}|^{p_M} = \sum_{j=1}^{p} \sum_{i \in C_m} |X_{ij} - Q_{mj}|^{p_M}$$

If we denote

$$\Phi_{mj}(t) = \sum_{i \in C_m} |X_{ij} - t|^{p_M}, \quad (j = 1, \ldots, p)$$

then we obtain

$$\Phi_m = \sum_{j=1}^{p} \Phi_{mj}(Q_{mj}). \tag{4.21}$$

It follows from (4.21) that to find the minimum of the function Φ_m that depends on the variables $Q_{m1}, \ldots, Q_{mp}$, one needs to find the minimum of each of the functions Φ_{mj} $(j = 1, \ldots, p)$ depending only on one variable. Now we present the algorithm of finding the minimum of the functions $\Phi_{mj}(t)$.

Since a function $\Phi_{mj}(t)$ may be non-differentiable, however it is definitely a convex function (by the definition of Minkowski norm), to find the minimum of the function, let us employ the technique based on finding the subgradient of a convex function. Calculating the subgradient of $\Phi_{mj}(t)$, we obtain

$$\partial \Phi_{mj}(t) = \sum_{i \in C_m} p_M \, | X_{ij} - t |^{p_M - 1} \, \partial(| X_{ij} - t |)$$

where

$$\partial(| X_{ij} - t |) = \begin{cases} -1 & \text{if} \quad X_{ij} > t \\ [-1, 1] & \text{if} \quad X_{ij} = t \\ 1 & \text{if} \quad X_{ij} < t \end{cases}.$$

Therefore, for any t, $\partial\Phi_{mj}(t)$ is either a number or an interval $[u(t), v(t)]$. To find the minimum of a function $\Phi_{mj}(t)$, we calculate first $a_0 = \min_{i \in C_m} X_{ij}$ and $b_0 = \max_{i \in C_m} X_{ij}$. It is evident that the point of the minimum $\tilde{Q}_{mj} \in [a_0, b_0]$.

Next we find the point c_0 that is the middle point of the interval $[a_0, b_0]$ and calculate $\partial\Phi_{mj}(c_0) = [u(c_0), v(c_0)]$.

There are three possible cases:

(i) if $u(c_0) > 0, v(c_0) > 0$ then the point of the minimum is on the interval $[a_0, c_0]$;

(ii) if $u(c_0) < 0, v(c_0) < 0$ then the point of the minimum is on the interval $[c_0, b_0]$;

(iii) if values of $u(c_0)$ and $v(c_0)$ have different signs or at least one of the values is equal to zero then the point c_0 is the point of the minimum because $0 \in [u(c_0), v(c_0)]$ (Polyak, 1987).

If the point of minimum has not been found yet then for further consideration, we chose that of intervals $[a_0, c_0]$ and $[c_0, b_0]$ to what the point of minimum belongs to. Since the interval under consideration reduces twice at each iteration, the process converges very fast to the point of the minimum of the function $\Phi_{mj}(t)$.

If the data set is mixed then recalculating the numerical parts of the prototypes is fulfilled by the above described algorithm and recalculating the categorical part follows the recalculating of the common k-prototypes algorithm.

4.6 Applications of the algorithms based on Minkowski metrics to data sets

The above normalisation procedure was applied to attributes of two data sets from the UC Irvine repository (Asuncion and Newman 2007). All records in this data set have the class labels and, hence, "true clustering" can be checked.

First the clustering procedure has been applied to the data set without normalisation of the data. Then the clustering procedure with normalisation of all attributes has been applied to the data set. Both procedures with and without normalisation have been applied 100 times to the data sets for various values of the Minkowski power p_M. In the case $p_M = 2$, the k-prototype algorithm was employed.

4.6.1. Adult data set

The Adult data set, also known as Census Income dataset, has 48842 records and 30162 records without missing values (N = 30162) with 14 attributes and one class attribute. Each record has eight categorical attributes plus a class attribute, while the rest of the attributes are numerical.

Table 4.1 presents the results of application of the algorithm described in Section 4.8 to the Adult data set without normalisation of attributes for various values of the Minkowski power p_M. The accuracy function has been calculated involving the ideas of the assignment problem as it has been described in Section 3.5.2. Tables present the values of the clustering accuracy corresponding to the best value of the objective function, because this is the condition to achieve clustering.

Table 4.1: Clustering of the Adult data set without normalisation of attributes for various values of the Minkowski power p_M

Minkowski power p_M	Accuracy corresponding to the best value of objective function
1.0	0.5253
1.5	0.5960
2.0	0.6131
2.5	0.6364
3.0	0.6587
3.5	0.6876
4.0	0.7192
4.5	0.7381
5.0	0.7439

Clustering of the Adult data set without normalisation of attributes shows that the values of the clustering accuracy corresponding to the best value of the objective function (this value should be used for unsupervised clustering based on objective function) vary considerably with the variation of the values of the Minkowski power p_M. The best value has been obtained for $p_M = 5$ and it is $Acc = 0.7439$.

Table 4.2 presents the results of application of the algorithm described in Section 4.8 to the Adult data set with normalisation of attributes for various values of the Minkowski power p_M. The presented values are the same as in Table 4.1.

Table 4.2: Clustering of the Adult data set with normalisation of attributes for various values of the Minkowski power p_M

Minkowski power p_M	Accuracy corresponding to the best value of objective function
1.0	0.5769
1.5	0.7560
2.0	0.7536
2.5	0.6198
3.0	0.7560
3.5	0.7560
4.0	0.7560
4.5	0.7560
5.0	0.7560

Clustering of the Adult data set with normalisation of attributes shows that the values of the clustering accuracy corresponding to the best value of the objective function are much less sensible to the particular value of the Minkowski power p_M. The best value has been obtained for $p_M = 1.5; 3 \div 5$ and it is $Acc = 0.7560$. This accuracy is better than the accuracy obtained for the clustering without normalisation of attributes.

One can see that it is more advantageous to apply the general Minkowski metrics to the Adult data set (the clustering accuracy without normalisation is $Acc = 0.7439$ for $p_M = 5$ and it is $Acc = 0.7560$ for $p_M = 1.5; 3 \div 5$ in the case with normalisation) than a particular case $p_M = 2$ (the clustering accuracy without normalisation is $Acc = 0.6131$ and it is $Acc = 0.7536$ with normalisation).

4.6.2 Shuttle data set

The full Shuttle data set, also known as Statlog (Shuttle) Data Set, has N = 14500 records with 9 numeric attributes and one class attribute.

Table 4.3 presents the results of application of the algorithm described in Section 4.5 to the Shuttle data set without normalisation of attributes for various values of the Minkowski power p_M. As above, the accuracy function has been calculated involving the ideas of the assignment problem (see Section 3.5.2). The presented values are the same as in Tables 4.1 and 4.2.

Table 4.3: Clustering of the Shuttle data set without normalisation of attributes for various values of the Minkowski power p_M

Minkowski power p_M	Accuracy corresponding to the best value of objective function
1.0	0.4454
1.5	0.4541
2.0	0.6971
2.5	0.8294
3.0	0.7916
3.5	0.7915
4.0	0.7915

Clustering of the Shuttle data set without normalisation of attributes shows that the values of the clustering accuracy corresponding to the best value of the objective function vary considerably with the variation of the values of the Minkowski power p_M. The best value has been obtained for $p_M = 2.5$ and it is $Acc = 0.8294$.

Table 4.4 presents the results of application of clustering procedure to the Shuttle data set with normalisation of attributes for various values of the Minkowski power p_M. The presented values are the same as in above Tables.

Table 4.4: Clustering of the Shuttle data set with normalisation of attributes for various values of the Minkowski power p_M

Minkowski power p_M	Accuracy corresponding to the best value of objective function
1.0	0.4621
1.5	0.4715
2.0	0.4548
2.5	0.6849
3.0	0.8581
3.5	0.8463
4.0	0.7912

Clustering of the Shuttle data set with normalisation of attributes shows that the values of the clustering accuracy corresponding to the best value of the objective function also vary considerably with variation of the values of the Minkowski power p_M. The best value has been obtained for $p_M = 3$ and it is $Acc = 0.8581$. Again this accuracy is better than the accuracy obtained for the clustering without normalisation of attributes. One can see that it is more advantageous to apply the general Minkowski metrics to the Shuttle data set than a particular case $p_M = 2$ (the Euclidean metrics).

4.7 Summary

Our statistical approach introduced in Chapter 3 has been developed further and applied to the case of the Minkowski metrics beign used as a measure for continuous numerical features, while the matching dissimilarity measure is applied to categorical attributes.

To obtain a new normalised metric, one should calculate the mean contribution of each attribute to the metric and to divide the attribute in all records by this mean. Estimators are used to calculate the mean contributions. Rigorous mathematical proofs of unbiasedness and consistency of estimators used are presented (see Appendix B). Although this last property is very important in Statistics, in the papers revised, nobody spoke about the consistency of their estimators to the best of the author's knowledge. Various other mathematical problems related to the normalisation of mixed metrics are resolved.

The clustering algorithm applied in the case of the general Minkowski metrics is discussed in detail. The algorithm is based on ideas that were suggested before by Miyamoto and Augusta (1996) and Hathaway, Bezdek and Hu. (2000) as a generalisation of the fuzzy clustering strategies using L_p norm distances. The novelty of our approach is that we employ the algorithm for hard clustering using Minkowski norm distances. This algorithm has been used instead of the k-prototypes algorithm for the cases where $p_M \neq 2$.

The described algorithm and the introduced normalised metrics are applied to the Adult and Shuttle data sets. These examples have demonstrated the advantages of the introduced normalised metrics. It is also shown that it is more advantageous to apply for these data sets the general Minkowski metrics and the corresponding algorithm instead of a particular case $p_M = 2$ (the Euclidean metrics) and the k-prototypes algorithm.

Chapter 5

Improving the k-Prototypes algorithm by Random Search

In this Chapter a new algorithm to cluster data sets with mixed numerical and categorical values is presented. The algorithm is called RANKPRO: the Random Search with k -Prototypes Algorithm. It combines the advantages of a recently introduced population-based optimisation algorithm called the Bees Algorithm (BA), and the k -prototypes algorithm. The BA works with elite and good solutions, and continues to look for other possible extremal solutions keeping the number of testing points constant. However, the improvement of promising solutions by the BA algorithm may be time consuming because this process is based on the random neighbourhood search. On the other hand, an application of the k -prototypes algorithm to a promising solution may be very effective because it improves the solution at each iteration. The RANKPRO algorithm balances two objectives: it explores the search space effectively due to random selection of new solutions, and improves promising solutions fast due to employment of the k -prototypes algorithm. The efficiency of the new algorithm is demonstrated by clustering several numerical, categorical and mixed data sets. It is shown that in the majority of the considered data sets when the average number of iterations that the k -prototypes algorithm needs to

converge is over 10, the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

5.1. Background

As we have seen before, in clustering analysis of numerical data sets it is very common to calculate the similarity or dissimilarity between two feature vectors $\mathbf{x}_1 = (x_{11},\ldots,x_{1p})$ and $\mathbf{x}_2 = (x_{21},\ldots,x_{2p})$ using the Euclidean metric ρ_E (or L_2 metric)

$$\rho_E(\mathbf{x}_1,\mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2 = \left(\sum_{j=1}^{p}(x_{1j} - x_{2j})^2\right)^{1/2} \tag{5.1}$$

For example, the most popular algorithm for clustering numerical data sets is the k-means algorithm that uses the Euclidean distance.

The generalisation of the k-means algorithm by Huang (1997) that is called the k-prototypes algorithm is also based on the Euclidean distance. The k-prototypes algorithm was introduced to cluster large data sets with mixed numerical and categorical values. It should be noted that both the k-means and k-prototypes algorithms have a disadvantage, namely the process converges often not to a global minimum but to a local minimum. Hence, to avoid this premature convergence one has to modify these algorithms. Recently, Pham et al. (2006b) have presented an approach to optimisation problems that is called the Bees Algorithm (BA).

The BA combines neighbourhood search with random search. The randomness of the search provides flexibility in the search and hence, the BA gives often results that are quite close to global minimum. In this Chapter a new tool for clustering mixed data sets is introduced that combines the advantages of both the k-prototypes and the BA algorithms. The Chapter is organised as follows:

5.2 presents a formal description of both the k-prototypes and BA algorithms.

5.3 presents a description of the random search with k-prototypes algorithm(RANKPRO).

5.4 RANKPRO is applied to several data sets. The effectiveness of the RANKPRO and the k-prototypes clustering algorithms are compared and the advantages of the former algorithm are shown.

5.2. Preliminaries

5.2.1. The k-means and k-prototypes algorithms

The k-means algorithm (MacQueen 1967) was introduced to cluster numerical data sets. The specific properties of the algorithm have been discussed in previous Chapters in detail. Here we present briefly the formal formulation of the algorithm along with the presentation of the known BA algorithm (Pham et al. 2006b).

The k-means algorithm minimises the cost function (objective function) J for "hard" k-partitions of data set into k clusters (Bezdek 1980, Huang 1997b)

$$J = \sum_{m=1}^{k} \sum_{i=1}^{N} u_{im} \rho_E^2(\mathbf{X}_i, \mathbf{Q}_m),$$

$$u_{im} \in \{0,1\}, \ 1 \le i \le N, \ 1 \le m \le k,$$

$$\sum_{m=1}^{k} u_{im} = 1, \quad \forall i, \quad \text{and} \quad \sum_{i=1}^{N} u_{im} > 0 \quad \forall m. \tag{5.2}$$

Here u_{im} is an element of the partition matrix. The condition $u_{im} = 1$ means that the record $\mathbf{X}_i$ is assigned to cluster m with prototype (centre) $\mathbf{Q}_m$. Since ρ_E defined by (5.1) is employed in this Thesis for clustering of numerical data, J is the within-group sum of squared errors objective function.

The implementation of k-means may have various forms, in particular its pseudo code can be written as:

Step 1. Select randomly k initial prototypes $\mathbf{Q}_1, ..., \mathbf{Q}_k$, one for each cluster.

Step 2. For each record $\mathbf{X}_i$ calculate the distances from the record to the prototypes of clusters; find the nearest prototype $\mathbf{Q}_m$ to the record according to the metric ρ_E defined by (5.1), and allocate the record $\mathbf{X}_i$ to the cluster C_m with this prototype.

Step 3. For each cluster C_m find a new prototype $\mathbf{Q}_m'$ so that the sum of square distances $\sum_{X_i \in C_m} \rho_E^2(\mathbf{X}_i, \mathbf{Q}_m')$ is minimum.

Step 4. If prototypes $\mathbf{Q}_1,...,\mathbf{Q}_k$ and $\mathbf{Q}_1',...,\mathbf{Q}_k'$ are not the same then take the latest as new prototypes and go to step 2, otherwise stop the procedure.

It is known (see e.g. Stevens, 1946) that to deal statistically with categorical data, one needs to deal with modes of the data instead of means or medians that are used to deal with numerical variables. In statistics the mode is that value which occurs most often or, in other words, has the greatest probability of occurring (see, e.g. (Spiegel 1975)). As it has been mentioned in Chapter 2, Huang (1997, 1998) introduced two extensions of the k-means algorithm, namely the algorithms called k-modes and k-prototypes. In the k-prototypes algorithm he considered a metric ρ_H, where ρ_H^2 is the sum of the square of the numerical metric (5.1) and a weighted categorical metric ρ_{cat}

$$\rho_H^2 = \rho_E^2 + \gamma \rho_{cat} \tag{5.3}$$

The categorical metric ρ_{cat} is defined as the number of mismatches of categories between two objects and the weight γ is introduced for the categorical metric to balance the two parts of the sum and to avoid favouring either type of attribute.

The pseudo codes of the k-modes and k-prototypes algorithms are very similar to the pseudo code of the k-means algorithm. The difference between the algorithms is mainly that different dissimilarity measures have to be used. The k-means algorithm has a great advantage that it converges very fast to a local minimum and at each

iteration it improves the solution. Consequently the k-prototypes algorithm has the same advantage. However, application of both k -means and k -prototypes algorithms to data sets have also a disadvantage, namely the process demonstrates normally a premature convergence, i.e. it converges not to a global minimum but to a local minimum. Hence, one needs to run the procedure many times to reach the global minimum. To increase the effectiveness of the procedure, one has to modify these algorithms.

5.2.2. The Bees Algorithm

As we have seen in Chapter 2, SI is a type of optimisation technique that mimics the collective behaviour of animals. There are several methods that can be considered as SI; the Bees Algorithm is one of them. It is a new technique that was introduced to mimics nature's evolutionary principles that drive the search of bees towards an optimal solution. In application to problems of optimisation, a bee means a point of the domain (the search space) of the objective function, while the fitness of the bee means the value of the objective function at this point. It was shown (Pham et al. 2006b) that using the BA for some optimisation problems is more effective than using the GA based techniques (Goldberg, 1989).

The BA starts by initialising a set of the following parameters: the number of scout bees (n) that define the total number of sites; the number of best sites (m) out of the total number of n sites; the number of elite sites (e); the size of each patch (a patch is

a region in the search space that includes the visited site and its neighbourhood) (d_{ngh}) around any of the best sites; the number of recruited bees (r_e) within the neighbourhood of the elite sites; the number of recruited bees (r_g) around other selected ($g = m - e$) sites, and stopping criteria. According to the pseudo code for the BA (Pham et al. 2006b), n bees are placed on the search space randomly, similar to scout bees. Every bee on the problem space evaluates the fitness of its field in step 2. Subsequently, in step 4, elite bees that have better fitness are selected and saved for the next population. In step 5, good sites for neighbourhood search are selected. In step 7, the bees search around these points within the neighbourhood boundaries and their individual fitness is evaluated. More bees will be recruited around elite points and fewer bees will be recruited around the remaining selected points.

The pseudo code for the Bee Algorithm can be written as (Pham *et al.* 2005)

Step 1. Initialise population with random solutions (n sites discovered by n scout bees).

Step 2. Evaluate fitness of the population (for n sites).

Step 3. While (stopping criterion not met)

Step 4. Select e elite sites.

Step 5. Select g good sites for neighbourhood search ($e + g = m$).

Step 6. Determine the patch size.

Step 7. Recruit r_e bees around each of selected elite sites and r_g bees around each of selected good sites.

Step 8. Evaluate fitness of solutions for all of these recruited bees and select the best bee for each neighbourhood.

Step 9. Assign remaining $(n-m)$ bees to search randomly and evaluate the fitness of each of the discovered sites.

Step 10. Forming new population.

Step 11. End While.

Some features of the BA algorithm are very similar to some features of Hill climbing (HC), Local beam search (LBS) and Stochastic beam search (SBS) strategies that are used to solve optimisation problems. The description of these techniques can be found elsewhere (see, e.g. Russell and Norvig 2003). Indeed, similarly to these methods, the BA starts with a random selection of solutions and then improves the solutions iteratively. However, contrary to the HC and LBS algorithms, it has probability to converge to the global extremum in a multiextremum problem due to random exploring of the search space at each iteration. The BA and SBS algorithms use different procedures of selection of the fittest solutions.

The GA search balances two objectives: utilising the best solutions and exploring the search space (see e.g. Michalewicz 1996). The BA tries also to balance these objectives. However, to explore the search space the BA uses random search instead of crossover and mutation operations used by GA. The BA works with the most promising and elite solutions. In application to genetic algorithms, the term elitism was first introduced by De Jong (1975) (see also Mitchell, 1996) in order to force a GA to retain some number of the best individuals at each generation. The introduction of elite individuals enables the algorithm under consideration to preserve the best solutions. Otherwise they could be lost or destroyed by crossover or mutation. This term has been employed by the BA that works with elite and good

points, and continues to look for other possible extreme points keeping the number of testing points constant.

The BA has also some common features with Controlled Random Search (CRS) algorithms introduced by Price (1978) (see also (Kaelo and Ali, 2006) for a review of recent modifications on CRS). Indeed, both CRS and BA algorithms employ initial points that are uniformly distributed over the search space and unlike gradient based methods, they calculate only the value of the function itself and they do not use any property of the function. However, CRS and BA algorithms have also a considerable difference. In the CRS the region of testing points is gradually contracted by replacing the current worst point with a better point (the trial point) that is chosen by a kind of interpolation, while the BA explores always the whole search space.

It was suggested to use the BA not only for optimisation problems but also for clustering (Pham et al., 2007) where a bee represents a potential clustering solution as a set of k cluster centres. However, again the BA suggests to use a random search in the neighbourhoods of selected solutions (partitions) and hence, the local improvement of promising solutions by the BA algorithm may be time consuming.

On the other hand, there is a faster way to improve the solutions, namely the employment of the k-prototypes algorithm that converges very fast to a local optimal solution. In this the algorithm is similar to gradient-like methods of search of local extrema (see e.g. Pham and Jin, 1995; Wen et al., 2003). Therefore, it is proposed to use the k-prototypes algorithm to improve the promising solutions. Thus, the new RANKPRO algorithm that includes the preservation of elite solutions and random

exploring the search space along with employment of k-prototypes algorithm for improvement of solutions is proposed.

5.3 Description of RANKPRO

To deal with mixed data sets one needs to use proper metrics. Hence, the normalisation of metrics for mixed data sets is discussed first and then the RANKPRO algorithm is described in detail.

5.3.1. Normalisation of metrics for mixed data sets

It has been mentioned above that in clustering analysis of numerical data sets the Euclidean metric (5.1) is commonly used. For data sets with categorical attributes, it is possible to introduce different metrics (see, e.g. Gibert and Cortes, 1997; Huang, 1998; Ralambondrainy, 1995). One of the most used variants of metrics is the matching dissimilarity measure between two categorical feature vectors $\mathbf{y}_1 = (y_{11},...,y_{1l})$ and $\mathbf{y}_2 = (y_{21},...,y_{2l})$ (see, e.g. Huang, 1998). It is defined as

$$\rho_{cat}(\mathbf{y}_1,\mathbf{y}_2) = \omega(y_{11},y_{21}) + ... + \omega(y_{1l},y_{2l}) \tag{5.4}$$

were

$$\omega(y_{1j},y_{2j}) = \begin{cases} 0 & \text{for } y_{1j} = y_{2j} \\ 1 & \text{for } y_{1j} \neq y_{2j} \end{cases}$$

Evidently, the square of the metric (5.4) is (see (3.23))

$$\rho^2_{cat}(\mathbf{y}_1,\mathbf{y}_2) = \omega^2(y_{11},y_{21}) + ... + \omega^2(y_{1l},y_{2l})$$

Combining ρ_E and ρ_{cat} for mixed data, one obtains that the square distance between two mixed feature vectors $(\mathbf{x}_1,\mathbf{y}_1)$ and $(\mathbf{x}_2,\mathbf{y}_2)$ is

$\rho^2((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2)) = \rho_E^2(\mathbf{x}_1,\mathbf{x}_2) + \rho_{cat}^2(\mathbf{y}_1,\mathbf{y}_2)$ where $\rho_E^2(\mathbf{x}_1,\mathbf{x}_2)$ is defined by (5.1) and $\rho_{cat}^2(\mathbf{y}_1,\mathbf{y}_2)$ is defined by (3.23).

As it has been mentioned above, the use of metric (5.3) may encounter some obstacles in practical realisation because it is not very clear how to find a proper weight γ for the metric to balance the two parts of the sum and to avoid favouring either type of attribute. A direct application of geometric measures (e.g. city block or Euclidean distances) for attributes with large ranges will implicitly assign bigger contributions to the metrics than those for attributes with small ranges. In addition, the attributes should be dimensionless. Indeed, the numerical values of the ranges of dimensional attributes depend on the units of measurements and therefore, the choice of the units of measurements may greatly affect the results of clustering. Hence, if all attributes are equally important to measure similarity between feature vectors then one should use a normalisation procedure. Here a normalisation procedure as was described in detail in Chapter 3 will be employed. To obtain a new normalised metric for the Euclidean metric, one should calculate the mean contribution of each j-th attribute to the metric $E\left|X_{1j} - X_{2j}\right|^2$ and to divide the attribute in all records by this mean. Hence, the normalised Euclidean mixed metric is (see (3.25))

$$\rho^*((\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2)) = \left(\sum_{j=1}^{p}\alpha_j\left|x_{1j} - x_{2j}\right|^2 + \sum_{j=1}^{l}\beta_j\omega^2(y_{1j},y_{2j})\right)^{1/2},$$

where $\alpha_j = 1/E\left|X_{1j} - X_{2j}\right|^2$ and $\beta_j = 1/E\omega^2\left(Y_{1j},Y_{2j}\right)$.

5.3.2 Pseudo code of RANKPRO

If there are a data set and a set of k prototypes $\mathbf{S}=\{\mathbf{Q}_1,...,\mathbf{Q}_k\}$ then a record $\mathbf{A}_i$ may be allocated to cluster C_m whose prototype $\mathbf{Q}_m$ is the nearest to the record according to the normalised mixed metric (3.25). Hence, a set of prototypes $\mathbf{S}$ gives a partition of the data set to k clusters $\{C_1,...,C_k\}$. In this Thesis a set of prototypes $\mathbf{S}=\{\mathbf{Q}_1,...,\mathbf{Q}_k\}$ will be called an approximate solution to the clustering problem if it gives a partition to k non-empty clusters.

The clustering algorithm has to minimise the objective function

$$J(S)=\sum_{m=1}^{k}\sum_{i=1}^{N}u_{im}\left[\rho^*(\mathbf{A}_i,\mathbf{Q}_m)\right]^2,$$

$$u_{im}\in\{0,1\},\ 1\le i\le N,\ 1\le m\le k,$$

$$\sum_{m=1}^{k}u_{im}=1,\forall i,\ \text{and}\ \sum_{i=1}^{N}u_{im}>0\ \ \forall m. \tag{5.5}$$

The condition $u_{im}=1$ for an element of the partition matrix means as above in (5.2) that the record $\mathbf{A}_i$ is assigned to cluster C_m with prototype $\mathbf{Q}_m$. Evidently, (5.5) can be written as

$$J(S)=\sum_{m=1}^{k}\sum_{\mathbf{A}_i\in C_m}\left[\rho^*(\mathbf{A}_i,\mathbf{Q}_m)\right]^2. \tag{5.6}$$

To start the RANKPRO one has to give a set of parameters, namely the number (n) of the approximate solutions to the clustering problem that are considered at each

step, the number $(e,\ 1 \le e < n)$ of elite solutions that are kept to the next step of the algorithm, the number $(r = n - e)$ of solutions that are used for random search, and the number (n_{iter}) of iterations of the k-prototypes algorithm that is applied to each solution to improve the solution. As stopping criterion, one can take either the approach of the process time to the given maximum time t_{max} or the approach of the number of process iterations to the given maximum number of iterations.

The pseudo code for RANKPRO can be described as following:

Step 1. Initialization. Select randomly n solutions $\mathbf{S}_p$, $1 \le p \le n$.

Step 2. While (the stopping criterion is not met yet) consider the selected solutions.

Step 3. Apply n_{iter} of iterations of the k-prototypes algorithm to each solution to improve the solution. The application of the algorithm to the solution has to be stopped if it becomes stable; this means that it has reached the local minimum.

Step 4. For each $\mathbf{S}_p$, calculate the objective function $J(\mathbf{S}_p)$.

Step 5. Select e solutions with the best values of the objective function for further study at the next step, the rest $r = n - e$ solutions are removed and replaced by randomly selected ones.

Step 6. End While.

The solution of the clustering problem is the best solution $\mathbf{S}$ obtained at the last step.

It is important to note that in order to save time in the process of improvement of solutions, the k-prototypes algorithm is applied to the solutions not until its

convergence but only n_{iter} times. This number can be estimated by a prior study of a specific data set.

Like the BA, the RANKPRO algorithm uses a population of solutions for each iteration instead of a single solution. The BA suggests using a random search in the neighbourhoods of selected solutions and hence, the local improvement of promising solutions by the BA algorithm may be time consuming. The employment of the k-prototypes algorithm that converges very fast to a local optimal solution is a faster way to improve the solutions. In this the algorithm is similar to gradient-like methods of search of local extreme (like Pham and Jin 1995).

5.4 Applications to data sets

5.4.1. Comparing the effectiveness of the clustering algorithms

The comparison of the effectiveness of the clustering algorithms is not an easy task. Goldberg and Deb (1991) reviewed and compared several selection schemes used in genetic algorithms. They noted that many claims and counterclaims were presented regarding the superiority of a selection scheme over another one in genetic algorithms. However, most of these claims are based on limited (and uncontrolled) simulation experience; while surprisingly little analysis was performed to understand relative expected fitness ratios, convergence times, or the functional forms of selective convergence. A similar situation may be encountered in the area of

comparison of clustering algorithms. Hence, the following procedure for comparing the effectiveness of the clustering algorithms has been suggested.

The above methods (the RANKPRO and the k-prototypes algorithms) are applied to several data sets from the UC Irvine repository (Asuncion and Newman 2007) after normalisation of the data in accordance with the above described method. The effectiveness of the clustering algorithms is compared with one another for different initial parameters and data sets. The scheme described below compares the average minimum values of the objective function obtained during the runs of the algorithms that are applied to the same data set during the same time.

Let the RANKPRO algorithm with a specified set of initial parameters be applied to a specified data set during the given time t_{exec} and $J(t_{exec})$ be the minimum value of the objective function (5.6) obtained by the execution of the algorithm during this time. If the algorithm is run a given number n_r of simulations then one obtains a set of $J^{(m)}(t_{exec})$ values of the objective function $(m=1,...,n_r)$. The average value $J_{av}(t_{exec}) = \sum_{m=1}^{n_r} J^{(m)}(t_{exec})/n_r$ is a characteristic of the effectiveness of the algorithm during t_{exec}. The less is $J_{av}(t_{exec})$ for the algorithm the greater is the effectiveness of the algorithm. However, if one takes the value t_{exec} rather large then all algorithms may give the same value of the objective function, namely its global minimum value. Hence, it is pointless to compare the algorithms for large values of t_{exec}. Evidently, $J_{av}(t_{exec})$ depends also on the values of n_r. However, if n_r is large enough then the variations of the $J_{av}(t_{exec})$ values will be rather small.

To compare the effectiveness of the RANKPRO and the k-prototypes algorithms, the latter is also applied to the same data set. If the k-prototypes algorithm converges before the process time will reach the value t_{exec} then the algorithm is run again and again until the allowed process time is not expired. Each time after convergence of the k-prototypes process, the minimum value of the objective function is recorded. The average of the minimum values of the objective function obtained during these runs is taken as $J_{av}(t_{exec})$.

5.4.2. Adult data set

The Adult data set, also known as Census Income dataset, has 48842 records and 30162 records without missing values (N = 30162) with 14 attributes and one class attribute. Each record has eight categorical attributes plus a class attribute, while the rest of the attributes are numerical. This is a standard data set that was studied a number of times to test clustering algorithms (see, e.g. (Huang 1998)). For ADULT data set, the Figures 1-4 show the graphs of the average values of $J_{av}(\mathbf{S})$ versus t_{exec} in seconds. The number of simulations is n_r = 100 in all cases.

One can see in Figure 5.1 that if the parameters n, e and r are fixed $(n=8, e=1, \text{and } r=7)$ and the parameter n_{iter} is varied then the best performance is for $n_{iter}=5$. In all cases the RANKPRO algorithm gave smaller values for the

average of the objective function $J_{av}(t_{exec})$, i.e. the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

Figure 5.2 shows that it is more efficient to keep only one elite solution $e=1$ than two elite solutions $e=2$ irrespectively on the number of iterations $(n_{iter}=3$ or $n_{iter}=5)$ of the k-prototypes algorithm applied to improve the solutions.

Figure 5.3 confirms the above conclusions. If the parameters n, e and r are fixed $(n=8, e=2, \text{and } r=6)$ and the parameter n_{iter} is varied then the best performance is for $n_{iter}=5$. The RANKPRO algorithm is more efficient than the k-prototypes algorithm. However, the performance of the former algorithm is worse than its performance in the case $e=1$, i.e. it is more efficient to keep only one elite solution than two elite solutions.

One can see in Figure 5.4 that if the number of randomly chosen solutions n is varied, while other parameters e and n_{iter} are fixed $(n_{iter}=5 \text{ and } e=1)$, the best performance is for $n=8\,(r=7)$. In all cases the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

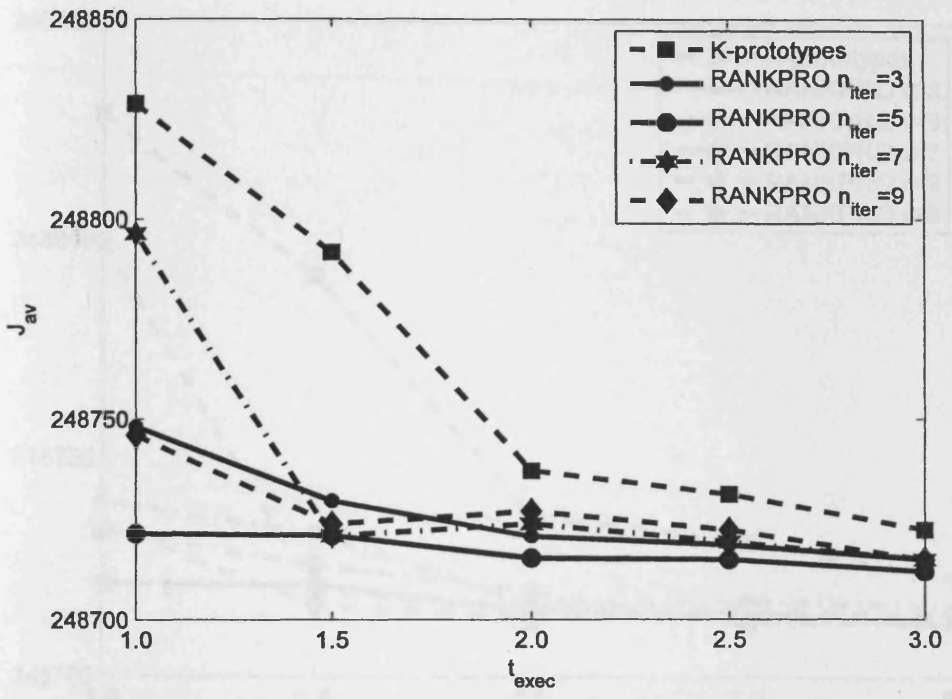


Figure 5.1: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for ADULT data set. Comparison of the k -prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 3, 5, 7 and 9, while other parameters are constant: $n = 8$, $e = 1$, and $r = 7$.

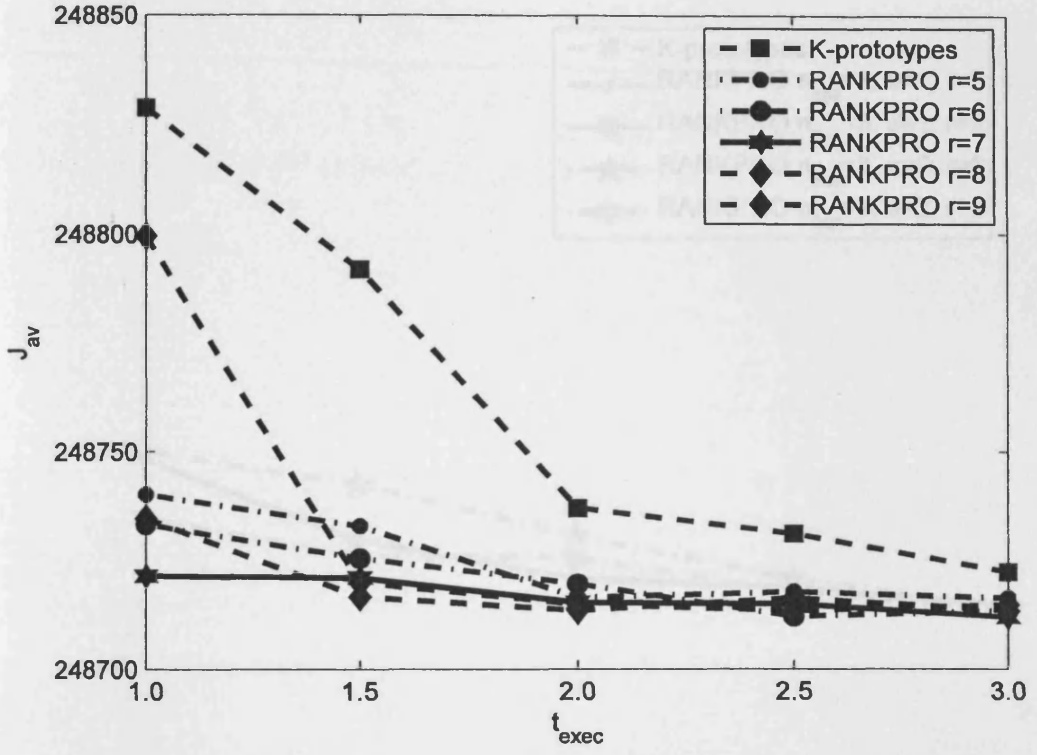


Figure 5.2: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for ADULT data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with the following parameters: (i) $n_{iter} = 3$, $e = 1$, $r = 7$ and $n = 8$, (ii) $n_{iter} = 5$, $e = 1$, $r = 7$ and $n = 8$, (iii) $n_{iter} = 3$, $e = 2$, $r = 6$ and $n = 8$, and (iv) $n_{iter} = 5$, $e = 2$, $r = 6$ and $n = 8$.

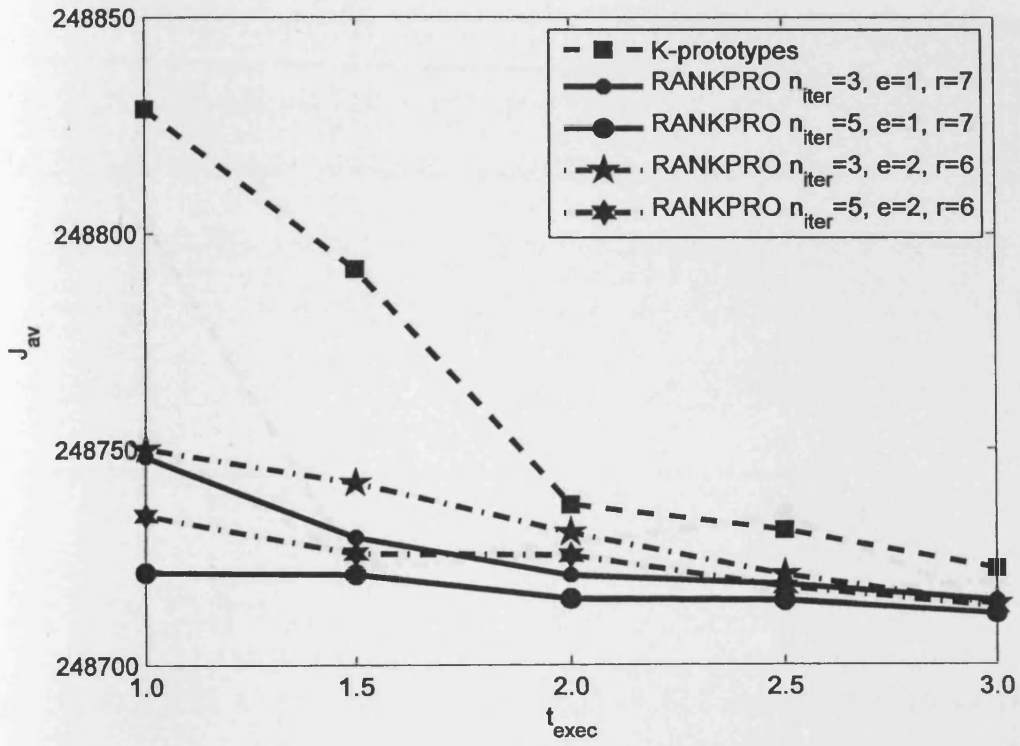


Figure 5.3: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for ADULT data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 3, 5, 7 and 9, while other parameters are constant: n = 8, e = 2, and r = 6.

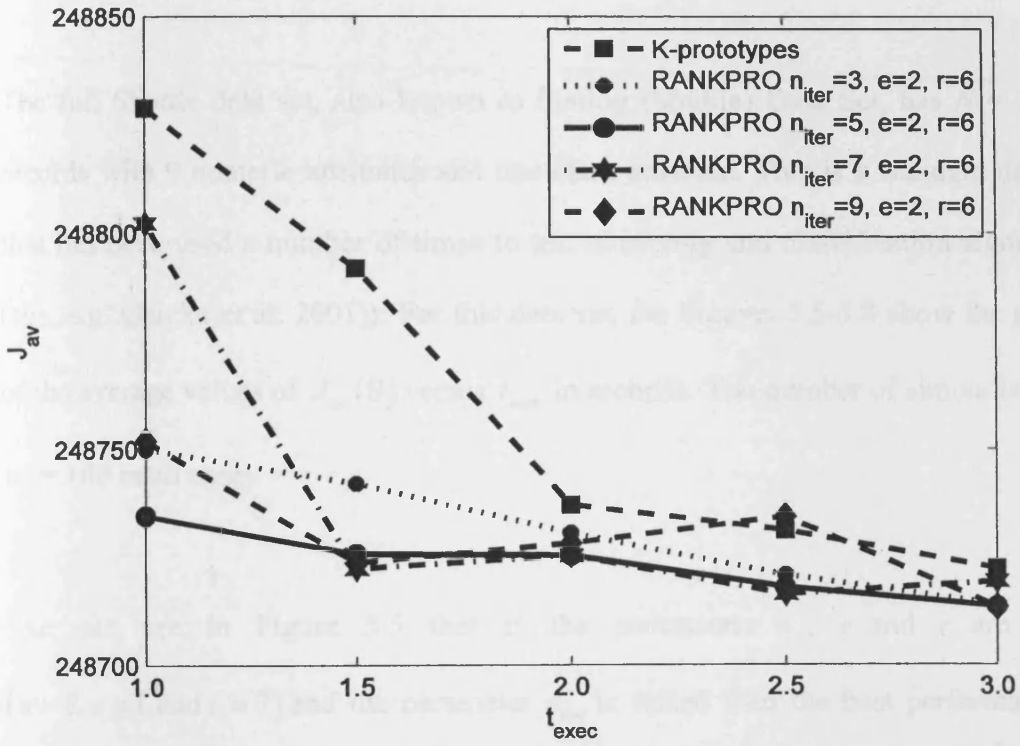


Figure 5.4: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for ADULT data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n = 6÷10 and correspondingly with different values of r = 5÷ 9, ($r = n - e$), and fixed parameters n_{iter} = 5 and e = 1.

5.4.3. Shuttle data set

The full Shuttle data set, also known as Statlog (Shuttle) Data Set, has N = 14500 records with 9 numeric attributes and one class attribute. This is a standard data set that has been used a number of times to test clustering and classification algorithms (see, e.g. Garcke et al. 2001)). For this data set, the Figures 5.5-5.8 show the graphs of the average values of $J_{av}(\mathbf{S})$ versus t_{exec} in seconds. The number of simulations is n_r = 100 in all cases.

One can see in Figure 5.5 that if the parameters n , e and r are fixed $(n=8, e=1, \text{and } r=7)$ and the parameter n_{iter} is varied then the best performance is for $n_{iter}=5$. In all cases the RANKPRO algorithm gave smaller values for the average of the objective function $J_{av}(t_{exec})$, i.e. the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

Figure 5.6 shows that for the Shuttle data set contrary to the ADULT data set, there is no evident advantage of keeping just one elite solution $(e=1)$ or two elite solutions $(e=2)$. In all cases the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

Figure 5.7 shows that if the parameters n, e and r are fixed $(n=8, e=2, \text{and } r=6)$ and the parameter n_{iter} is varied then the best performance is for $n_{iter}=3$ The RANKPRO algorithm is more efficient than the k-prototypes algorithm.

One can see in Figure 5.8 that if the number of randomly chosen solutions n is varied, while other parameters e and n_{iter} are fixed $(n_{iter}=5 \text{ and } e=1)$, there is no evident advantage for a specific number n. In all cases the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

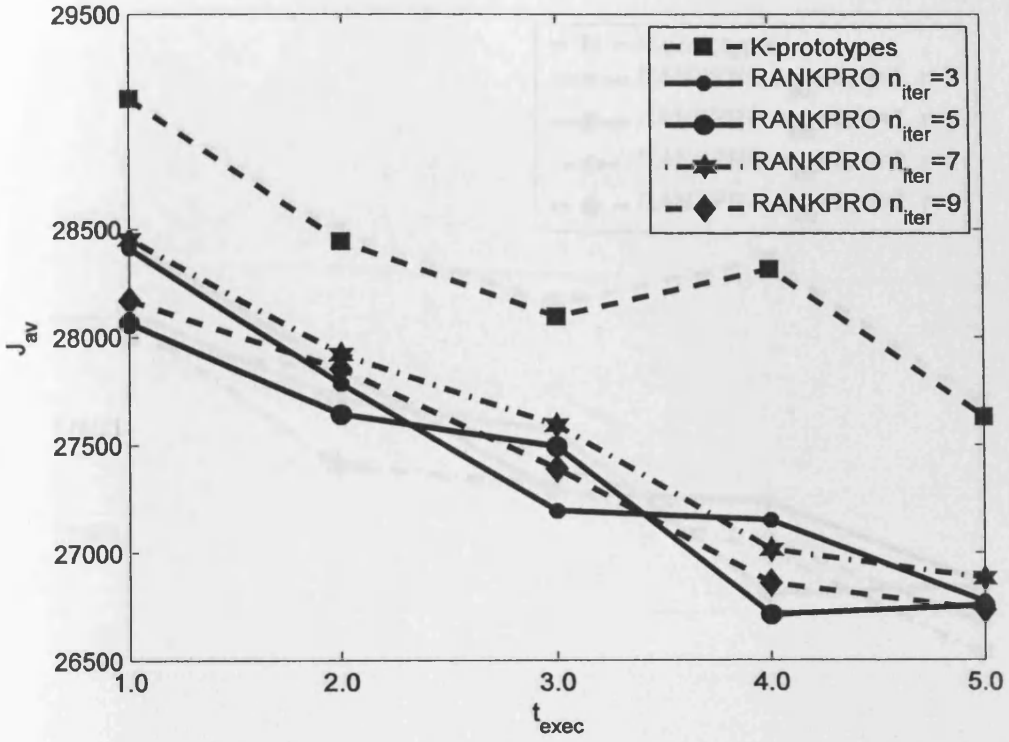


Figure 5.5: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Shuttle data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 3, 5, 7 and 9, while other parameters are constant: $n = 8$, $e = 1$, and $r = 7$.

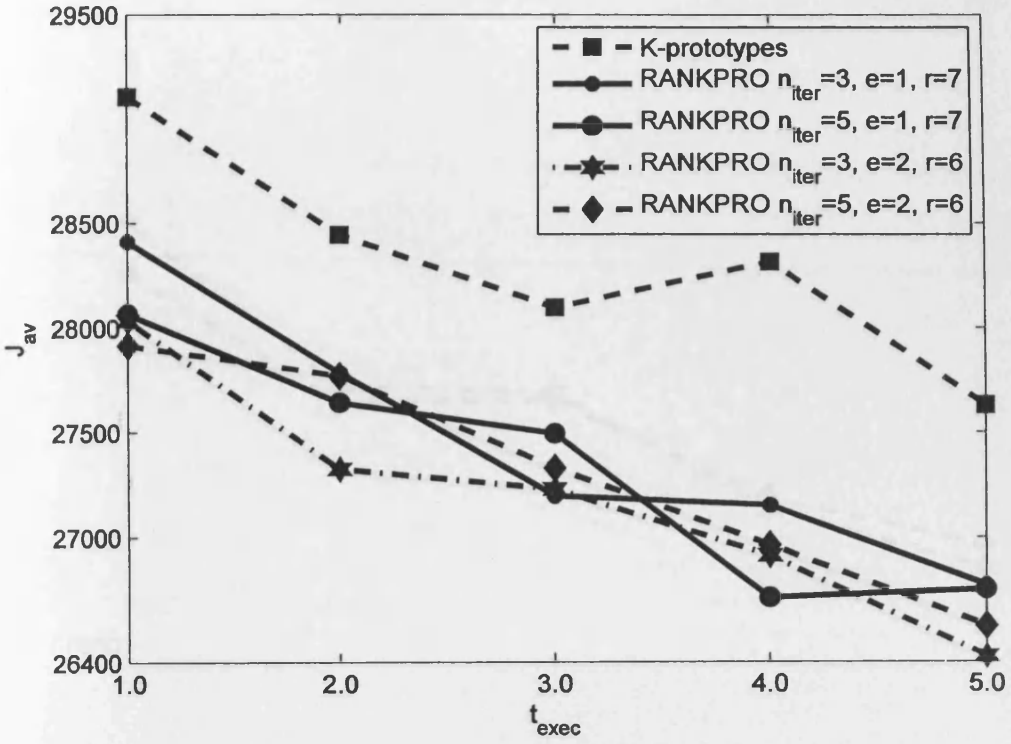


Figure 5.6: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Shuttle data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with the following parameters: (i) $n_{iter} = 3$, $e = 1$, $r = 7$ and $n = 8$, (ii) $n_{iter} = 5$, $e = 1$, $r = 7$ and $n = 8$, (iii) $n_{iter} = 3$, $e = 2$, $r = 6$ and $n = 8$, and (iv) $n_{iter} = 5$, $e = 2$, $r = 6$ and $n = 8$.

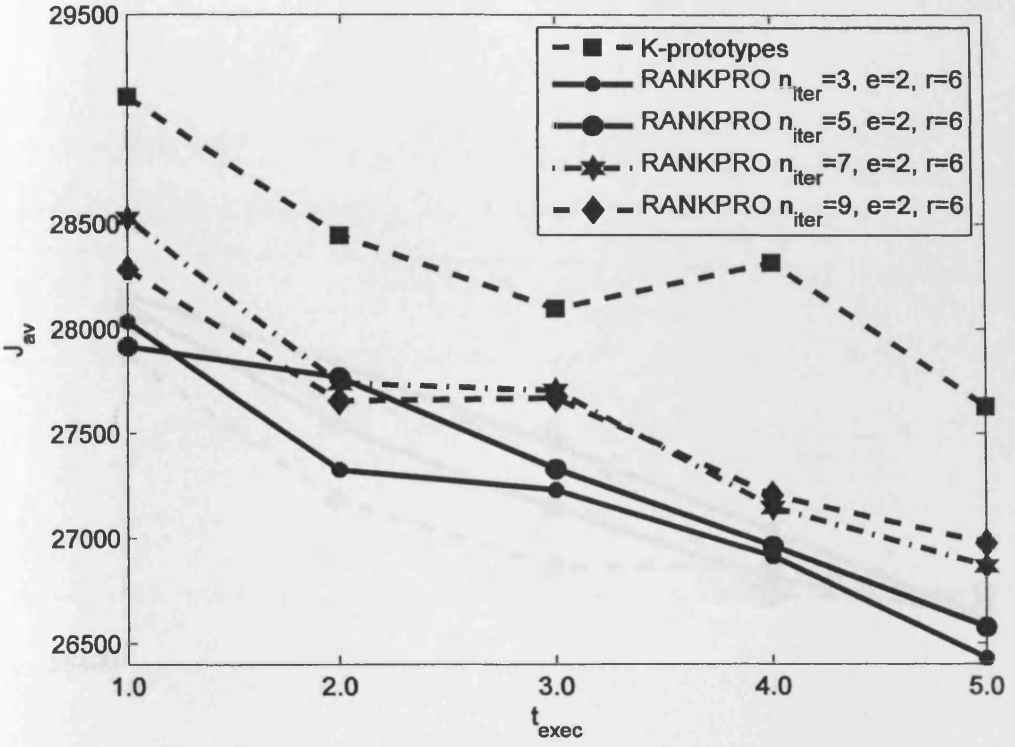


Figure 5.7: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Shuttle data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 3, 5, 7 and 9, while other parameters are constant: $n = 8$, $e = 2$, and $r = 6$.

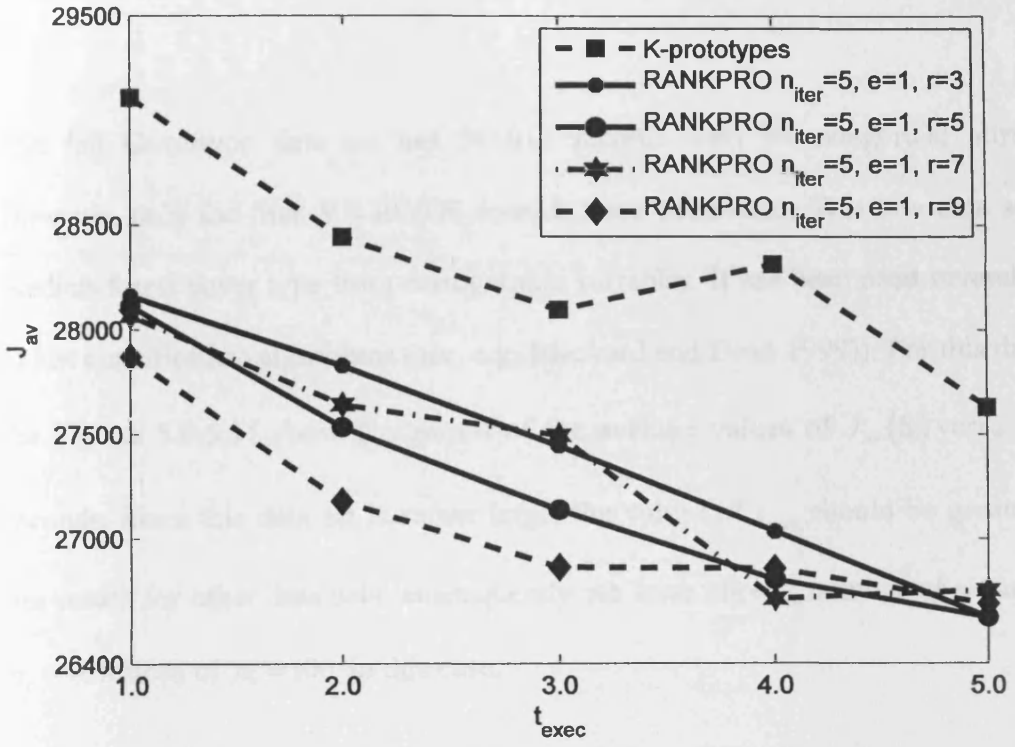


Figure 5.8: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Shuttle data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of $n = 4 \div 10$ and correspondingly with different values of $r = 3 \div 9$, $(r = n - e)$, and fixed parameters $n_{iter} = 5$ and $e = 1$.

5.4.4. Covertype data set

The full Covertype data set has 581012 records with 54 categorical attributes. However, only the first $N = 100000$ records have been used. This is a data set that predicts forest cover type from cartographic variables. It has been used several times to test classification algorithms (see, e.g. Blackard and Dean 1999)). For this data set, the Figures 5.9-5.11 show the graphs of the average values of $J_{av}(\mathbf{S})$ versus t_{exec} in seconds. Since this data set is rather large, the values of t_{exec} should be greater than the values for other data sets, consequently we have chosen number of simulations $n_r = 10$ instead of $n_r = 100$ in this case.

One can see in Figure 5.9 that if the parameters n, e and r are fixed $(n = 9, e = 1, \text{and } r = 8)$ and the parameter n_{iter} is varied then it is difficult to give any preference to a specific value of n_{iter}. However, the RANKPRO algorithm gave smaller values for the average of the objective function $J_{av}(t_{exec})$ in all cases.

Figure 5.10 shows that for the Covertype data set, there is an advantage of keeping just one elite solution $(e = 1)$ rather than two elite solutions $(e = 2)$. In all cases the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

Figure 5.11 shows that in the case of two elite solutions, there is an advantage of keeping $n_{iter} = 6$. Again the RANKPRO algorithm is more efficient than the k-prototypes algorithm.

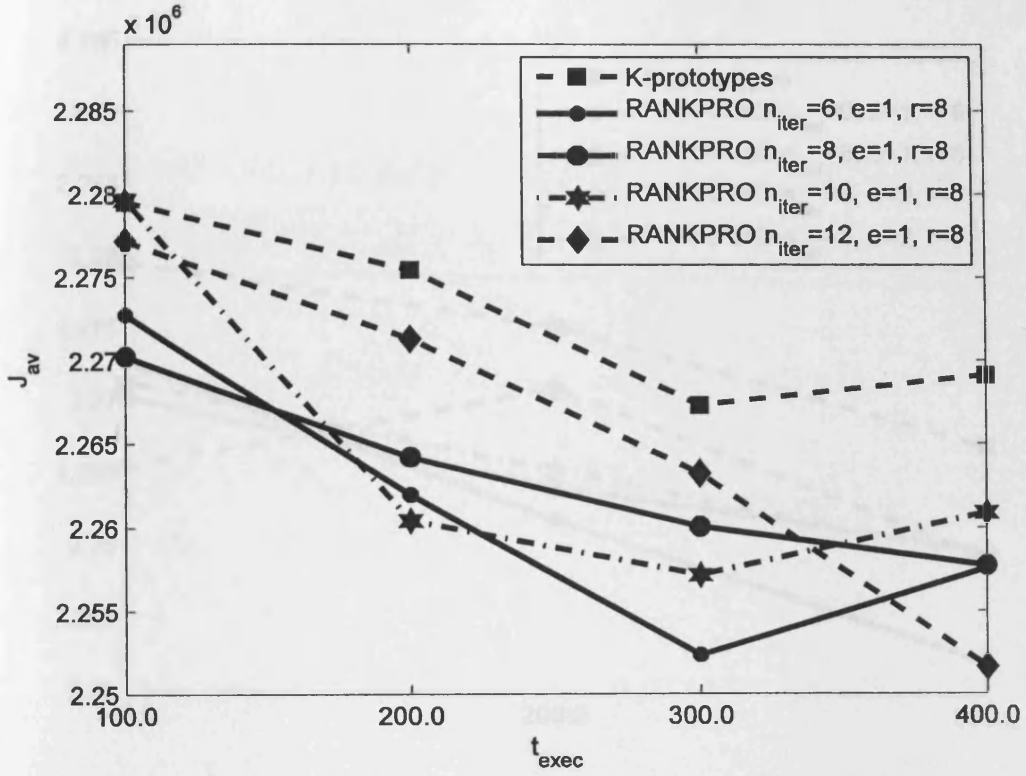


Figure 5.9: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Covertype data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 6, 8, 10 and 12, while other parameters are constant: $n = 9$, $e = 1$, and $r = 8$.

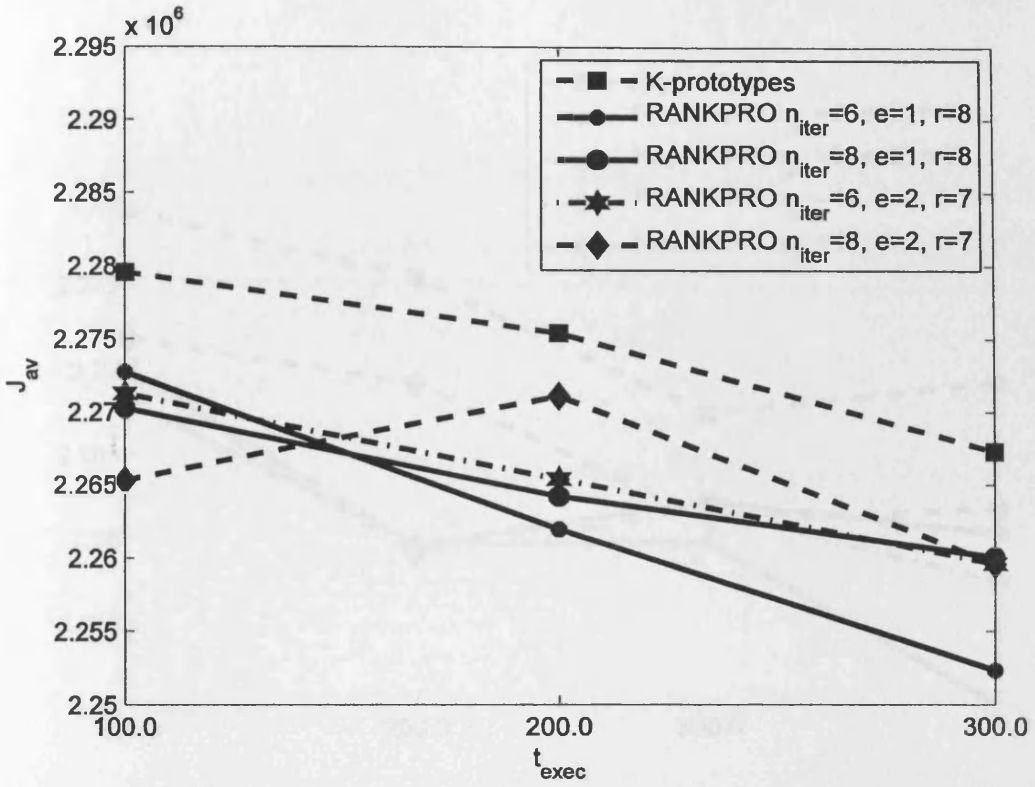


Figure 5.10: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Covertype data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with the following parameters: (i) $n_{iter} = 6$, $e = 1$, $r = 8$ and $n = 9$, (ii) $n_{iter} = 8$, $e = 1$, $r = 8$ and $n = 9$, (iii) $n_{iter} = 6$, $e = 2$, $r = 7$ and $n = 9$, and (iv) $n_{iter} = 8$, $e = 2$, $r = 7$ and $n = 9$.

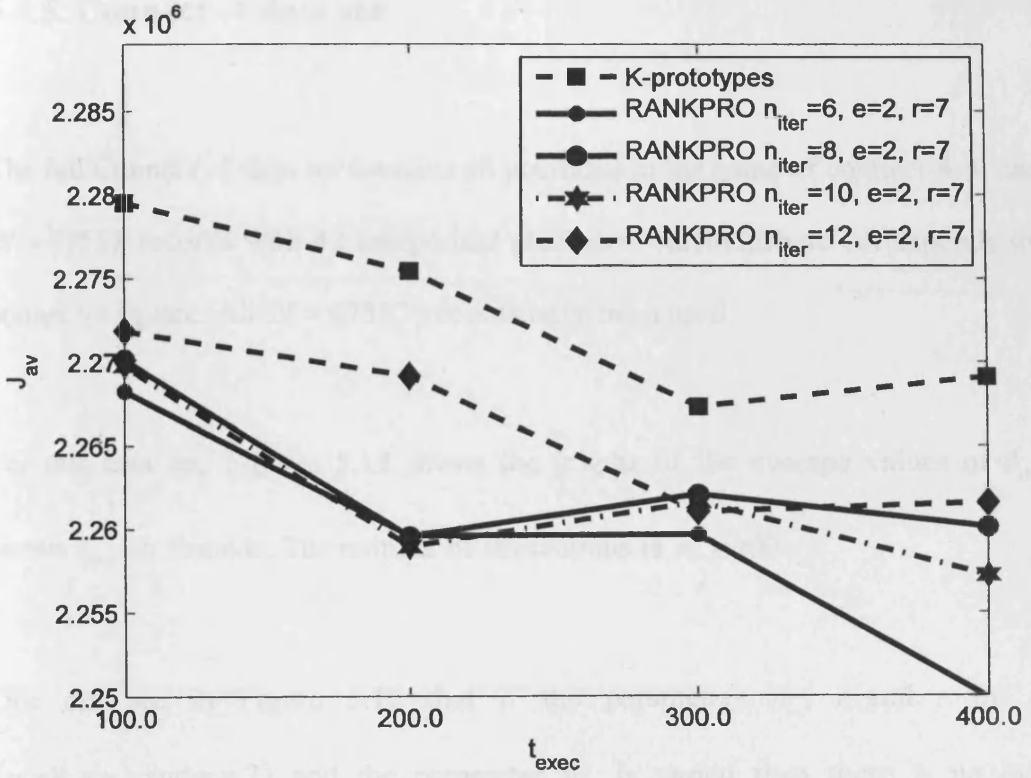


Figure 5.11: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Covertype data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 6, 8, 10 and 12, while other parameters are constant: n = 9, e = 2, and r = 7.

5.4.5. Connect -4 data set

The full Connect -4 data set contains all positions in the game of connect-4. It has $N = 67557$ records with 42 categorical attributes, each attribute corresponds to one connect-4 square. All $N = 67557$ records have been used.

For this data set, Figures 5.12 shows the graphs of the average values of $J_{av}(\mathbf{S})$ versus t_{exec} in seconds. The number of simulations is $n_r = 100$.

One can see in Figure 5.12 that if the parameters n , e and r are fixed $(n = 8, e = 1, \text{and } r = 7)$ and the parameter n_{iter} is varied then there is no evident advantage for a specific number n_{iter}.

For $t_{exec} \geq 2.5$ sec , the RANKPRO algorithm is less or equally efficient than the k -prototypes algorithm. The explanation could be the following. For a specified data set, it is assumed that n_{iter} prescribed for the RANKPRO algorithm, is less than the average number of iterations that the k -prototypes algorithm needs to converge. Hence, it is assumed that the RANKPRO algorithm does not spend much time to explore current solutions. For the Connect-4 data set, the k -prototypes algorithm converges very fast contrary to other data sets under consideration. Indeed, the average number of iterations that the k -prototypes algorithm needs to converge is equal to 10.07, 26.29, 24.88 and 1.53 for the Adult, Shuttle, Covertype and Connect-4 data sets respectively. One can see that in the case of Connect-4 data set, the

k-prototypes algorithm converges to a local minimum very fast and it can be applied to the data many times during a prescribed t_{exec}. Hence, the above assumption that the RANKPRO algorithm spends less time than the k-prototypes algorithm to explore local minima is not satisfied and the RANKPRO algorithm loses its advantage. In this case the algorithms have approximately equal effectiveness.

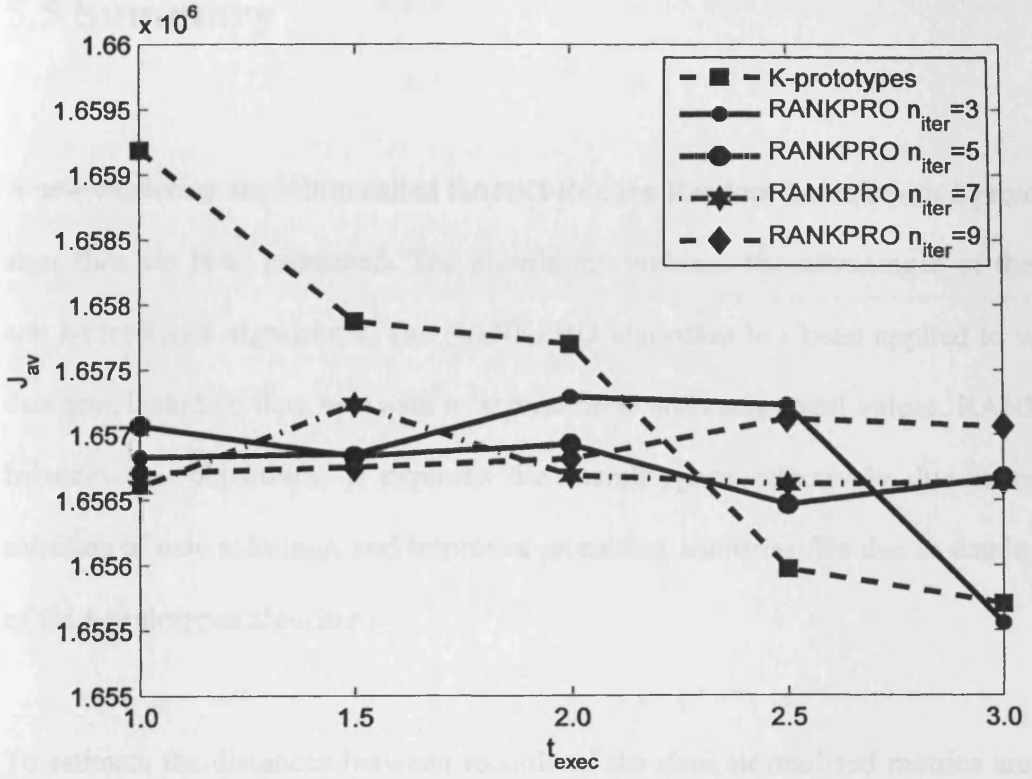


Figure 5.12: The average values of $J_{av}(\mathbf{S})$ vs. t_{exec} for Connect-4 data set. Comparison of the *k*-prototypes algorithm and the RANKPRO algorithm with different values of n_{iter}, n_{iter} = 3, 5, 7 and 9, while other parameters are constant: $n = 8$, $e = 1$, and $r = 7$.

5.5 Summary

A new clustering algorithm called RANKPRO: the Random Search with *k*-prototypes algorithm has been presented. The algorithm combines the advantages of the Bees and *k*-prototypes algorithms. The RANKPRO algorithm has been applied to various data sets, including data sets with mixed numeric and categorical values. RANKPRO balances two objectives: it explores the search space effectively due to random selection of new solutions, and improves promising solutions fast due to employment of the *k*-prototypes algorithm.

To estimate the distances between records of the data, normalised metrics are used. Since, a mixed database is treated as a random sample of an object under consideration, the normalised metrics have been obtained using statistical approach. These normalised metrics are more general than the metric introduced by Huang (1997b) for mixed data sets.

It can be expected that the new RANKPRO algorithm will have less probability for premature convergence than *k*-prototypes algorithm due to the employment of random search. On the other hand, the application of several iterations of the *k*-prototypes algorithm for very fast improvement of the promising (elite) solutions resembles gradient-like methods. Hence, this is a more effective procedure than the attempts to improve the promising solutions by random neighbourhood search as it is used in the BA algorithm.

The obtained results demonstrate the efficiency of the new algorithm. It is shown that in the majority of the considered data sets when the *k*-prototypes algorithm needs many iterations for convergence, the RANKPRO algorithm is more efficient than the *k*-prototypes algorithm. However, if for a specific data set, the *k*-prototypes algorithm converges to a local minimum very fast (just in few iterations) then the algorithms have approximately equal effectiveness.

Chapter 6

Conclusions and Future Work

This chapter concludes the thesis. In this chapter the contributions and conclusions of this thesis are listed and suggestions for future work provided.

6.1. Contributions

The main contributions of this thesis are:

1. A formal and rigorous formulation of accuracy of clustering is introduced. The new approach may be used for an arbitrary number of clusters.
2. The introduction of new normalisation techniques for the Euclidean metric for numerical data. The proposed normalisation procedure secures that the average contributions of all attributes to the measures are equal to each other from statistical point of view and therefore, these variables give equal contributions to the similarity measures.
3. The proposed approach is extended to the case of mixed metrics, i.e. when the metric is a combination of an arbitrary Minkowski metric and the matching dissimilarity measure that are used for numerical and categorical data respectively. Rigorous mathematical proofs of unbiasedness and consistency

of estimators used for normalisation of the Minkowski mixed metrics are presented.

4. Since the k-prototypes algorithm cannot be used in the cases where $p_M \neq 2$, a clustering algorithm with the objective functions $\sum \rho_{p_M}^{p_M}$ that was earlier suggested only for fuzzy clustering, has been developed and applied for hard clustering.
5. A new algorithm RANKPRO that combines the advantages of the Bees and k-prototypes algorithms and outperforms the latter algorithm has been introduced.

Various developed and implemented algorithms have been applied to data sets from the UCI repository.

6.2. Conclusions

The first main result of the thesis (Chapter 3) is the development of a mathematically rigorous approach to normalisation of the feature vectors for mixed data sets based on a unified statistical approach. The most common cases of metrics, namely the Euclidean metrics are used as a measure for continuous numerical features, while the matching dissimilarity measure is used to deal with categorical attributes. The introduced normalised metrics secure that the average contributions of all attributes to the measures are equal to each other from statistical point of view.

The second main result of the thesis (Chapter 4) is application of the unified statistical approach to general cases of the Minkowski distances and the development of a novel algorithm for hard clustering using the mixed Minkowski metrics with an appropriate objective function. The algorithm may be used in these cases, while the k-prototypes is not applicable.

The third main result of the thesis (Chapter 5) is the introduction of the RANKPRO (the Random Search with k-prototypes algorithm). The algorithm combines the advantages of the Bees and k-prototypes algorithms, and outperforms the latter algorithm. The RANKPRO balances two objectives: first it explores the search space effectively due to random selection of new solutions, and on the other hand it improves promising solutions fast due to employment of several steps of the k-prototypes algorithm.

6.3. Future Research Directions

A number of aspects of the algorithms introduced in this thesis could be developed further. Possible extensions include:

Comparison of the effectiveness of the RANKPRO algorithm with several variants of Genetic Algorithms, e.g. Maulik and Bandyopadhyay (2000), in application to clustering of mixed data sets. It is expected that our algorithm will be more effective; however, this has to be confirmed by practical applications.

Comparison of the effectiveness of the RANKPRO algorithm with the Bees Algorithm (Pham et al., 2007), in application to clustering of mixed data sets. According to theoretical arguments, our algorithm will be more effective; however, this has to be confirmed by practical applications.

Estimation of effectiveness of the developed algorithms using the Mann–Whitney–Wilcoxon statistical criterion.

Generalisation of the types of objective functions. If in Chapter 3 and Chapter 5 the standard $\sum \rho_2^2$ objective function has been employed in application to numerical attributes and the function has been extended to the functions $\sum \rho_{p_M}^{p_M}$ in Chapter 4, it is of interest to consider the case $\sum \rho_{p_M}^{\beta}$ for arbitrary β, $\beta \geq 1$ in application to numerical attributes. Evidently there will be a problem of recalculating the new prototypes. However, we can expect that the techniques of optimisation of convex functions may be applied in this case because the function $\sum \rho_{p_M}^{\beta}$ is still convex.

Application of the Bees algorithm to clustering of mixed data sets employing the Minkowski distances and the general $\sum \rho_{p_M}^{\beta}$ objective function.

References

Agresti, A. (1996) *An Introduction to Categorical Data Analysis*, Wiley, New York.

Ahmad, A. and Dey, L. (2007) A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering,* **63**, 503–527.

Aksoy, S. and Haralick, R.M. (2001) Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, **22**, 563-582.

Anderberg, M.R. (1973) *Cluster Analysis for Applications*, Academic Press, New York.

Andreopoulos, B., An, A., Wang, X. and Labudde, D. (2009) Efficient layered density-based clustering of categorical data. *Journal of Biomedical Informatics*, **42**, 365–376.

Asuncion, A. and Newman, D.J. (2007) UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

Bachner, J. (2000) A probabilistic clustering model for variables of mixed type. *Quality & Quantity*, **34**, 223-235.

Banfield, J.D. and Raftery, A.E. (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803-821.

Bandyopadhyay, S. and Maulik, U. (2002) An evolutionary technique based on K-means algorithm for optimal clustering in R^N. *Information Sciences*, **146**, 221-237.

Barbara, D, Couto, J and Li, Y (2002) COOLCAT: an entropy-based algorithm for categorical clustering. In: *Proceedings of the eleventh international conference on Information and knowledge management,* McLean, Virginia, USA, 582 – 589.

Berkhin, P. (2002) Survey of Clustering Data Mining Techniques. *Technical report,* Accrue Software.

Berry, M.J.A. and Linoff, G. (1997) *Data Mining Techniques for Marketing, Sales and Customer Support.* John Wiley & Sons, NY, USA.

Bezdek, J.C. (1980) A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 1–8.

Bezdek, J. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum, New York.

Bezdek, J. (1987) Some non-standard clustering algorithms. In: *Developments in Numerical Ecology, P. and L. Legendre*, Eds., Springer-Verlag, Berlin, 225-287.

Bezdek, J. and Hathaway, R. (1988) Recent convergence results for the fuzzy c-means clustering algorithms. *J. Classification*, **5**, no. 2, 237-247.

Bezdek, R., Hathaway, R., Howard, C., Wilson, and Windham, M. (1987) Local convergence analysis of a grouped variable version of coordinate descent. *J. Opt. Theory Appl.*, **54**, no. 3, 471-477.

Bezdek, J. C. and Pal, S. K. (1992) *Fuzzy Models for Pattern Recognition.* IEEE Press, New York.

Bigot, S. (2003) New Techniques for Handling Continuous Values in Inductive Learning. *Ph.D. thesis, Systems Engineering Division*, University of Cardiff, UK.

Blackard, J.A. and Dean, D.J. (1999) Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, **24**, 131–151.

Bobrowski, L. and Bezdek, J.C. (1991) c-means clustering with the l_1 and l_∞ norms. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**, 545–554.

Bock, H.H. (1996) Probabilistic models in cluster analysis. *Computational Statistics and Data Analysis*, **23**, 5–28.

Bonner, R. (1964) On some clustering techniques. *IBM J. Research and Development,* **8,** 22-32.

Bradley, P.S., Fayyad, U.M. and Reina, C.A. (1999) Scaling EM (Expectation-Maximization) Clustering to Large Databases. *Technical report, MSR-TR-98-35,* Microsoft Research, Seattle.

Brucker, P. (1978) On the complexity of clustering problems. In: *Optimization and Operations Research: Proceedings of the workshop held at the University of Bonn, R. Henn, B. Korte, and W. Oetti, editors,* Springer Verlag, Berlin, *Lecture Notes in Economics and Mathematical Systems*, **157**, 45-54.

Campos, M.M and Milenova, B.L. (2003) Clustering large databases with numeric and nominal values using orthogonal projections. In: *Proceeding of the 29th VLDB Conference*, Berlin.

Chan, E.Y., Ching, W.K., Ng, M.K. and Huang, J.Z. (2004) An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, **37**, 943-952.

Chen. C.-H. (1973) *Statistical Pattern Recognition.* Hayden Book Company. Rochelle Park.

Coppock, S. and Mazlack, L. (2004) Multi-modal Data Fusion: A Description. In: *KES 2004, 8th International Conference on Knowledge-Based Intelligent Information & Engineering,* 1136-1142.

De Jong, K.A. (1975) An analysis of the behavior of a class of genetic adaptive systems. *PhD Thesis*, University of Michigan, Ann Arbor.

Dempster, A., Laird, N. and Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B,* **39**, 1-38.

Doherty, K., Adams, R., Davey, N. (2004) Non-Euclidean Norms and Data Normalisation. In: *Proc. 12th Euro. Symposiumon Artificial Neural Networks, M. Verleysen (Ed.)*, Brugges, Belgium, d-side publications, Brugges, 181-186.

Dom, B. (2001) An information-theoretic external cluster-validity measure. *IBM Research Report RJ 10219,* IBM's Almaden Research Center, San Jose, CA, October 5th 2001.

Domingos, P and Hulten, G. (2001) Learning from infinite data in finite time, In: *Advances in Neural Information Processing Systems 14, T.G. Dietterich, S. Becker, and Z. Ghahramani, editors*, MIT Press, Cambridge, MA, 673-680.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, **26**, 29–41.

Dubes, R. (1993) Cluster analysis and related issues. In: *Handbook of Pattern Recognition and Computer Vision, C. Chen, L. Pau, and P. Wnag, editors*, River Edge, NJ,. World Scientific Publishing Co., 3-32,

Duda, R.O. and Hart, P.E. (1973) *Pattern Classification and Scene Analysis*, Wiley, New York.

Duran, B.S. and Odell, P.L. (1974) *Cluster Analysis. A Survey.* Springer-Verlag, Berlin.

Endo, Y., Murata, R., Toyoda, H., and Miyamoto, S. (2006) L_1-norm based fuzzy clustering for data with tolerance. In: *Proc. 2006 IEEE International Conference on Fuzzy Systems.* IEEE, Vancouver, 770-777.

Estivill-Castro, V. (2002) Why so many clustering algorithms - a position paper. *ACM SIGKDD Explorations Newsletter*, **4**, 65-75.

Estivill-Castro, V. and Murray, A. (1998) Discovering associations in spatial data -an efficient medoid based approach. In: *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98), X. Wu, R. Kotagiri, and K. Korb, editors,* Melbourne, Australia, Springer-Verlag, *Lecture Notes in Artificial Intelligence,* **1394**, 110-121.

Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P. (1996) From data mining to knowledge discovery: An overview. In: *Advances in Knowledge Discovery and Data Mining. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, Eds.,* Menlo Park, CA, AAAI Press, 181-203.

Fisher, D.H. (1987) Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, **2**,139–172.

Fraley, C. and Raftery, A.E. (1998) How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer J.*, **41**, 578- 588.

Fraley, C. and Raftery, A.E. (2002) Model-based clustering, discriminant analysis, and density estimation. *J. American Statistical Association*, **97**, 611–631.

Ganti, V., Gehrke, J. and Ramakrishnan R. (1999) CACTUS - Clustering Categorical Data Using Summaries. In *Proceedings of the fifth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 73–83.

Garcke, J., Griebel, M. and Thess, M. (2001) Data mining with sparse grids. *Computing*, **67**, 225-253.

Gibert, K. and Nonell, R. (2003) Impact of Mixed Metrics on Clustering. In: CIARP *Lecture Notes in Computer Science,* **2905**, 464-471

Gibert, K. and Cortes, U. (1997) Weighing quantitative and qualitative variables in clustering methods. *Mathware and Soft Computing*, **4**, 251–266.

Gibson, D., Kleingberg, J.and Raghavan, P. (2000) Clustering categorical data: An approach based on dynamical systems, *The International Journal on Very Large Data Bases,* **8,** 222-236.

Giudici, P. (2003) *Applied Data Mining: Statistical Methods for Business and Industry.* John Wiley & Sons, Chichester.

Goldberg, D.E. (1989) *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading.

Goldberg, D.E. and Deb, K. (1991) A comparison of selection schemes used in genetic algorithms. In: *Proceedings of the First Workshop on Foundations of Genetic Algorithms, Ed.: G.J.E. Rawlins* , Morgan Kaufmann, San Fransisco, 69–93.

Goodall, D.W. (1966) A new similarity index based on probability. *Biometrics*, **22,** 882-907.

Gowda, K. and Diday, E. (1991) Symbolic clustering using a new dissimilarity measure. *Pattern Recognition*, **24**, 567-578.

Gower, J. (1971) A general coefficient of similarity and some of its properties. *Biometrics*, **27**, 857-871.

Guha, S., Rastogi, R. and Shim, K. (1997) A Clustering Algorithm for Categorical Attributes, *Technical Report*, Bell Laboratories, Murray Hill.

Guha, S., Rastogi, R. and Shim, K. (1998) CURE: An efficient clustering algorithm for large databases, In *Proc. ACM-SIGMOD Int. Conf. Management of DATA (SIGMOD'98)*, Seattle, WA, 73-84.

Guha, S., Rastogi, R. and Shim, K. (1999) ROCK: A robust clustering algorithm for categorical attributes. In: *Proceedings of the 15th International Conference on Data Engineering*, Sydney, Australia, 512-521.

Gupta, S., Rao, K. and Bhatnagar, V. (1999) K-Means clustering algorithm for categorical attributes. In: Data warehousing and knowledge discovery, pp. 203-208.

Guyon, I and Elisseeff, A. (2006) An Introduction to Feature Extraction. In: *Studies in Fuzziness and Soft Computing*, Springer, Berlin, **207**, 1-25.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001) On clustering validation techniques. *J. Intelligent Info. Syst.*, **17**, 107-145.

Halkidi, M., Vazirgianis, M. and Batistakis, Y. (2000) Quality scheme assessment in the clustering process. In: *Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD, H. Zighed, D.A. Komorowski and J. Zytkow, editors,* Lyon, France, Springer Verlag, *Lecture Notes in Computer Science*, **1920**, 265-276.

Hastie, T., Tibshirani, R., Friedman, J. (2001) *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Berlin: Springer.

Hathaway, R.J. and Bezdek, J.C. (1995) Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems*, **3**, 241-245.

Huang, Z. (1997a) A fast clustering algorithm to cluster very large categorical data sets in data mining. In: *Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Dep. of Computer Science, The University of British Columbia, 1–8.

Huang, Z. (1997b) Clustering large data sets with mixed numeric and categorical values. In: *The First Pacific-Asia Conference on Knowledge Discovery and Data Mining. H. Lu, H. Motoda and H. Liu, Eds.*, World Scientific, Singapore, 21-34.

Huang, Z. (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, **2**, 283–304.

Huang, Z. and Ng, M.K. (1999) A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, **7**, 446- 452

Huang, J.Z., Ng, M.K., Rong, H. and Li, Z. (2005) Automated variable weighting in k-means type clustering. *IEEE Trans. Pattern Anal. Machine Intelligence*, **27**, 657-668.

Hubert, L. and Arabie, P. (1985) Comparing partitions. *Journal of Classification*, **2**, 193–218.

Jain, A.K. and Dubes, R.C. (1988) *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs.

Jain, A.K., Murty, M.N., and Flynn, P.J. (1999) Data clustering: A review. *ACM Computing Surveys*, **31**. 264-323

Jensen, R. (1969) A dynamic programming algorithm for cluster analysis. *Operations research*, **17**, 1034–1057

Kaelo, P. and Ali, L.C.W. (2006) Some variants of the controlled random search algorithm for global optimization. *J. Optimization Theory and Applications.* **130**, 253-264.

Kalbfleisch, J. (1985) *Probability and Statistical Inference.* Vol.2: *Statistical Inference*. Springer-Verlag, NY.

Kaufman, L. and Rousseeuw, P.J. (1990) Finding groups in data: an introduction to cluster analysis, Wiley, New York.

Kim, J., Bezdek, T. and Hathaway, E. (1988) Optimality tests for fixed points for the fuzzy c-means algorithm. *Patt. Recog.*, **21**, 651-663.

Koga, T., Miyamoto, S. and Takata, O. (2001) Fuzzy c-means and mixture distribution model for clustering based on L_1-space. In: *JSAI 2001 Workshops, LNAI 2253, Eds.: T. Terano et al.,* Springer-Verlag, Berlin, 289-294.

Krishna, K.K. and Murty, M.N. (1999) Genetic K-means algorithm. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, **29**, 433–439.

Larose, D.T. (2005) *Discovering knowledge in data: an introduction to data mining.* Wiley, Hoboken.

Lavine, B.K. (2000) Clustering and Classification of Analytical Data. In: *Encyclopaedia of Analytical Chemistry: Instrumentation and Applications*, John Wiley & Sons Ltd., Chichester, 9689-9710.

Lebowitz, M. (1987) Experiments with incremental concept formation. *Machine Learning*, **2**,103–138.

Liu, H. and Huang, S.-T. (2003) Evolutionary semi-supervised fuzzy clustering. *Pattern Recognition Letters*. **24**, 3105 - 3113

Looney, C.G. (1997) *Pattern Recognition Using Neural Networks*. Oxford University Press, New York.

MacQueen, J.B. (1967) Some methods for classification and analysis of multivariate

observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, **1**, University of California Press, Berkeley, 281–297.

Malerba, D., Sanarico, L. and Tamma, V. (2000) A comparison of dissimilarity measures for symbolic data analysis. In: *Proceedings of the ECML'2000 workshop on dealing with structured data in machine learning and statistics*, Barcelona, Spain, May 30, 2000

Mali, K. and Mitra, S. (2002) Clustering of symbolic data and its validation. In: *Proceedings of the 2002 AFSS International Conference on Fuzzy Systems*, Calcutta, *Advances in Soft Computing*, 339-344.

Maulik, U. and Bandyopadhyay, S. (2000) Genetic algorithm-based clustering technique. *Pattern Recognition*, **32**, 1455-1465.

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin.

Michalski, R.S and Stepp, R.E. (1983) Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**, 396–410.

Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (1994) *Machine learning, neural and statistical classification*, Prentice Hall, Englewood Cliffs, NJ.

Milligan, G.W. and Cooper, M.C. (1988) A study of variable standardization in cluster analysis. *Journal of Classification*, **5**, 181–204.

Mirkin, B. (1996) *Mathematical Classification and Clustering*, Kluwer Academic Press, Dordrecht.

Mirkin, B. (1997) L1 and L2 approximation clustering for mixed data: Scatter decompositions and algorithms. *L1 -Statistical Procedures and Related Topics*, **31**, 473-486.

Mirkin, B. (1998) Least-squares structuring, clustering, and data processing issues. *Computer Journal*, **41**, 518-536.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, USA.

Mitchell, T. (1999) Machine learning and data mining. *Communications of the ACM*, **42**, 30-36.

Mitchell, T. M. (1997) *Machine Learning.* McGraw Hill, New York.

Miyamoto, S. and Agusta, Y. (1995) An efficient algorithm for l_1 fuzzy c-means and its termination. *Control and Cybernetics*, **24**, 421-436.

Miyamoto, S. and Agusta, Y. (1998) Algorithms for L1 and Lp fuzzy c-means and their convergence, In: *Studies in Classification, Data Analysis, and Knowledge Organization; Data Science, Classification, and Related Methods. Eds.: C. Hayashi, N. Ohsumi, K. Yajima, Y. Tamaka, H.H. Bock, Y. Baba.* Springer-Verlag, Tokyo, 295-302.

Müller, A. (2002) *Finding Groups in Large Data Sets*, CEPE Working Paper No. 18, Zurich, 1-17.

Murthy, C. A. and Chowdhury N. (1996) In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, **17**, 825-832

Neal, R.M. and Hinton, G.E. (1999) A view of the EM algorithm that justifies incremental, sparse, and other variants. In: *Learning in Graphical Models, M. I. Jordan (ed.),* MIT Press.

Ng, R .T. and Han, J. (1994) Efficient and effective clustering methods for spatial data mining. In: *Proceedings of Very Large Data Bases Conference,* 144-155

Ng, M.K. and Wong, J.C. (2002) Clustering categorical data sets using tabu search techniques. *Pattern Recognition*, **35**, 2783-2790.

Ordonez, C. and Omiecinski, E. (2002) FREM: Fast and robust EM clustering for large data sets. In: *Proc. ACM Conf. Information and Knowledge Management,* 590 – 599.

Pal, N.R., Pal, K. and Bezdek, J.C. (1997) A mixed c-means clustering model, In: *IEEE Int. Conf. Fuzzy Systems*, Spain, 11–21.

Park, N.H., Ahn, C.W. and Ramakrishna, R.S (2005) Adaptive clustering technique using genetic algorithms. *IEICE - Transactions on Information and Systems*, **E88-D**, 2880-2882.

Pedrycz, W. (2005) *Knowledge-Based Clustering: From Data to Information Granules.* Wiley-Interscience, Hoboken.

Peters, M. and Zaki, M.J. (2004) CLICK: Clustering Categorical Data using K-partite Maximal Cliques TR 04-11, CS Dept., RPI. 1-31.

Pham, D. T. and Afify, A.A. (2007) Clustering techniques and their applications in engineering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **221**, 1445-1459.

Pham, D.T. and Jin, G. (1995) Genetic algorithm using gradient-like reproduction operator. *Electronics Letters*, **31**, 1558–1559.

Pham, D.T., Suarez-Alvarez, M.M. and Prostov, Y.I. (2006a) Statistical approach to numerical databases: clustering using normalised Minkowski metrics. In: *Intelligent Production Machines and Systems. Proceedings of 2nd I*PROMS Virtual Conference, 3-14 July 2006*, Elsevier, Amsterdam, 356–361.

Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M. (2006b) The Bees algorithm - a novel tool for complex optimisation problems. In: *Intelligent Production Machines and Systems. Proceedings of 2nd I*PROMS Virtual Conference, 3-14 July 2006*, Elsevier, Amsterdam, 454–461.

Pham, D.T., Otri, S., Afify, A.A., Mahmuddin, M. and Al-Jabbouli, H. (2007) Data clustering using the Bees Algorithm, In: *Proc. 40th CIRP Int. Manufacturing Systems Seminar*, Liverpool, 1–8.

Pham, D.T., Suarez-Alvarez, M.M. and Prostov, Y.I.(2009) Normalisation of feature vectors and clustering of mixed data sets. *submitted.*

Polyak, B.T. (1987) *Introduction to optimization.* Optimization Software.

Pregibon, D. and Elder, J. (1996) A statistical perspective on knowledge discovery in databases. In: *Advances in Knowledge Discovery and Data Mining*, Fayyad, U. G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy(Eds.), 83-116. MIT Press.

Price, W.L. (1978) Global optimization by controlled random search. *J. Optimization Theory and Applications*, **40**, 333–348.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

Ralambondrainy, H. (1995) A conceptual version of the *K*-means algorithm. *Pattern Recognition Letters*, **16**, 1147–1157.

Rand, W.M. (1971) Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**, 846–850.

Russell, S.J. and Norvig, P. (2003) *Artificial Intelligence: A Modern Approach.* Prentice Hall, Upper Saddle River, NJ.

San, O.M., Huynh, V.-N. and Nakamori, Y. (2004) An alternative extension of the k-means algorithm for clustering categorical data. *Int. J. Appl. Math. Comput. Sci.*, **14**, 241-247.

Spiegel M.R. (1975) *Schaum's Outline of Theory and Problems of Probability and Statistics*. McGraw-Hill, New York.

Stevens, S.S. (1946) On the theory of scales of measurement. *Science*, **103**, 677-680.

Takata, O., Miyamoto, S. and Umayahara, K. (2001) Fuzzy clustering of data with uncertainties using minimum and maximum distances based on L_1 metric. *IEEE,* 2511-2516.

Varmuza, K. and Filzmoser, P. (2009) *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, Boca Raton.

Wen, J.Y., Wu, Q.H., Jiang, L. and Cheng, S.J. (2003) Pseudo-gradient based evolutionary programming. *Electronics Letters*, **39**, 631–632.

Wu, K.-L. and Yang, M.-S. (2002) Alternative c-means clustering algorithms. *Pattern Recognition*, **35**, 2267–2278.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z., Steinbach, M., Hand, D. J., and Steinberg, D. (2007) Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **14**, 1-37.

Zhang, P., Wang, X. and Song, P. X.-K. (2006) Clustering categorical data based on distance vectors. *Journal of the American Statistical Association*, **101**, 355-367.

Zhang, T., Ramakrishnan, R. and Livny, M. (1997) BIRCH: A new data clustering algorithm and its applications, *Data Mining and Knowledge Discovery*, **1**, 141-182.

Zhang, T., Ramakrishnan R and Livny M. (1996) Birch: An Efficient Data Clustering Method for Very Large Databases.In: *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96),* 103-114.

Zhang, Y., Fu, A., Cai, C. and Heng, P. (2000) Clustering categorical data. In: *Proceedings of the 16th Int. Conf. Data Eng.*, 305, San Diego, CA.

Appendix A

Data Sets

All data sets used in this thesis are from the UCI repository of machine learning databases [Blake and Merz, 1998]. These databases were contributed by many researchers, mostly from the field of machine learning, and collected by the machine learning group at the University of California, Irvin. These data sets are described briefly below.

Vote data set. The database includes votes for each of the U.S. House of Representatives Congressmen on 16 key votes, such as water project cost sharing, crime and duty-free exports. The problem is to identify whether a person is a republican or a democrat based on these votes.

Chess data set. This database has 36 features to describe chess board positions and the task is to determine which position will lead to a win.

Crx data set. This data set was originally used by Quinlan on the C4.5 induction learning algorithm. The data is used to determine whether or not to give a credit card to an applicant. All the feature names and values have been changed to meaningless symbols to protect the confidentiality of the data.

Horse Colic data set. There are 368 instances in this data set. 22 features are used to describe information about the horses, including their age, pulse, rectal temperature etc, and the task is to classify whether a lesion is surgical or not.

Hypothyroid data set. The data comes from an assay screening service related to thyroid functions, and concerns one aspect of thyroid diagnosis. The 25 features are a mixture of measured values and information obtained from the referring physician. There are four classes.

Annealing data set. The application concerns appropriate actions to take during the coating of steel products. The data set contains 898 cases described in terms of 38 features that cover aspects such as the width of the steel slab, its type, hardness, composition, surface quality etc. There are five classes corresponding to alternative coating sub-procedures.

Hepatitis data set. The data contains 155 instances; each instance is represented by 19 features, describing the age, sex and other 17 attributes of a patient. The task is to determine whether the patient has a risk of death.

Mushroom data set. This data base consists of descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota family. Each species is identified as definitely edible or definitely poisonous. There are 8124 records, and each record is described by 22 nominally valued features.

Soybean-large data set. The data set consists of 683 records with 35 features, describing leaf properties and various abnormalities. The task is to diagnose soybean disease based on the measures and observations.

Vehicle data set. The data set is used to classify a given silhouette as one of four types vehicle using a set of features extracted from the silhouette. Each vehicle is described by 18 continuous valued features.

Diabetes data set. There are 768 instances in the data set; each is described by 8 continuous valued attributes, such as the number of times pregnancies, diastolic blood pressure, body mass index, etc. The data is used to classify whether the patient tested is positive or negative for diabetes.

Breast Cancer data set. The *breast cancer* data contains 699 cases. Each case is described by 10 continuous attributes that cover aspects such as the age of the patient, tumour size, menopause etc. There are two classes which identify whether the tumour is benign or malignant.

Iris data set. This is the most widely used data set in the literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Each instance is described by four continuous attributes, namely, sepal length, sepal width, petal length and petal width.

Abalone data set. The *abalone* data is used to predict the age of abalone from physical measurements. There are a total of 4177 instances in the data, and each is described by 8 attributes.

Adult data set. There are 48842 instances in the data. Each instance is described by 14 attributes, such as age, work class, native country, education, marital status and so on. These attributes are used to predict whether such a person can earn a salary greater or less than $50,000 in the USA.

Australian data set. The Australian data is almost the same as the original *Crx* data, but all the missing values have been replaced with their medians.

Car data set. The car evaluation data set is used to evaluate cars according to the features that describe their price, technical characteristics, and safety. There are a total of 1728 instances, each described by 6 attributes and categorised into one of 4 classes.

Appendix B

Proof of unbiasedness and consistency of estimators used for normalisation of the Minkowski mixed metrics

Here a rigorous proof is given for the above statements that the estimators (4.6) and (4.13) for the mean contribution of each j-th attribute (in numerical and categorical cases respectively) to the Minkowski mixed metric are unbiased and consistent. More precisely, these statements are corollaries of the following general Proposition.

Proposition B1. Let a random variable X have a distribution law $\mathrm{L}(X)$ and $\{x_1,\ldots,x_N\}$ be a sample of its values. Let Z_1 and Z_2 be independent random variables having the same distribution law $\mathrm{L}(X)$. Let $\varphi(z_1,z_2)$ be a function of two real valued arguments such that the random variable $Z=\varphi(Z_1,Z_2)$ has finite mean and variance. Then the estimator $\hat{E}Z$ of the mean of the random variable Z given by

$$\hat{E}Z=\frac{2}{N(N-1)}\sum_{1\le i<j\le N}\varphi(x_i,x_j), \tag{B.1}$$

is unbiased and consistent.

To proof Proposition B1 one needs to use two Lemmas. The first lemma is often used in mathematical statistics.

Lemma 1. If $\hat{\theta}_n$ is an unbiased estimator of a parameter θ and the variance of $\hat{\theta}_n$ goes to 0 for $n \to \infty$. Then $\hat{\theta}_n$ is also a consistent estimator.

Proof. The estimator $\hat{\theta}_n$ is consistent if for any $\varepsilon > 0$

$$\lim_{n\to\infty} P\left\{|\hat{\theta}_n - \theta| > \varepsilon\right\} = 0. \tag{B.2}$$

Due to the Tchebysheff inequality and since the estimator $\hat{\theta}_n$ is unbiased, we obtain

$$P\left\{|\hat{\theta}_n - \theta| > \varepsilon\right\} = P\left\{|\hat{\theta}_n - E\theta| > \varepsilon\right\} \le \frac{D\hat{\theta}_n}{\varepsilon^2}$$

Since it follows from the formulation that $\lim_{n\to\infty} D\hat{\theta}_n = 0$, (4.15) follows from the above inequality and this proves Lemma 1.

Let us denote by T_N the set of all pairs of indices (i, j) such that $1 \le i < j \le N$.

Let us call a subset $U \subset \mathrm{T}_N$ admissible if for any arbitrary two distinct pairs (i_1, j_1) and $(i_2, j_2); (i_1, j_1), (i_2, j_2) \in U$ all indices $\{i_1, j_1, i_2, j_2\}$ are distinct.

The following Lemma was obtained by M.I. Prostov and with his permission the proof is given here for the sake of completeness.

Lemma 2. The set T_N can be divided into $\lambda(N)$ non-overlapping admissible subset $U_1,\ldots,U_{\lambda(N)}$, such that there are $\mu(N)$ elements in each subset of the partition. The numbers $\lambda(N)$ and $\mu(N)$ are defined as $\lambda(N)=N$, and $\mu(N)=(N-1)/2$ for odd N; and $\lambda(N)=N-1$, and $\mu(N)=N/2$ for even N.

Let us give examples of such partitions:

a) $T_4 = \bigcup_{i=1}^{3} U_i$, where $U_1=\{(1,2),(3,4)\}$, $U_2=\{(1,3),(2,4)\}$ and $U_3=\{(1,4),(2,3)\}$.

b) $T_5 = \bigcup_{i=1}^{5} U_i$, where $U_1=\{(1,2),(3,4)\}$, $U_2=\{(1,3),(4,5)\}$, $U_3=\{(1,4),(2,5)\}$, $U_4=\{(1,5),(2,3)\}$ and $U_5=\{(1,4),(3,5)\}$.

Proof of Lemma 2.

Let us consider a circle on a plane with a unit radius and with centre (0,0). Let the circle have N points $A_1,\ldots,A_N$. These points are the corners of a regular polygon. Let us denote by C_N the set of all chords connecting the points $A_1,\ldots,A_N$. Let each pair $(i,j)\in T_N$ correspond to the chord $\psi(i,j)=A_iA_j$ of the circle. One can see that there is a one-to-one correspondence ψ between the sets T_N and C_N.

Let us consider the case of odd N, i.e. $N=2K+1$. Let

$U_1 = \{(i, 2K+1-(i-2)) : i = 2,\ldots,K\}$ and $V_1 = \psi(U_1)$ be the set of chords corresponding to pairs of indices U_1 (see Fig. 4.1a). Further, we denote by V_m, $(m = 2,\ldots,K)$ the set of chords, that can be obtained from chords of the set V_1 by counter clockwise rotation with an angle $2\pi(m-1)/(2K+1)$, and put $U_m = \psi^{-1}(V_m)$ (see Fig. 4.1b).

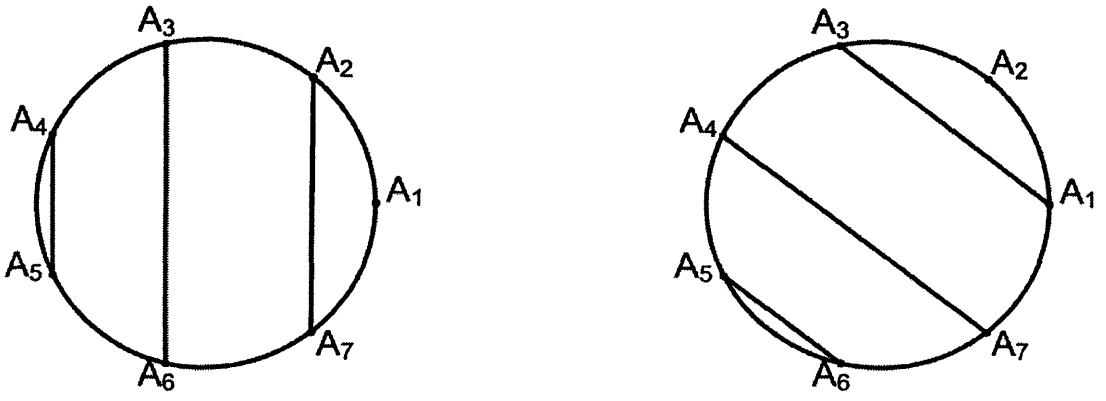


Figure B.1: Sets V_1 and V_2 for $N = 7$: a) the set V_1 and b) the set V_2.

Since the chords of the set V_1 are parallel to each other, the chords of each of the sets $V_m (m = 2,\ldots,K)$ are also parallel to each other. Hence, all sets $V_1,\ldots,V_K$ are pair wise non- overlapping, and consequently $U_1,\ldots,U_N$ are also pair wise non-overlapping. In addition, since the cords of each of the sets V_m do not have the mutual end points, all sets $U_1,\ldots,U_N$ are admissible.

Further, each of the sets U_m has $2K+1$ elements, therefore, their union has $K(2K+1)=N(N-1)/2$ elements. Thus, we obtain $\bigcup_{m=1}^{K} U_m = \mathrm{T}_N$ and Lemma 2 has been proved for odd N.

Let us consider now the case of even N, i.e. $N=2K$. It has been shown above that the set T_{2K-1} can be divided into $2K-1$ admissible subsets $U'_1,\ldots,U'_{2K-1}$ such that each of these subsets has $K-1$ pairs of indices whose values are less or equal to $N-1$.

Consider the set U'_m, $(m=1,\ldots,2K-1)$ and the corresponding set of chords V'_m. There exists exactly one point $A_{\nu(m)}$ with $1\le\nu(m)\le 2K-1$ that is not the end point of any of the chords of the set V'_m. Hence, the chord $A_{\nu(m)}A_N$ does not belong to V'_m and therefore the set $V_m = V'_m \bigcup \{A_{\nu(m)}A_N\}$ consists of chords that do not have mutual end points. Consequently, the set $U_m = \psi(V_m)$ is admissible. Further, since all points $A_{\nu(1)},\ldots,A_{\nu(2K-1)}$ are distinct, the chords $A_{\nu(1)}A_{2K},\ldots,A_{\nu(2K-1)}A_{2K}$ are also distinct. This leads to the conclusion that the sets $V_1,\ldots,V_{2K-1}$ are pair wise non-overlapping, and consequently $U_1,\ldots,U_{2K-1}$ are also pair wise non-overlapping. Each of the sets U_m has K pairs of indices, and the union of these sets has $K(2K-1)=N(N-1)/2$ pairs. Thus, we obtain

$\bigcup_{m=1}^{K} U_m = \mathrm{T}_N$ and this proves Lemma 2.

Proof of Proposition B1. Let $X_1,\ldots,X_N$ be independent random variables having a distribution law $\mathrm{L}(X)$. Since $E[\varphi(X_i,X_j)] = E[\varphi(Z_1,Z_2)]$ for any $i \neq j$, it follows from the equality

$$E\left[\frac{2}{N(N-1)}\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right]=\frac{2}{N(N-1)}\sum_{1\le i<j\le N}E[\varphi(X_i,X_j)]=E[\varphi(Z_1,Z_2)],$$

that the estimator (B.1) is unbiased.

Further, let us estimate the variance (dispersion)

$$D\left[\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right] \tag{B.3}$$

where D denoted the dispersion.

It follows from Lemma 2 that

$$\sum_{1\le i<j\le N}\varphi(X_i,X_j)=\sum_{1\le m\le\lambda(N)}\sum_{(i,j)\in U_m}\varphi(X_i,X_j), \tag{B.4}$$

where each of sets U_m is admissible and consists of $\mu(N)$ pairs of indices. For each m, the above sum $\sum_{(i,j)\in U_m}\varphi(X_i,X_j)$ consists of $\mu(N)$ independent random variables having the same law of distribution and therefore, we have for (B.3)

$$D[\sum_{(i,j)\in U_m}\varphi(X_i,X_j)]=\mu(N)D[\varphi(Z_1,Z_2)] \tag{B.5}$$

For independent random variables $\xi_1,\ldots,\xi_N$ having the same law of distribution $L(\xi))$, we have

$$D[\xi_1+\ldots+\xi_N]=\sum_{1\le i,j\le N} Cov(\xi_i,\xi_j)\le \sum_{1\le i,j\le N}\sqrt{D\xi_i}\sqrt{D\xi_j}=N^2 D\xi.$$

From the above inequality along with (B.4) and (B.5), we obtain

$$D\left[\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right]\le \lambda(N)^2\mu(N)D[\varphi(Z_1,Z_2)]\le (N^3/2)D[\varphi(Z_1,Z_2)], \qquad \text{(B.6)}$$

From (B.6), we obtain

$$D\left[\frac{2}{N(N-1)}\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right]=\frac{4}{N^2(N-1)^2}D\left[\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right]$$
$$\le\frac{2N}{(N-1)^2}D[\varphi(Z_1,Z_2)]. \qquad \text{(B.7)}$$

It follows from (B.7) that

$$\lim_{N\to\infty} D\left[\frac{2}{N(N-1)}\sum_{1\le i<j\le N}\varphi(X_i,X_j)\right]=0. \qquad \text{(B.8)}$$

Using Lemma 1 and (B.8), we obtain Proposition B1.