Diplomarbeit

zum Thema

Cluster Analysis with Application to Data from Geochemistry

ausgeführt am
Institut für Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter der Anleitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

durch
Matthias Templ
Matr.Nr.: 9656209
Tannenstraße 5
4481 Asten

Wien, am 12. Mai 2003

Preface

The goal of this master's thesis is to give a deeper insight with the help of cluster analysis and fuzzy cluster analysis into the large data sets of the C-horizon and the O-horizon mineral soil samples taken from a big area in a regional study in the European Arctic, and into the large data set of the Walchen data taken from a regional study in Austria. The aim of cluster analysis is to find groups in data, in which objects of the same group should be as similar to each other as possible, whereas objects of different groups should be as dissimilar as possible.

By clustering a number of elements, various groups of rock types are to be recognised in the data set of the C-horizon. Then they are to be compared with a lithological map. We have two goals when clustering elements from the O-horizon: firstly, to localise the environmental pollution from heavy industry and secondly, to illustrate the influence of weather (especially in connection with the sea) on the chemical composition of the O-horizon. The distinction of various rock types is the aim of clustering the Walchen data.

The first chapter explains the steps necessary for doing cluster analysis. The selection of variables is of crucial importance for the clarity of the groups produced by the clustering process. We will also see that it is necessary to standardise the data before clustering. In the second chapter the cluster algorithms are described. In Chapter 3 a detailed description of the data is given. The fourth chapter contains an evaluation of the quality of the clustering. We distinguish between external, internal and relative criteria. A small selection of the results of the clustering and their qualities is presented in Chapter 5. Additionally, further results are to be found in Appendix A. Appendix B contains a description of the software used. Furthermore, the written programmes are put at everybody's disposal.

Acknowledgements

My special thanks go to my promoter Prof. Dr. Peter Filzmoser for his help and professional advice. His infective enthusiasm and motivation and the pleasant working atmosphere greatly contributed to the success of this thesis.
I also would like to thank Dipl.-Ing. Herbert Guttmann for his generous help concerning software problems.
Thanks to Dr.mont. Clemens Reimann from the Geological Survey of Norway (NGU) for taking his time in the discussions on the results.
Special thanks to Aleksandra Gleich who helped me with my English and for her grammar correction of this paper.
Furthermore, I would like to thank my parents Maria and Franz Templ whose help and support made these studies possible at all.
Finally, I would like to thank my brother Robert who mostly contributed to my decision to enroll at the Technical University of Vienna.

Contents

List of Figures

List of Tables

Chapter 1

Preparations for Cluster Analysis

1.1 Introduction

Cluster analysis serves as a statistical technique for assigning a number of objects into several groups. Objects can be characterised by more than one attribute. An example for an object could be measurements of a soil sample with attributes iron, copper and aluminium. In this case the object has got three dimensions. Objects in the same group (class, cluster) should be as similar to each other as possible (homogeneity inside the class), whereas objects in different groups should be as dissimilar as possible (dissimilarity between the classes).
The boom of cluster analysis started approximately 40 years ago as an aftereffect of increasing computer capacity. The classification of similar objects into groups has many applications, e.g. in archaeology, where the large number of measurements demands clusters in order to classify newly founded fossils. Further applications can be found in social studies (e.g. typologies of social objects as a tool to describe their behaviour), in biology (e.g. studies on animal movements or on bird territories), in medical science (e.g. to classify diseases), in linguistics (e.g. to define isogloss, see Filzmoser, 1993), in marketing (e.g. to separate groups in customer bases), in bio-informatics (to predict gene functions), in city-planning (to identify groups of houses according to their house type, value, and geographical location) and in many other scientific disciplines.
Cluster analysis and its application in geology is the topic of many publications, e.g. Everitt (1974), Clark (1979), Howarth (1983), Pielou (1984), Davis (1986), Jongman et al. (1987), Rock (1988), Swan and Sandilands (1995), Legendre and Legendre (1998), McGarigal et al. (2000).

1.2 Procedures of Cluster Analysis

This section provides an overview of the initial steps of cluster analysis. A sketch of stages in a clustering study (Fayyad et al., 1996) is given below, although it should be noted that

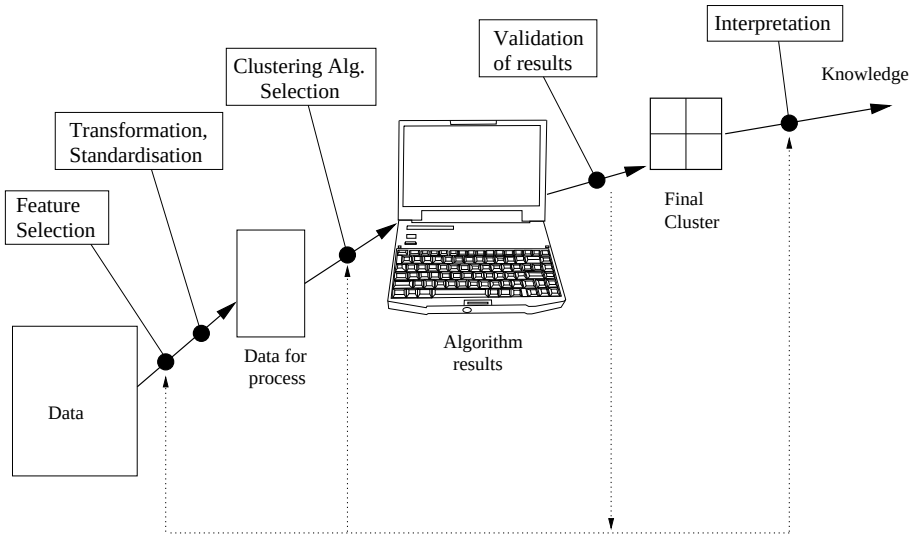


Figure 1.1: Steps of the clustering process

some studies are concerned with only a subset of the stages shown in the Figure 1.1. Similar suggestions are given in Pirktl (1983), Gordon (1999), Jain et al. (1999), Halkidi et al. (2001) and Bradley et al. (1999).
Cluster analysis consists of the following procedures:

- The choice of objects,
- determination of variables (see Sections 1.4, 3.2.1),
- transformation and standardisation of variables (see Section 1.7),
- determination of the distance measure (see Section 1.8),
- the choice and implementation of the clustering algorithm(s),
- evaluation of the results with the help of cluster validity measures (see Chapter 4),
- interpretation of the clustering results.

1.3 Data Format

A raw data matrix with n rows, which correspond to the objects and p columns, which correspond to the variables, forms the starting point for cluster analysis. In our case of the C-horizon data the raw data matrix contains for each of the 605 examined samples 89 measured concentrations of chemical elements.
If the j-th measurement of the i-th object is denoted by x_{ij} $(i = 1, \ldots, n; j = 1, \ldots, p)$ this matrix looks like

$$\mathrm{X} = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{pmatrix}$$

1.4 Selection of Variables and Dimension Reduction

Some scientists have the view that a large number of underlying variables should be used, in order not to exclude some that are possibly relevant. Gordon (1999) has shown that this is not recommendable for some cases. Only those variables that are believed to be helpful for grouping the data should be included in the analysis. The addition of only one or two irrelevant variables can have dramatical consequences in finding the clusters. The inclusion of one irrelevant variable may serve to mask or hide real clustering in the data. In that case, we speak about *masking variables*. Usually, clustering, if it exists occurs only within a relatively small unknown subset of attributes.

On the internet site *http://www.fas.umontreal.ca/biol/casgrain/en/labo/ovw.html* one can find a program from Makarenkov and Legendre (2001) for optimal variable weighting based on the idea of De Soete (1986). This program gives high weights to those attributes that most exhibit clustering on the objects and small weights to those that do not participate in the clustering.

An extension of this approach has been done by Friedman and Meulman (2001). They consider different subsets of the attributes (variables) and compute the weight for each variable by minimising an objective function. Afterwards, that subset is chosen which results in a minimum over all objective functions. This approach is called COSA (Clustering Objects on Subsets of Attributes), and a software package written in `R` is freely available at *http://www-stat.stanford.edu/∼jhf/COSA.html*. For the underlying data of the C-horizon this method did not lead to satisfying results.

In fact the selection of variables reduces the dimensionality of the data for clustering. Dimension reduction, however, can also be achieved by multivariate techniques, like Principal Component Analysis (PCA). This approach was suggested by Everitt (1974). PCA is first performed on the data, and then only the first few principal component scores are used as input for cluster analysis. But for most clustering methods, one does not assume that the data follow a normal distribution nor that there are uncorrelated variables. However, the routine application of PCA or other factoring techniques prior to clustering is naive (see Milligan, 1980). Clusters embedded in a high-dimensional variable space will not be properly represented by a smaller number of orthogonal components (see e.g. Yeung and Ruzzo, 2001). The transformation process may result in partial distortion of the true clustering, may fail to find distinct clusters, or may even result in completely meaningless partitions.

In our experience, the objective function for most clustering techniques which were used in this diploma thesis resulted in a worse value if the clustering was performed on the significant principal components.

1.5 Outlier Detection

Outliers in a data set can have severe influence on statistical methods. This is also true for cluster analysis because they can negatively affect proximity measures and eliminate clustering tendency. Outliers can be viewed as legitimate records having abnormal behaviour. Statistics defines outliers as observations that do not fit the underlying probability distribution. There are some interests in eliminating negative effects of outliers on the cluster construction. Most of the research of outlier detection is not directly related to clustering. A common method is to downweight outlying observations, where the weights are chosen according to a robust distance measure. One can use for example the Mahalanobis distance with robust estimations of centre and covariance (see Rousseeuw and van Zomeren, 1990).

1.6 Data Transformation

Although cluster analysis, in general, does not need normally distributed data, some clustering methods assume that the data are multivariate normally distributed. Hence, a transformation should be applied which turns the data distribution to approximate normal distribution.

Most of the modern textbooks on geochemistry still claim that geochemical data is close to a log-normal distribution. This is the reason why log-transformation is widely used when working with geochemical data. Reimann and Filzmoser (1999) have done a detailed investigation for different geochemical data sets whether the elements are normally or log-normally distributed. Using different statistical tests, it turned out that log-normal distribution could only rarely be accepted.

A demonstration with the element Magnesium (Mg) should give a short insight about the distribution of this element. Figures 1.2 to 1.4 show the histograms of original and log-transformed Mg in three different media. Due to Figure 1.5 we conclude that for Mg in C-horizon and in O-horizon the log-transformation was able to turn the distribution close to normal distribution.

These examples should demonstrate that log-transformation is not necessarily advantageous. At the contrary, log-transformation can even turn the data distribution further away from normality.

There are of course other possibilities for transformation, like power transformations. A general transformation procedure is the Box-Cox transformation (see Box and Cox, 1964).

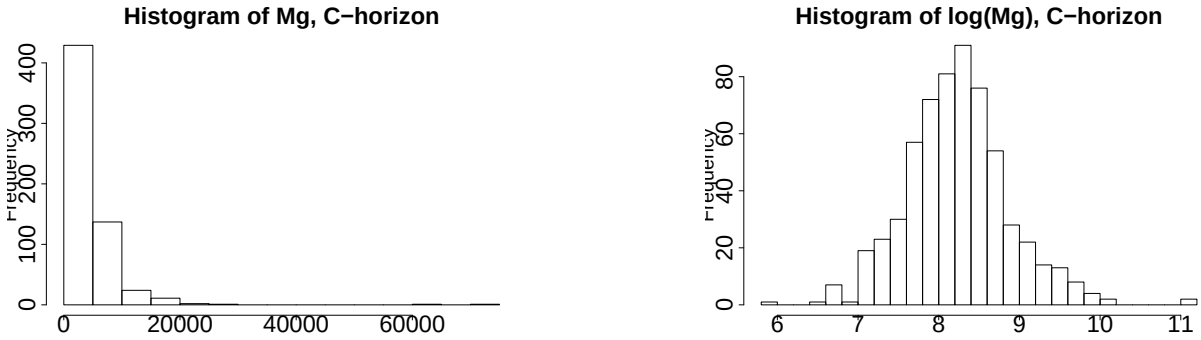


Figure 1.2: Log-transformation of Mg in C-horizon

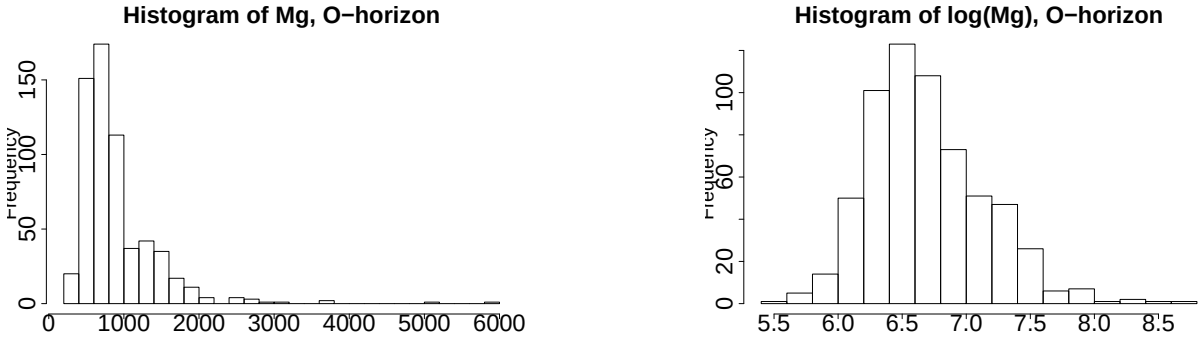


Figure 1.3: Log-transformation of Mg in O-horizon

1.7 Standardisation

It is possible that the variables show a striking difference in the amount of variability across a data set. If this is the case, then the range of the variables must be scaled suitably in order to guarantee a comparability.

In our data set from the C-horizon the values of Silicium lie at approximately 300.000 ppm and the ones of Arsenic are smaller than 0.1 ppm. Figure 1.6 represents the influence of the elements in each of the clusters. Here, influence means the value of the cluster centers, which are drawn in the vertical direction. Furthermore, the size of the clusters is also given in this figure ($\sum_{i=1}^{n_c} \text{size}_i = 1 \quad n_c \ldots$number of clusters). The influence of the element Si and a bit of the element Al is dominant in every cluster because of their high values. This is the reason why the cluster results are very similar to displays only the univariate map of Si (see Reimann et al., 1998) with a little influence of Al. Figure 1.7 has different influences of the elements in each of the clusters. It shows the influence of the variables in the clusters after the standardisation. There is no doubt that this data set must be scaled before cluster analysis can be applied.

The effect of different scaling is shown in Figure 1.8. There, the same data points are displayed in different units, like in ppm or mg/kg. Obviously, any cluster procedure would act like the human eye by once combining elements 1-2 and 3-4 (left) and once 1-3 and 2-4

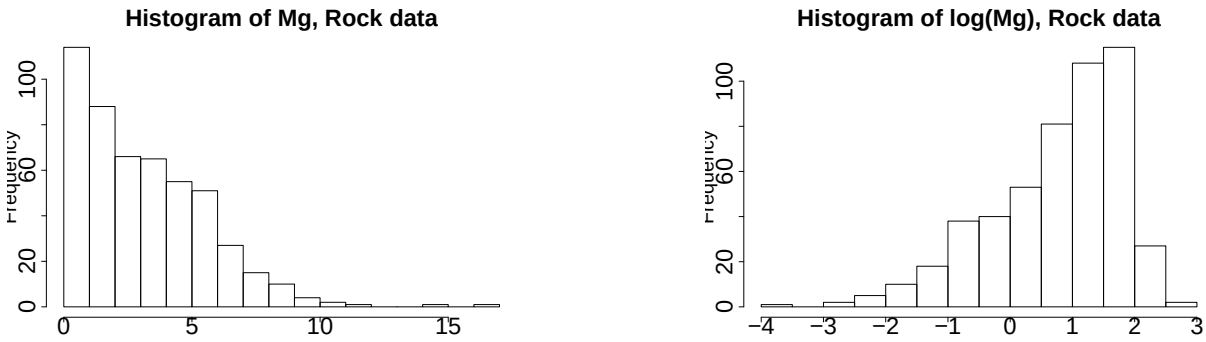


Figure 1.4: Log-transformation of Mg for the "Rock" data

(right).

On the other hand, many scientists (e.g. Everitt, 1974; Gordon, 1999) particularly emphasise that standardisation should not be made into a rule. Figure 1.9 shows data from two standard normal distributions with centres $(-2, 0)^\top$ and $(2, 0)^\top$, respectively. The clustering of the data in Figure 1.9 with a hierarchical clustering technique shows that without standardisation of the variables, the structure of the two groups has been recognised. With the standardisation of the data the clear structure of the data is lost (cf. Schöll, 2002).

Usually, in statistics we speak about standardisation when both centring and scaling are applied. The question of scaling the variables is actually, from a more general point of view, a problem of weighting. Sometimes, the researcher has reasons for assigning subjective weights to different variables. In general, however, this process is difficult to define in practise (see e.g. De Soete, 1986).

A more common procedure is to assign the weights in an objective manner. There are a lot of different approaches to scale (standardise) variables:

1.7.1 (0-1)-Transformation

A quite simple standardisation method is the *(0-1)-transformation* which is defined as

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij} - \min x_{.j}}{\max x_{.j} - \min x_{.j}} \tag{1.1}$$

where

$$x_{.j} = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

includes the values of the j-th variable. By this transformation each variable is mapped into the interval $[0, 1]$.

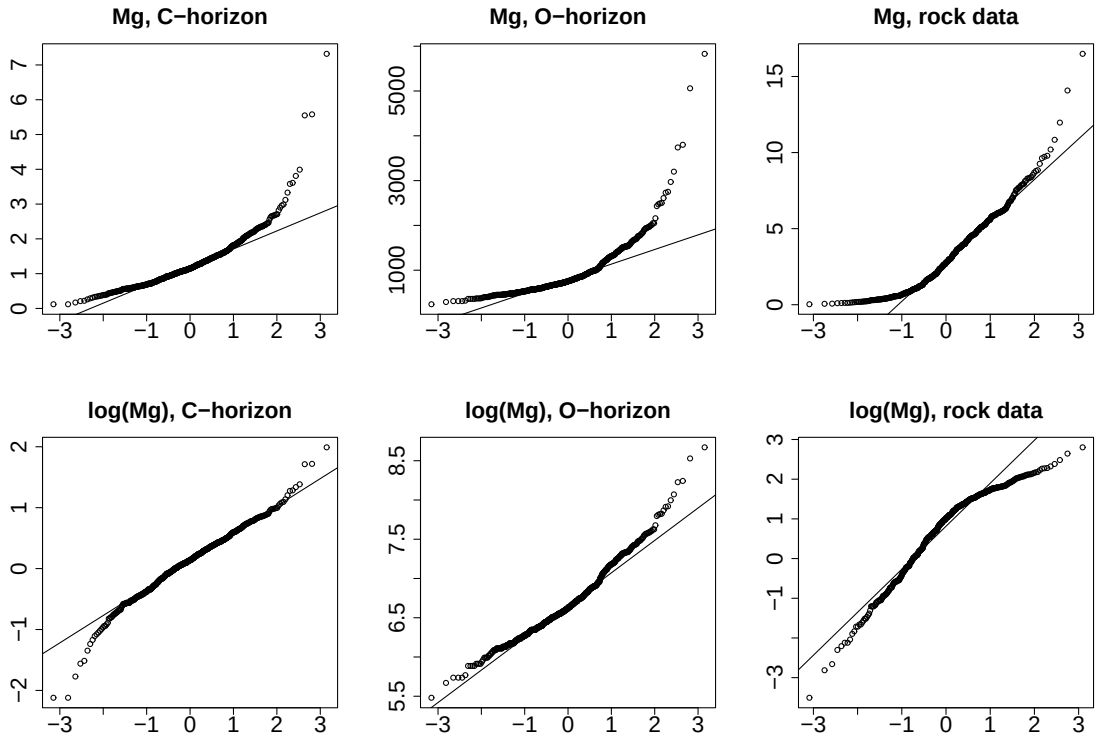


Figure 1.5: Normal QQ-plots of Mg and log(Mg)

1.7.2 Z-Transformation

The most commonly used and most recommended method of converting the original measurements into unitless variables with mean 0 and variance 1 is the *z-transformation*.
The standardised measurements are defined as

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij} - \overline{x}_j}{s_j} \tag{1.2}$$

where

$$\overline{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \tag{1.3}$$

is the mean value of variable j for each $j = 1, \ldots, p$ and s_j is the *standard deviation*

$$s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)^2} \ . \tag{1.4}$$

An alternative which is much less sensitive to outliers is to replace $\overline{x}_j$ by the median of the

j-th variable and s_j by the Median Absolute Deviation (MAD) which is defined by

$$\text{MAD } (x_{.j}) = 1.4826 \cdot \underset{i}{\text{med}} \, |x_{ij} - \underset{l}{\text{med}}(x_{lj})| \quad . \tag{1.5}$$

1.7.3 Other Standardisation Methods

Milligan (1996) suggests other four standardisation methods. The first suggestion for standardisation divides each value by the standard deviation

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij}}{s_j} \quad . \tag{1.6}$$

This method produces a set of transformed variables with variances of 1, but different means and ranges. In the second suggestion

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij}}{\max(X)} \tag{1.7}$$

each variable is mapped into the interval $\left[\frac{min(X)}{max(X)}, 1\right]$.
The standardisation

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij}}{\max(X) - \min(X)} \tag{1.8}$$

divides each value by the range of all values. With this transformation each variable is mapped into the interval $\left[0 + \frac{\min(X)}{\max(X) - min(X)}, 1 + \frac{\min(X)}{\max(X) - min(X)}\right]$. The last suggestion is dividing each value by the sum of all values

$$x_{ij} \quad \leftarrow \quad \frac{x_{ij}}{\sum_{i,j} x_{ij}} \quad . \tag{1.9}$$

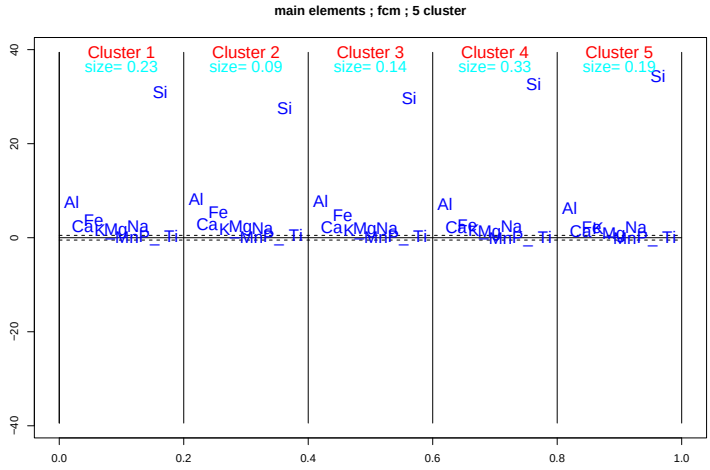


Figure 1.6: Cluster result: Influence of the main elements of the C-horizon in the clusters without standardisation.

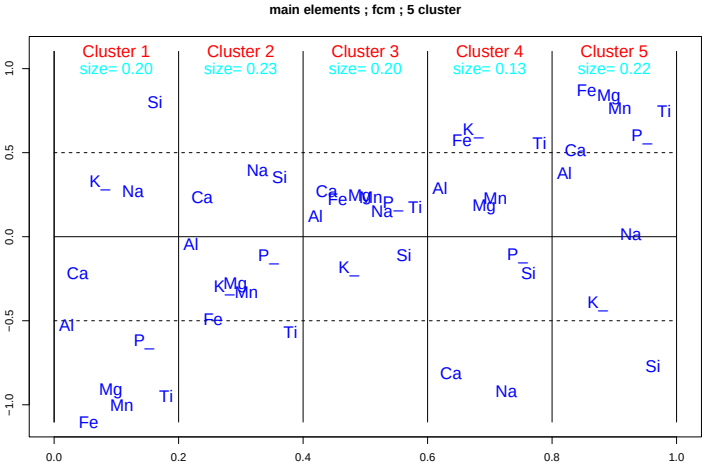


Figure 1.7: Cluster result: Influence of the main elements of the C-horizon in the clusters with standardisation.

Figure 1.8: Different scale level

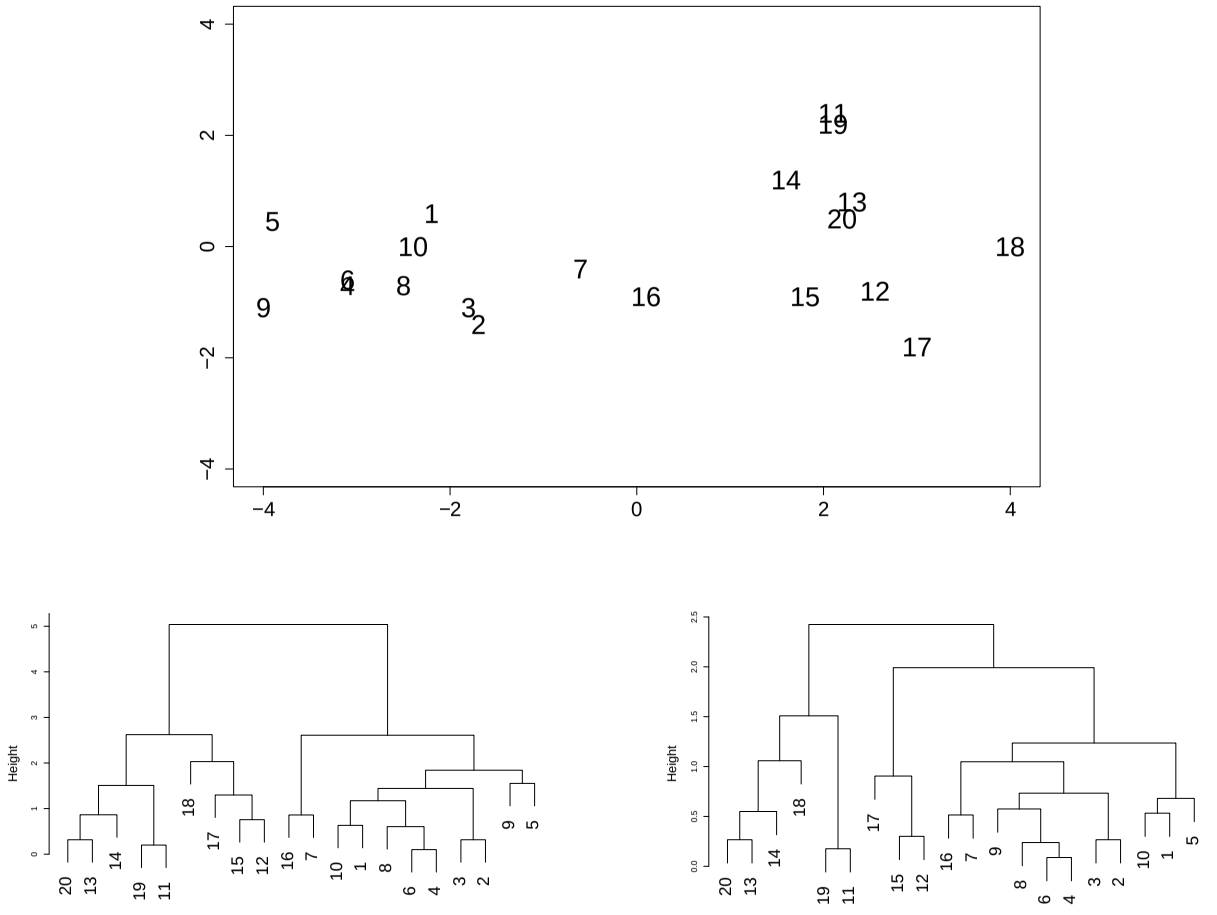


Figure 1.9: Standardisation problem. Dendrogram for the raw data and the scaled data

1.8 Distance Measures

The next step is to measure distances between the objects, in order to qualify their degree of dissimilarity.
Let us denote the i-th object by $x_i = (x_{i1}, \ldots, x_{ip})^T$ for $i = 1, \ldots, n$. The distance between the i-th and the j-th object, for example, can be measured with the *Euclidean distance*

$$d(i,j) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2} = \|x_i - x_j\|_2 \tag{1.10}$$

or the *Manhatten distance*

$$d(i,j) = \sum_{k=1}^{p}|x_{ik} - x_{jk}| = \|x_i - x_j\|_1 \tag{1.11}$$

A generalisation of both is the *Minkowski distance*

$$d(i,j) = \sqrt[q]{\sum_{k=1}^{p}(x_{ik} - x_{jk})^q} = \|x_i - x_j\|_q \tag{1.12}$$

where q is any real number larger than or equal to 1.

Other well-known distance measures are, for instance, the *Canberra metric* and the *Czekanowski coefficient* (see Johnson and Wichern, 1998, for details). Bandemer and Näther (1992) give an overview of 40 other measurement units for distance. A new approach for a distance measure is the *Voroni distance* (see in Höppner and Klawonn, 2001).

In cluster analysis it is quite common to speak about *dissimilarties* rather than distances. Basically, dissimilarities are nonnegative numbers $d(i,j)$ which are small (close to zero) when i and j are "near" to each other and large when i and j are very different.

The elements of the dissimilarity matrix satisfy certain minimum conditions:

1. $d(i,j) \geq 0$
2. $d(i,i) = 0$
3. $d(i,j) = d(j,i)$
4. $d(i,j) = d(i,h) + d(h,j)$

for all i, j and h.

Distances can in principle also be computed between the variables. This is useful if one is interested in clustering the variables rather than the objects. A usual way to define distances between variables is to calculate the *Pearson product moment correlation*

$$r(j,k) = \frac{\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik} - \overline{x}_k)^2}} \tag{1.13}$$

between the variables j and k. The coefficient does not depend on the choice of measurement units and lies between -1 and 1. If there is a strong relationship between two variables the coefficient is close to 1 or -1.
One possibility of converting this correlation coefficient to dissimilarities is to apply

$$d(j,k) = \frac{(1 - r(j,k))}{2} \quad . \tag{1.14}$$

With this formula, variables with a high positive correlation receive a dissimilarity coefficient close to zero, whereas variables with a strong negative correlation will be considered as very dissimilar. In other applications one might prefer to use

$$d(j,k) = 1 - |r(j,k))| \tag{1.15}$$

in which case also variables with a strong negative correlation will be assigned a small dissimilarity.

Chapter 2

Methods

Clustering can be loosely defined as the process of organising objects or variables into groups whose members are similar in some way. Cluster analysis was developed in the mid 70-ies, see e.g. Fasulo (1999). There is a number of methods of clustering data. The most important techniques are:

- Incomplete clustering method: e.g. reduction of the data dimension down to two or three dimensions by applying principal component analysis. Clusters can be detected by the viewer simply by having a look at the graphical output.
- Partitioning: every object is assigned to exactly one group.
- Overlapping cluster methods: an object can be assigned to more than one group.
- Hierarchical clustering methods: a tree of clusters is built where every cluster node contains child clusters.
- Fuzzy clustering methods: the objects have no clear allocation into a group. Memberships are evaluated to specify how far the object is located from a cluster.

In the literature many other methods of clustering are described. Below an overview of the most important methods which have not been mentioned so far and their algorithms are given. This other clustering methods are:

- Density-based clustering algorithms: DBSCAN (Ester et al., 1996), GDBSCAN (Sander et al., 1998), OPTICS (Ankerst et al., 1999), DBCLASD (Xu et al., 1998), DENCLUDE, DBCLASD, used to detect irregular shapes.
- Grid-based clustering algorithms: DENCLUE, CLIQUE, MAFIA (Goil et al., 1999), BANG (Schikuta and Erhart, 1997), GRIDCLUST (Schikuta, 1996), STING (Wang et al., 1997), WaveCluster (Sheikholeslami et al., 1998), FC (Barbara and Chen, 2000) , which inherit the topology from the underlying attribute space and can handle outliers.

- Co-occurrence algorithms of categorical data: ROCK (Guha et al., 2000), CURE (Guha et al., 1998), SNN, CACTUS (Ganti et al., 1999), HMETIS (Karypis et al., 1997), STIRR (Gibson et al., 1998), are concepts of a variable size transaction.
- Subspace clustering algorithms: CLIQUE (Aggarwal et al., 1998), ENCLUS (Cheng et al., 1999), OPTIGRID (Hinneburg and Keim, 1999), ORCLUS (Aggarwal and Yu, 2000), PROCLUS (Aggarwal et al., 1999), FLOC (Yang et al., 2002), which are techniques that are specially designed to work with high dimensional data.
- Co-clustering algorithms: OLAP, which turns attributes to numerical ones.
- Constraint-based clustering algorithms: frequently sensitive C-means, COD (Tung et al., 2001), which are modifications of the C-means algorithm.
- Neural networks and learning algorithms: CLTree, with which points are reassigned in order to correspond to the forecasting.
- Evolutionary algorithms: GGA, GCA (Lucasius et al., 1993), which modifies a set of C-means systems.
- Simulated annealing clustering algorithms: SINICC (Brown and Huntley, 1991), CLASA (Chu et al., 2001), which amounts a relocation of a point from its current cluster to a new randomly chosen one.
- Tabu search algorithms (Al-Sultan, 1995).
- Scalable clustering algorithms: DIGNET (Thomopoulos et al., 1995), BIRCH (Zhang et al., 1996), CLARANS (Ng and Han, 2002), which face problems of scalability in terms of computing time and memory space.
- Particular fuzzy clustering algorithms for pattern recognition (see e.g. Baraldi and Blonda, 1999).

Generally, there are some important characteristics which should be the basis for the choice of the clustering algorithm. Some of these characteristics are (see e.g. Berkhin, 2002)

- type of scale of the variables
- scalability to large data sets
- ability to work with high dimensional data
- ability to find clusters of irregular shape
- handling outliers
- time complexity
- data order dependency

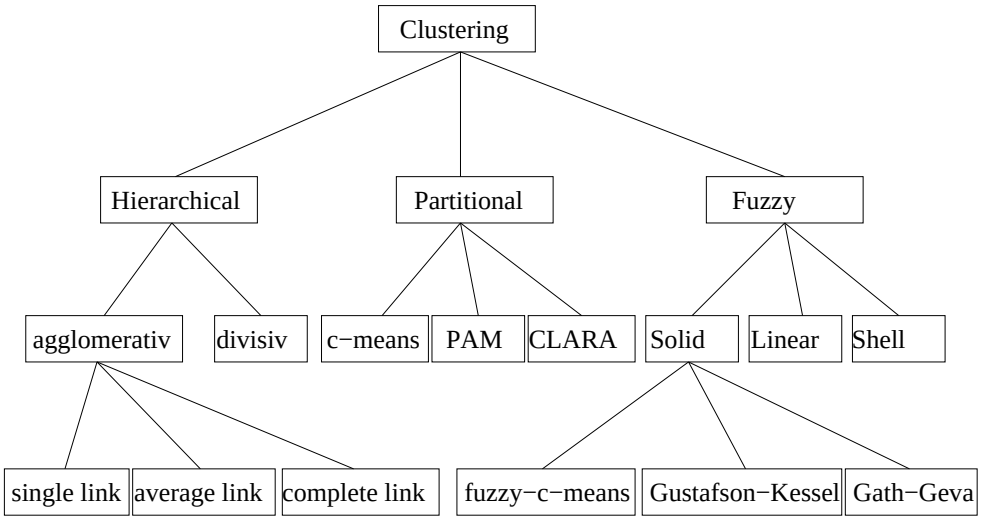


Figure 2.1: Cluster algorithms which are discussed in this section

- type of the clustering technique
- reliance on prior knowledge and user defined parameters
- interpretability of results.

At the beginning of this section we mentioned different clustering techniques. In the following the most widely used techniques will be explained in more detail. An overview over these methods which will be treated in the following is presented in Figure 2.

2.1 Hierarchical Clustering

The result of a hierarchical clustering procedure is composed of a sequence of clustering partitions. This sequence can visually be displayed by a clustering tree which is called *dendrogram*.

2.1.1 Representation of a Hierarchical Classification

A construct which is relevant to describe a hierarchy is an *n-tree*. An *n-tree* is a set $T = \{A, B, \ldots\}$ of subsets of the set of objects $\Omega = \{1, 2, \ldots, n\}$ satisfying the following conditions:

(i): $\{i\} \in T$ for all $i \in \Omega$

(ii): $\emptyset \notin T$ ($\emptyset$ denotes the empty set)

(iii): $\Omega \in T$

(iv): if $A, B \in T$ then $A \cap B \in \{\theta, A, B\}$

Condition (iv) ensures that the subsets are hierarchically-nested.

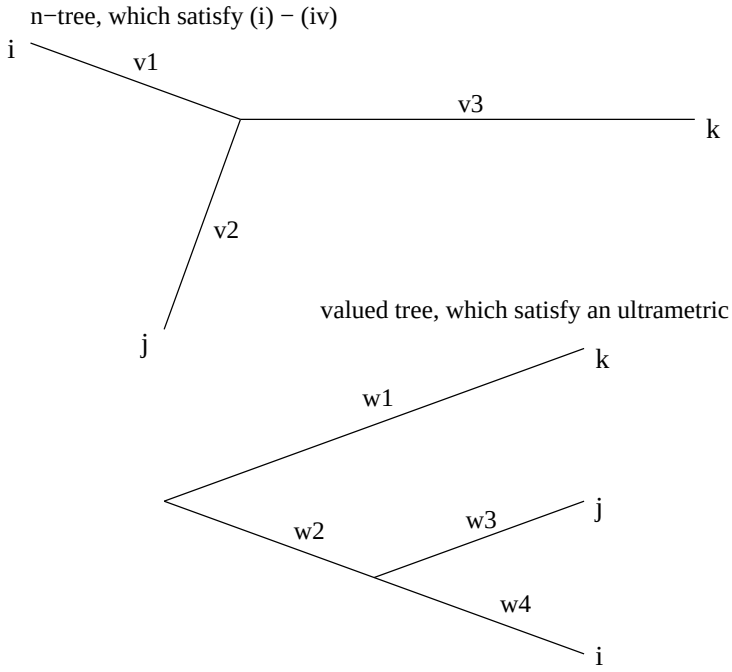


Figure 2.2: Display of a n-tree and a valued tree

In the upper part of Figure 2.2 an additive n-tree is shown. In order to correspond with a valued tree (as shown in the lower part of Figure 2.2) more conditions need to be formulated.

A *valued tree* associated with a height h in each hierarchical node, which satisfies the condition for nested subsets A and B always fulfills

$$h(A) \leq h(B) \Leftrightarrow A \subseteq B \quad .$$

For each pair of objects (i, j), h_{ij} $(= h_{ji})$ is defined to be the height of the smallest subset containing both the i-th and j-th object. The value of h_{ij} measures the difference between the i-th and the j-th object in the classification, with small values of h_{ij} indicating that the

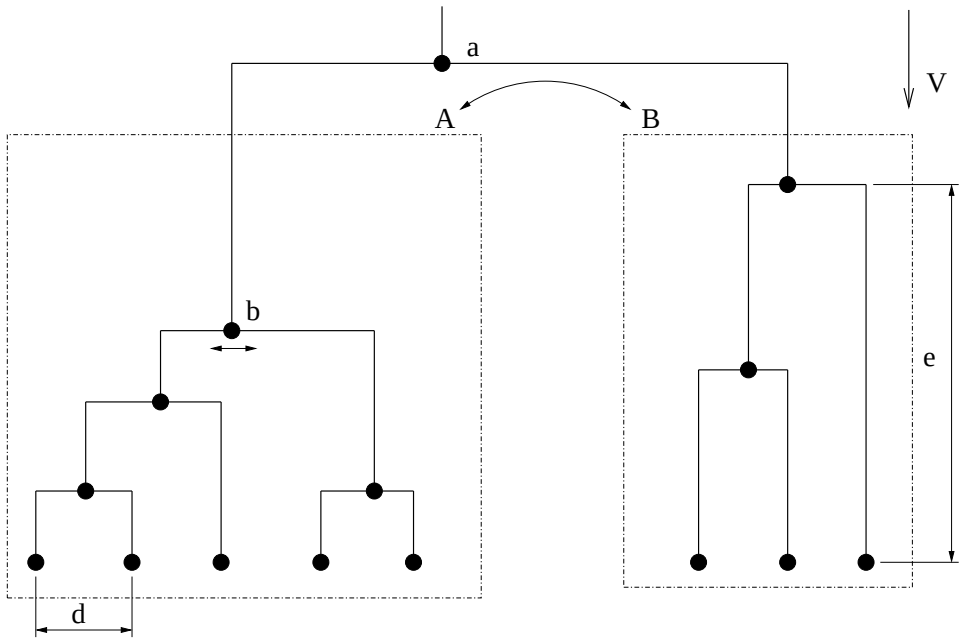


Figure 2.3: Representation of a dendrogram with nodes and preferred direction (divisive V or agglomerative) with space for following rearrangements: Permutation of Cluster A and B, horizontal displacement of b, distance d between trees and the height e of a tree.

corresponding objects are perceived as similar to one another. The set of the values h_{ij} for $i, j \in \Omega$ satisfies either the un-equation of an *ultra-metric* for a similarity measure

$$h_{ij} \geq \min(h_{ik}, h_{kj}) \text{ for all } i, j, k \in \Omega \tag{2.1}$$

and equivalent to the similarity measure, the un-equation for a dissimilarity measure

$$h_{ij} \leq \max(h_{ik}, h_{kj}) \text{ for all } i, j, k \in \Omega \quad . \tag{2.2}$$

So a valued ultra-metric tree is a tree where each node is equidistant from some specific node, called the *root* of the tree. One can examine easily that the tree in Figure 2.2 fulfils the conditions of an ultra-metric.

This presentation of a valued tree is called a dendrogram and it looks like Figure 2.3.

There are several different formats in which dendrograms are presented (see e.g. Gordon, 1996). A selection is shown in Figure 2.4. The most common formats are those shown in (a) and (b), or versions of them rotated by 90 degrees or 180 degrees.

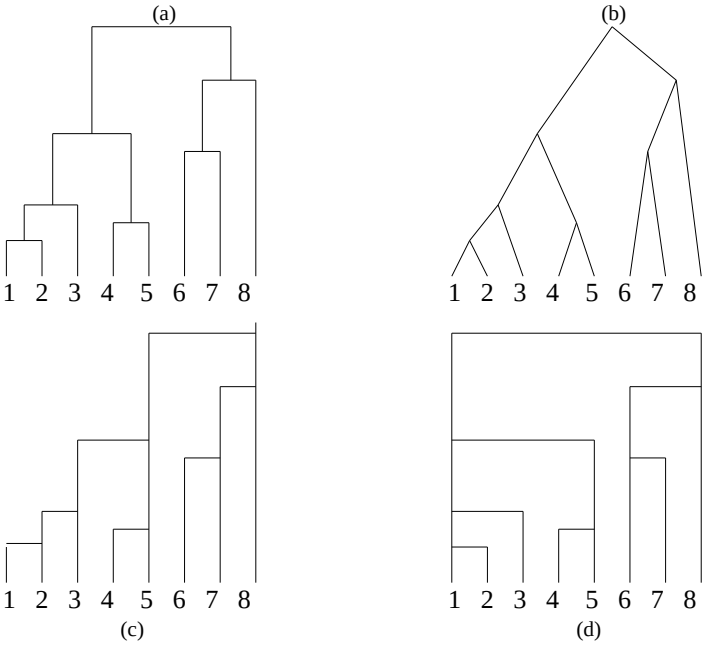


Figure 2.4: Four formats for representing the same hierarchical classification

2.1.2 Algorithms to Generate a Hierarchical Classification

There exists a variety of different algorithms for obtaining a hierarchical classification. Day (1996) proposed that all hierarchical classification algorithms, if carefully implemented, exhibit $O(n^2)$ time complexity. Due to space limitations we will restrict to some procedures which lead to a dendrogram. We will further restrict to *agglomerative techniques* which start with single object clusters and enlarge the cluster step by step. The reverse procedure which starts with one cluster containing all objects and splits the group(s) step by step, is called *divisive algorithm.*

2.1.3 Agglomerative Algorithms

At the beginning, each object forms an own class, leading to n different clusters. At each step of the algorithm, the number of clusters is reduced by one, where the most similar classes are combined. The "similarity" of the combined pair can be measured, and a "height" is associated with this newly-formed class. At the end of the process there is only one single cluster left.

If classes C_i and C_j are combined, a general scheme of evaluating the dissimilarity between $C_i \cup C_j$ and some other class C_k, given by Lance and Williams (1966), is defined as:

$$d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) + \gamma |d(C_i, C_k) - d(C_j, C_k)| \quad (2.3)$$

Table 2.1 presents the most commonly used parameters. The methods for three of these choices of the parameters will be discussed below in detail.

In the first step of the algorithm, the cluster C_i consists only of object i and cluster C_j only of object j. The distances $d(C_i, C_j)$ for single element classes can be selected according to Section 1.8.

Table 2.1: Hierarchical clustering strategies

Clustering criterion	α_i, α_j	β	γ
Single linkage	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete linkage	$\frac{1}{2}$	0	$\frac{1}{2}$
Average linkage	$\frac{1}{2}$	0	0
Centroid	$\frac{n_i}{n_i+n_j}$	$\frac{n_i n_j}{(n_i+n_j)^2}$	0
Ward	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{-n_k}{n_i+n_j+n_k}$	0

n_l is the number of objects in class C_l $(l = i, j, k)$

Single Linkage

In the first step, the two closest objects are combined. Hence, the combined objects generate a new group. Equation 2.3 is used with the coefficients for single linkage (Table 2.1), and this formula can be simplified to

$$d(C_i \cup C_j, C_k) = min\{d(C_i, C_k), d(C_j, C_k)\} \quad . \tag{2.4}$$

Single linkage dendrograms tend to be very unbalanced in the sense that big classes are quickly combined. This procedure tends to produce many small groups and few big groups. Single linkage is suitable to detect outliers.

Complete Linkage

Complete linkage clustering proceeds in much the same manner as single linkage clusterings, with one important exception: Not the smallest distances are considered but the biggest ones. So the equation looks like

$$d(C_i \cup C_j, C_k) = max\{d(C_i, C_k), d(C_j, C_k)\} \quad . \tag{2.5}$$

The complete linkage algorithm tends to produce a balanced dendrogram.

Average Linkage

The average linkage criteria were designed to take a middle road between single linkage and complete linkage and by far surpasses them in the sensitivity with respect to single objects. This method treats the distance between two clusters as the average distance between all pairs of items where one member of a pair belongs to each cluster. If $z_{(ij)}$ denotes an object from $C_i \cup C_j$, and z_k an object from C_k, than the average distance can be expressed as

$$d(C_i \cup C_j, C_k) = \frac{\sum_{(ij)} \sum_k d(z_{(ij)}, z_k)}{n_{(ij)} n_k} \tag{2.6}$$

where $n_{(ij)}$ is the number of objects in $C_i \cup C_j$.

2.2 Partitioning

First a number of clusters n_c has to be determined into which the data-set is to be divided and which satisfies the requirements of a partition:

Each group must contain at least one object.
Each object must belong to exactly one group.

Choosing an appropriate number n_c is often impossible and therefore partitioning is done with different choices of n_c and the "best" result is taken.

2.2.1 C-means Method

Given n objects, characterised by p variables, we like to partition them into n_c clusters $\{C_1, C_2, \ldots, C_{nk}\}$ such that cluster C_k has $n_{(k)}$ members and each object is in one cluster. The mean vector (centre, prototype), v_i, of a cluster C_i is defined as the centroid of the cluster

$$v_i = \frac{1}{n_{(i)}} \sum_{l=1}^{n_{(i)}} x_l^{(i)} \tag{2.7}$$

where $n_{(i)}$ is the number of objects in C_i and $x_l^{(i)}$ is the l-th object belonging to cluster C_i.

Here, we also need to determine the number of clusters of the output partition. Starting from a given initial locations of the n_c cluster centroids, the algorithm uses the data points to iteratively relocate the centroids and reallocate points to the closest centroid. The process is composed of these steps:

1. Select an initial partition with n_c clusters.
2. Assign each object to the closest cluster centre.

3. Recompute the cluster centres using the current cluster memberships.

4. Go to step 2 until the cluster memberships and thus cluster centroids do not change beyond a specified bound.

C-means clustering optimises the objective function

$$J(X, V, U) = \sum_{i=1}^{n_c} \sum_{j=1}^{n} u_{ij} d^2(x_j, v_i) \tag{2.8}$$

where $X = \{x_1, \ldots, x_n\}$ ... data set
$V = \{v_1, \ldots, v_{n_c}\}$ ... cluster centres (prototypes)
$U = [(u_{ij})]$ ... matrix with the membership coefficients u_{ij} for objects x_j to a cluster C_i
d ... Euclidean distance between the objects and the cluster centres.
n ... number of objects
n_c ... number of clusters

The C-means algorithm can be implemented as follows. Fix $n_c, 2 \leq n_c < n$, and choose the termination tolerance $\delta > 0$, e.g., between 0.01 and 0.001. Initialise $U^{(0)}$ (e.g. randomly).

REPEAT for $r = 1, 2, \ldots$

1. Calculate the centres v_i of the clusters

$$v_i^{(r)} = \frac{\sum_{j=1}^{n} u_{ij}^{(r-1)} \cdot x_j}{\sum_{j=1}^{n} u_{ij}^{(r-1)}}, \quad 1 \leq i \leq n_c \tag{2.9}$$

2. Update $U^{(r)}$: Reallocate cluster memberships

$$u_{ij}^{(r)} = \begin{cases} 1 & \text{if } d(x_j, v_i^{(r)}) = \min_{1 \leq l \leq n_c} d(x_j, v_l^{(r)}) \\ 0 & \text{otherwise} \end{cases}$$

UNTIL $\|U^{(r)} - U^{(r-1)}\| < \delta$.

2.2.2 CLARA (Clustering Large Applications)

The program CLARA was developed by Kaufman and Rousseeuw (1990) for clustering large data sets.
The C-means algorithm attempts to minimise the average squared distance, yielding so-called centroids. A more robust method is to minimise the average distance, which is called the medoid of it's cluster. This method of partitioning around medoids (PAM) is called the C-medoid technique (Kaufman and Rousseeuw, 1990).
The difference between the PAM and CLARA algorithms is that the latter one is based upon sampling. Only a small portion of the real data is chosen as a representative of the data and medoids are chosen from this sample using PAM. The idea is that if the sample is selected

in a fairly random manner, then it correctly represents the whole data set and therefore, the representative objects (medoids) chosen, will be similar as if chosen from the whole data set. Having drawn and clustered five samples, the one for which the lowest average distance was obtained is selected.

2.3 Fuzzy Clustering

In the process of probabilistic fuzzy-clustering the data is assigned to the clusters with a probability between 0 and 1. A membership of 0.7, however, does not mean that the object was assigned to the cluster with a probability of 70%. Instead, the membership degree is to be interpreted in the sense of *fuzzy-logic*.
A *fuzzy set* is defined by the function $\mu : U \rightarrow [0, 1]$. A value of $\mu(u_1) = 1, u_1 \in U$, means that the element u_1 is fully in accordance with the concept described by the fuzzy set, while a value $\mu(u_2) = 0, u_2 \in U$, means that the element u_2 is not in accordance with the concept described by the fuzzy set.

The aim of cluster analysis is to assign objects to classes. However, there are many applications in which the assignment of an object to only one cluster has not much use. This problem becomes clear in the data set presented in Figure 2.5. The object right in the middle can be assigned neither to the left nor the right cluster. If this object was clearly assigned to a cluster the information that the two clusters are symmetric would be lost. Another disadvantage of a clear assignment is that the information how typical the object is represented by the clusters would also be lost. Sometimes this information, however, may be of some interest.
One possibility to model fluent transitions between clusters and to consider them in the cluster analysis is the application of gradual memberships. To every object x_j a membership degree $u_{ij} \in [0, 1]$ is assigned for every cluster C_i. A membership degree of 1 means that the object fully belongs to a cluster. However, a membership degree of 0 means that the object does not at all belong a cluster.
The object lying in the middle of Figure 2.5 can be assigned to both of the clusters shown. Thus the membership degree should be equal for every of these two clusters.

In fuzzy cluster analysis this concept of describing memberships is applied. The values of the memberships can be visualised by various shades of grey. This assignment of the data in memberships and its visualisation corresponds to human intuition.

The idea of fuzzy clustering originated in the hard C-means algorithm founded by Ruspini (1969), and the fuzzy C-means algorithm was developed by Dunn (1974). The most commonly used fuzzy clustering algorithm is the fuzzy C-means algorithm of Bezdek (1981).

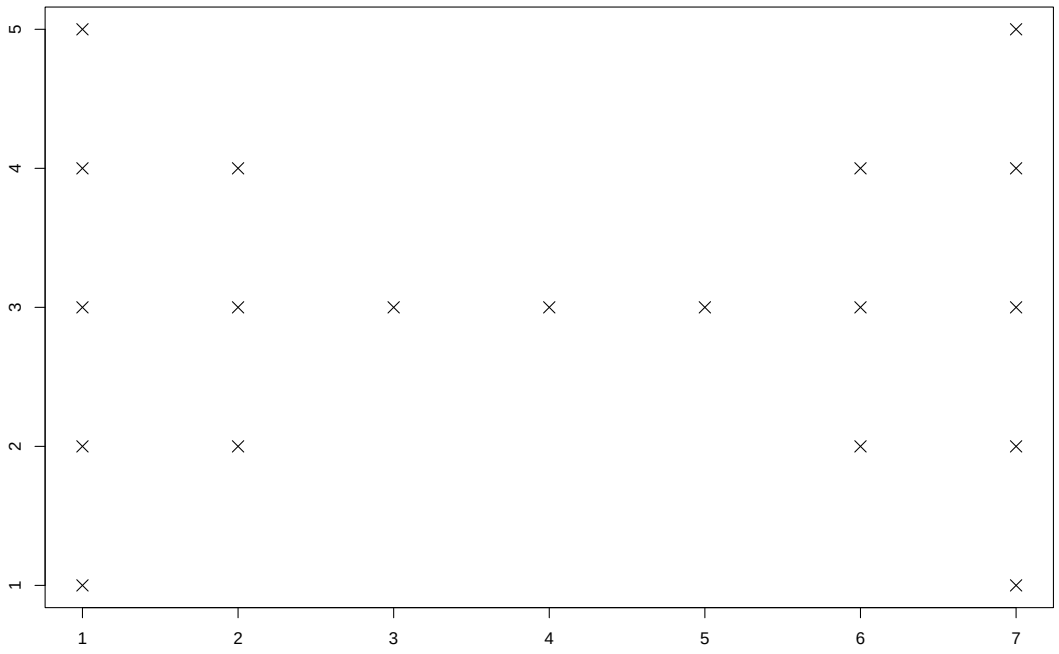


Figure 2.5: Butterfly data set with two clusters

2.3.1 Objective Functions

The *objective function* used by Dunn (1974) is defined as

$$J(X, U, V) = \sum_{i=1}^{n_c} \sum_{j=1}^{n} u_{ij} \cdot d^2(v_i, x_j) \quad . \tag{2.10}$$

Bezdek (1981) generalised this fuzzy objective function by introducing the weighting exponent m, which is called the fuzzifier;

$$J(X, U, V) = \sum_{i=1}^{n_c} \sum_{j=1}^{n} u_{ij}^m \cdot d^2(v_i, x_j) \quad . \tag{2.11}$$

A common choice for the fuzzifier is $m = 2$. If the value m increases, the memberships get smaller.

A robust objective function is based on modifying the objective function (2.10) and looks like

$$RJ(X, U, V) = \sum_{i=1}^{n_c} \sum_{j=1}^{n} u_{ij} \cdot d^2(v_i, x_j) + \eta \sum_{j=1}^{n_c} \sum_{i=1}^{n} (1 - u_{ij}) d^2(v_i, x_j) \ , \ 0 < \eta < 1 \quad . \tag{2.12}$$

If u_{ij} is close to 1, then the last term is zero. The smaller u_{ij} becomes, the bigger is the influence of the last term on the equation (2.12) which is to be minimised. Hence, low values of u_{ij} are "punished". For this reason this equation is not so sensitive in terms of outliers.

Tai-Ning and Sheng-De (2002) propose another robust objective function:

$$RFJ(X,U,V) = \sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij} \cdot d^2(v_i, x_j) + \sum_{j=1}^{n_c}\sum_{i=1}^{n} f(u_{ij})^m \eta_i \tag{2.13}$$

where $f(u_{ij})$ is the fuzzy complement of u_{ij}, which may be interpreted as the degree to which x_j does not belong to the j-th data cluster and η_i is a constant. One choice of $f(u_{ij})^m$ is e.g. $m = 1$ and

$$f(u_{ij}) = 1 + u_{ij} \cdot \log(u_{ij}) - u_{ij} \tag{2.14}$$

2.3.2 Fuzzy Cluster Algorithm

Fuzzy analysis works without a clear allocation of the data to classes or clusters. Instead it defines for each object degrees of memberships to its clusters. There is a big number of different fuzzy cluster algorithms. Basically these algorithms are divided in three classes. *Solid fuzzy cluster algorithms* are methods for recognising full clusters. The most commonly used are the fuzzy C-means algorithm, the Gustafson-Kessel algorithm and the Gath-Geva algorithm. *Linear fuzzy cluster algorithms* are recently developed methods for recognising lines. The most famous one is the *fuzzy C-variates algorithm* (FCV) proposed by Bezdek (1981). *Shell fuzzy cluster algorithms* are also recently developed methods for recognising circular, elliptical and parabolic shapes (see e.g. in Höppner et al., 1996).

2.3.3 Fuzzy C-means Algorithm

The input to the FCM (fuzzy C-means) algorithm consists of the data objects. The FCM algorithm calculates the prototypes of the clusters and membership degrees for each object to the clusters.

The FCM algorithm can detect spherical (ball-shaped) clusters which are of nearly the same size. Each cluster is represented by its prototype. The FCM algorithm minimises the objective function (2.10) under the restrictions

$$\sum_{i=1}^{n_c} u_{ij} = 1 \quad \forall j \in \{1,\ldots,n\} \tag{2.15}$$

$$u_{ij} \geq 0 \quad \forall i \in \{1,\ldots,n_c\},\ \forall j \in \{1,\ldots,n\} \tag{2.16}$$

The updated prototypes are given by

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \quad . \tag{2.17}$$

By minimising the objective function (2.10) with respect to the restrictions (2.15) and (2.16), the calculation of the memberships looks like (Bezdek, 1981):

$$u_{ij} = \begin{cases} \frac{1}{\sum_{k=1}^{n_c} \left(\frac{d^2(v_i, x_j)}{d^2(v_k, x_j)} \right)^{\frac{1}{m-1}}}, & \text{if } I_j = \emptyset, \\ 0, & \text{if } I_j \neq \emptyset \text{ and } i \notin I_j, \\ x, x \in [0,1], \text{so that} \sum_{i \in I_j} u_{ij} = 1, & \text{if } I_j \neq \emptyset \text{ and } i \in I_j, \end{cases} \tag{2.18}$$

where $I_j = \{i | 1 \leq i \leq n_c, d^2(v_i, x_j) = 0\}$.
If $I_j = 0$ then x_j is not a prototype and the value of u_{ij} must be newly calculated. If $I_j \neq \emptyset$ and $i \in I_j$ then the object x_j is a prototype. (2.18) show that the calculation of the memberships is based only of the distance of the data to the prototypes.

Procedure of the FCM algorithm

- Given a data set $X = \{x_1, x_2, \ldots, x_n\}$
- Choose the cluster size $n_c, 2 \leq n_c < n$. Each cluster is defined by its prototype v_i. Set the iteration number to 0.
- Choose $m > 1, m \in \mathbb{R}$
- Choose a threshold for convergence and a maximum number of iterations.
- Initialise the cluster prototypes and the memberships.
- REPEAT
  - Increase the number of iterations by 1
  - Calculate the cluster prototypes $v_i, i \in \{1, \ldots, n_c\}$ due to (2.17).
  - Calculate the memberships u_{ij} due to (2.18).
- UNTIL the change of the cluster prototypes or the change of the membership coefficients is smaller than the threshold for convergence or the maximum number of iterations.

2.3.4 Gustafson-Kessel Algorithm

The Gustafson-Kessel (GK) algorithm (Gustafson and Kessel, 1979) is able to detect elliptical shaped clusters. This algorithm replaces the (squared) Euclidean distance by the (squared) Mahalanobis distance

$$d^2(v_j, x_i) = (x_j - v_i)^T A_i (x_j - v_i) \quad . \tag{2.19}$$

The prototype centres are defined by (2.17) and A_i is a norm matrix given by

$$A_i = \sqrt[p]{det(S_i)} S_i^{-1} \tag{2.20}$$

where S_i is the fuzzy covariance matrix

$$S_i = \frac{\sum_{j=1}^{n} u_{ij}^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.21}$$

and p is the number of variables.

The centre v_i describes the position of the cluster and the covariance matrix S_i describes the shape of the cluster.
The GK algorithm is computationally more expensive than the FCM algorithm, because the covariance matrix of each cluster must be inverted.

2.3.5 Gath-Geva Algorithm

The algorithm of Gath and Geva (1989), also called the Gaussian mixture decomposition algorithm, is an extension of the Gustafson-Kessel algorithm, which also deals with size and density of the cluster. The Gath-Geva (GG) algorithm assumes that the features come from a mixture of n_c Gaussian normal distributions.
The a priori probability of a feature belonging to a normal distribution (cluster) can be interpreted as an additional cluster size parameter.
We introduce the prior probability P_i for the belongingness of a object to the cluster (default value is $\frac{1}{n_c}$).
The algorithm uses the objective function (2.11) together with the distance

$$d^2(v_j, x_i) = \frac{\sqrt{det(S_i)}}{P_i} \exp\left(\frac{1}{2}(x_j - v_i)^T A_i (x_j - v_i)\right) \quad . \tag{2.22}$$

The prototype centres are defined by the membership-weighted mean of feature vectors (2.17), the norm matrices by

$$A_i = S_i^{-1} \tag{2.23}$$

where S_i is the fuzzy covariance matrix (2.21). The prior probability is given by

$$P_i = \frac{\sum_{j=1}^{n} u_{ij}^m}{\sum_{j=1}^{n} \sum_{t=1}^{n_c} u_{tj}^m} \quad . \tag{2.24}$$

The GG program of Höppner et al. (1996) takes the objects x_j as an input and yields the prototype size, a priori probability P_i, prototype centres v_i, covariance matrix S_i, norm matrix, squared distances and memberships. The fuzzy hyper-volume is used to set the global validity measure.

Due to the exponential function in (2.22) the distance value increases very quickly. Often, this causes numerical overflows and the GG algorithm returns NaN (not a number) values, especially in the presence of outliers.

2.4 Initialisation

Cluster algorithms which use an iterative procedure to find a (locally) optimal solution are sensitive to initial starting values. Either one initialises cluster centres directly or the membership matrix.

2.4.1 Centre Initialisation

A very appealing method to generate medoids (or centroids) comes from the mathematical area of combinatorial optimisation and is called simulated annealing (SA). It is a powerful tool for solving scheduling problems (search for a local optimum) like the travelling salesman problem. Originally, SA was developed in the steel industry and was adapted for the area of combinatorial mathematics by Johnson et al. (1989). This algorithm initialises the medoids (see e.g. Chu et al., 2002), and it works as follows:

1. Select randomly n_c (number of clusters) medoids which form an initial state S.
2. Choose an initial start temperature $T = T_0$ (see e.g. Johnson et al., 1989).
3. Randomly choose another state S' and calculate the difference $\Delta d = d(S') - d(S)$. Here d measures the average distance between all objects and the medoids. If $\Delta d < 0$ replace the state S by S'. Otherwise replace S by S' if $e^{\frac{-\Delta d}{T}} > \gamma$, where γ is a random value generated uniformly on the interval $[0, 1]$, and go to step 4.
4. If S has not been changed for a larger number of iterations, or if a fixed number of permutations is reached, go to step 5; otherwise go to step 3.
5. Terminate the program and return the selected medoids if the temperature T is below some predefined freezing temperature T_f or the total number of permutations is reached; otherwise lower the temperature T and go to step 3.

Another approach can be done by using genetic algorithms (see e.g. in Lee and Antonsson, 2000). The idea is as follows: Create individuals initially and evaluate these individuals. Create child derivatives with crossover (e.g. with partially matched crossover) and evaluate these new individuals. Transform these individuals with a "fitness function" and select a pair of individuals with the help of e.g. the stochastic universal sampling method or the roulette wheel method (see e.g. Whitley, 1994). Now, restart the algorithm and create new

individuals from the selected individuals with crossover until a maximum number of "generations" is reached.

These two methods were not only developed for initialising, but are also used as cluster algorithms (algorithm: CLASA for SA, Chu et al. (2001), and e.g. GCA, Lucasius et al. (1993), for genetic cluster algorithms).

In the fuzzy clustering program of Höppner (2000) there are three different ways to initialise the clusters:

- Use n_c points uniformly distributed along the "diagonal" of the space containing X (default).
- Use the first distinct points in the data set X.
- Use n_c points randomly drawn from the space containing X.

Other approaches can be found in Bradley and Fayyad (1998).

2.4.2 Membership Initialisation

The default initialisation of the fuzzy clustering program of Höppner (2000) assigns consecutive objects to the same cluster. For one-dimensional, ordered data it makes sense for some clustering algorithms to assume that the first $\frac{n}{n_c}$ objects belong to the first cluster, the second $\frac{n}{n_c}$ features to the second, etc.

Chapter 3

Data

3.1 Geology of the Kola Data

The Central Kola Expedition (CKE) in Russia, the Geological Survey of Finland (GTK) and the Geological Survey of Norway (NGU) cooperated from 1991 to 1998 to produce an ecogeochemical atlas (Reimann et al., 1998) of the central Barents Region, an area of 188.000 km^2.

Figure 3.1 shows a map at the investigated project area and the first project area (pilot project) in the region around Kirkenes and Zapoljarnij on an area of more than 12.000 km^2.

An overview of this project can be found on the web site *http://www.ngu.no/Kola*. The average sampling density is 1/300 km^2 (1/100 km^2 near the pollution sources and $1/400-500$ km^2 in background areas, see Figure 3.2) and the samples collected at each location consist of terrestrial moss, humus, topsoil and podzol profiles. The typical podzol profile consists of 5 main layers, the O-, E-, B-, BC, C-horizon.

In this master's thesis the C-horizon and O-horizon of the podzol samples are examined. The chemical composition of the O-horizon is represented by the humus sample (0-3 cm under the surface). The O-layer is usually brown in colour, loose and rich in plant roots and can accumulate large amounts of airborne heavy-metal input; it acts as a natural trap for heavy metals. In this layer we expect to find many deposits caused by the heavy metal industry marked in Figure 3.3. Mining and the mineral industry, in general, are the major fields of industrial activity. The area around Nikel, Zapoljarnij and Monchegorsk is one of the regions most polluted by SO_2 and heavy metals. Unfortunately, the local population is economically dependent on the heavy metal industry. So the ore and nickel industries are not expected to be shut down. Especially Russia does not - in opposition to Norway - obtain most of its energy from hydroelectric power plants. Therefore it needs this mineral industry.

Austria has an emission of SO_2 of 57 tones per year (according to the environment data bank of the Austrian Federal Economic Chamber) while the emission of SO^2 in Nikel (see Table 3.1; data from Reimann et al., 1998) is more than twice as high. Even the Netherlands have a lower emission of SO_2 than Nikel.

Figure 3.1: Project area of the Kola project with the major towns and the industrial centres.

Geology has the biggest influence on the C-horizon which is the deepest horizon (approx. 30 cm under the surface) and therefore least spoilt by industrial pollution. For this reason the clusters should display clear regional distinctions, and similarities with the lithological map, which is illustrated in Figure 3.4. Further explanations on the lithological map can be found in Reimann and Filzmoser (1999) and Reimann et al. (1998).

C-horizon samples which were taken at 605 sites and subquently analysed by a number of different techniques for more than 50 elements, displayed a result of 89 variables. For cluster analysis it is not advisable to use all variables. A better strategy is to reduce the dimensionality by selecting elements (see Section 1.4).

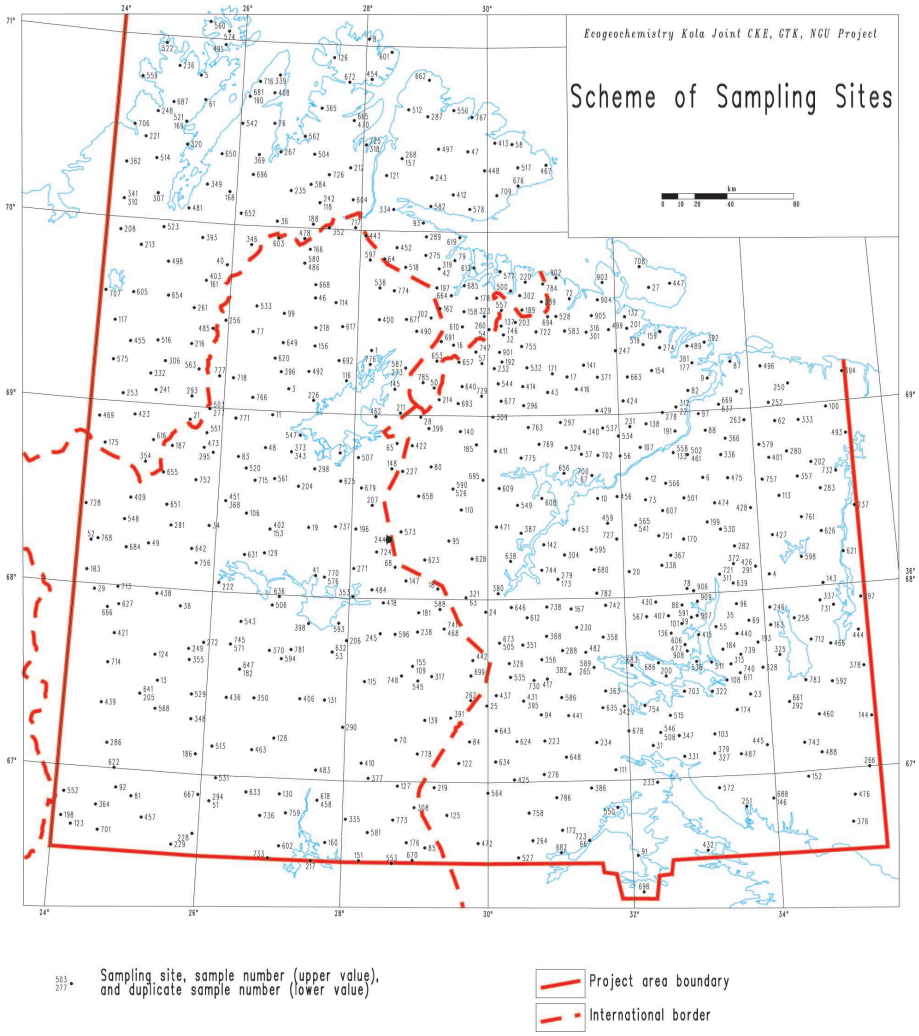


Figure 3.2: Sample sites and numbers for the regional mapping part

3.2 Data

A clustering of the data without a prior selection of variables can lead to results which are difficult to interpret and may serve to mask or hide real clustering in data (see Section 1.4). For this reason special care was given to the selection of the elements for clustering.

The selection of the elements of the O-horizon for clustering was carried out in two steps. For the first step it was crucial to know that the elements As, Co, Cu, Ni, and V are mainly responsible for the pollution and that the elements Ca, Fe, Mg, Na and Sr are good indicators for the sea spray (see the univariate maps of the elements in Reimann et al., 1998 and the geochemical characteristics of the elements in Reimann et al., 1998 or Reimann, 1998). In the second step elements were experimentally exchanged, added or ignored in order to reach a better structure of the clusters. The elements for the clustering of the C-horizon are predominantly the main elements, the trace elements and the mafic rocks. Some experiments were also carried out by mixing the main elements with the trace elements

Table 3.1: Emissions (in tonnes per year) of the major pollutants

Location	SO_2	Co	Cu	Ni	V_2O_2
Monchegorsk	97.715	81.5	934	1619	60
Nikel	129.160	5.2	82	136	13
Zapoljarnij	69.208	5.4	81	161	21
SUM	296.083	92.1	1097	1916	94

(mixed elements). Also further selections of elements were taken (see Section 5.2 and Figure 5.12). Altogether eight different groups of elements were thoroughly examined.

3.2.1 C-horizon

The eight characteristic groups are:

1. Main elements extracted from the XRF (X-ray fluorescence) method (Al, Ca, Fe, K, Mg, Mn, Na, P, Si, Ti)
2. Main elements without Si and Ti
3. A selection of the main elements (Al, Ca, K, Mn, Si, Ti)
4. Trace elements (As, Ba, Co, Cr, Cu, La, Ni, Pb, Rb, Sc, Sr, Th, V, Y, Zn)
5. Trace elements without As
6. A selection of the trace elements (As, Ba, Co, Cr, Cu, La, Pb, Rb, Th, Zn)
7. Mixed data (As, Ba, Ca, Co, Cr, Cu, Eu_INAA (INAA: analysed by instrumental neutron activation analysis), Fe, K,La, Mg, Mn, Na, Ni, P, Pb, Rb, Sc, Sr, Th, V, Y, Zn)
8. Mafic rocks (Co, Cr, Cu, Fe_INAA, Mg, Mn, Ni, Sc, Ti, V)

Tables 3.2 - 3.5 show the statistical summary of the examined groups. The remaining groups which are not shown in tables in an explicit way are part-groups of the four groups shown in Tables 3.2 - 3.5.

Table 3.2: Group1: Statistical summary[1] of the main elements from C-horizon

Element	Unit	DL	%<DL	min	max	mean	med	stddev	mad
Al_XRF[2]	wt.-%	0.03	0	2.92	12.08	7.34	7.38	0.969	0.667
Ca_XRF	wt.%	0.005	0	0.03	6.76	2.133	2.17	0.899	0.801
Fe_XRF	wt.-%	0.02	0	0.59	12.35	3.605	3.43	1.4	1.342
K_XRF	wt.-%	0.004	0	0.36	5.24	1.558	1.41	0.593	0.482
Mg_XRF	wt.-%	0.02	0	0.12	7.32	1.271	1.15	0.677	0.526
Mn_XRF	wt.-%	0.008	0	0.015	0.356	0.059	0.054	0.031	0.022
Na_XRF	wt.-%	0.01	0	0.08	4.87	2.26	2.45	0.678	0.504
P_XRF	wt.-%	0.004	0	0.004	0.589	0.045	0.039	0.032	0.019
Si_XRF	wt.-%	0.23	0	17.05	40.27	31.461	31.74	2.579	2.216
Ti_XRF	wt.-%	0.003	0	0.053	1.9	0.362	0.347	0.16	0.151

[1]DL: detection limit, < DL: percentage of samples below the detection limit, minimum, maximum, mean, median, standard deviation and median absolute deviation.
[2]_XRF: analysed by X-ray fluorescence.

Table 3.3: Group4: Statistical summary of the trace elements from C-horizon

Element	Unit	DL	%<DL	min	max	mean	med	stddev	mad
As	mg/kg	0.1	1.7	<0.1	30.7	1.25	0.5	2.349	0.445
Ba	mg/kg	0.5	0	4.7	1300	60.15	43.5	74.33	28.91
Co	mg/kg	0.2	0	1.2	44.3	8.22	7	5.029	3.706
Cr	mg/kg	0.5	0	2.2	471	36.16	28.35	35.09	16.23
Cu	mg/kg	0.5	0	2	149	21.96	16.2	18.44	10.82
La	mg/kg	0.5	0	3.5	203	17.94	12.8	20.96	6.449
Ni	mg/kg	1	0	1.2	228	23.41	18.65	21.09	11.56
Pb	mg/kg	0.2	0	0.3	45.3	2.748	1.6	3.326	0.741
Rb	mg/kg	15	6.3	<15	270	60	54	33.55	26.68
Sc	mg/kg	0.1	0.2	<0.1	15.4	2.816	2.3	1.809	1.186
Sr	mg/kg	0.5	0	1.6	1040	25.34	7.7	98.23	3.781
Th_INAA[1]	mg/kg	0.2	0	1	54	7.164	5.8	4.953	3.41
V	mg/kg	0.5	0	4.5	183	34.99	30.9	19.65	15.72
Y	mg/kg	0.5	0	0.9	169	6.366	4.4	10.97	2.372
Zn	mg/kg	0.5	0	3.7	348	27.40	20.9	24.17	12.45

[1]INAA: analysed by instrumental neutron activation analysis

Table 3.4: Group7: Statistical summary of the mixed elements from C-horizon

Element	Unit	DL	%<DL	min	max	mean	med	stddev	mad
As	mg/kg	0.1	1.7	<0.1	30.7	1.25	0.5	2.349	0.445
Ba	mg/kg	0.5	0	4.7	1300	60.15	43.5	74.33	28.91
Ca	mg/kg	3	0	110	41700	2279	1905	2383	1075
Co	mg/kg	0.2	0	1.2	44.3	8.22	7	5.029	3.706
Cr	mg/kg	0.5	0	2.2	471	36.16	28.35	35.09	16.23
Cu	mg/kg	0.5	0	2	149	21.96	16.2	18.44	10.82
Eu_INAA	mg/kg	0.2	0	0.3	14.3	1.239	1.05	1.006	0.371
Fe	mg/kg	10	0	3310	79200	17236	14700	10189	7154
K	mg/kg	200	0.5	<200	11000	1478	1100	1295	741
La	mg/kg	0.5	0	3.5	203	17.94	12.8	20.96	6.449
Mg	mg/kg	5	0	370	70500	4741	3720	4815	2002
Mn	mg/kg	0.5	0	33.8	2140	185	128.5	180	65.83
Na	mg/kg	15	0	20	19400	338	140	1368	88.96
Ni	mg/kg	1	0	1.2	228	23.41	18.65	21.09	11.56
P	mg/kg	7	0	59	7170	446	393	368	185
Pb	mg/kg	0.2	0	0.3	45.3	2.748	1.6	3.326	0.741
Rb	mg/kg	15	6.3	<15	270	60	54	33.55	26.68
Sc	mg/kg	0.1	0.2	<0.1	15.4	2.816	2.3	1.809	1.186
Sr	mg/kg	0.5	0	1.6	1040	25.34	7.7	98.23	3.781
Th	mg/kg	3	6.1	<3	66	8	6	6.160	3.709
V	mg/kg	0.5	0	4.5	183	34.99	30.9	19.65	15.72
Y	mg/kg	0.5	0	0.9	169	6.366	4.4	10.97	2.372
Zn	mg/kg	0.5	0	3.7	348	27.40	20.9	24.17	12.45

Table 3.5: Group8: Statistical summary of the mafic rock elements from C-horizon

Element	Unit	DL	%<DL	min	max	mean	med	stddev	mad
Co	mg/kg	0.2	0	1.2	44.3	8.22	7	5.029	3.706
Cr	mg/kg	0.5	0	2.2	471	36.16	28.35	35.09	16.23
Cu	mg/kg	0.5	0	2	149	21.96	16.2	18.44	10.82
Fe_INAA	mg/kg	100	0	6800	119000	37800	35750	14552	14455
Mg	mg/kg	5	0	370	70500	4741	3720	4815	2002
Mn	mg/kg	0.5	0	33.8	2140	185	128.5	180	65.83
Ni	mg/kg	1	0	1.2	228	23.41	18.65	21.09	11.56
Sc	mg/kg	0.1	0.2	<0.1	15.4	2.816	2.3	1.809	1.186
Ti	mg/kg	0.5	0	48.8	5730	895	807	515	405
V	mg/kg	0.5	0	4.5	183	34.99	30.9	19.65	15.72

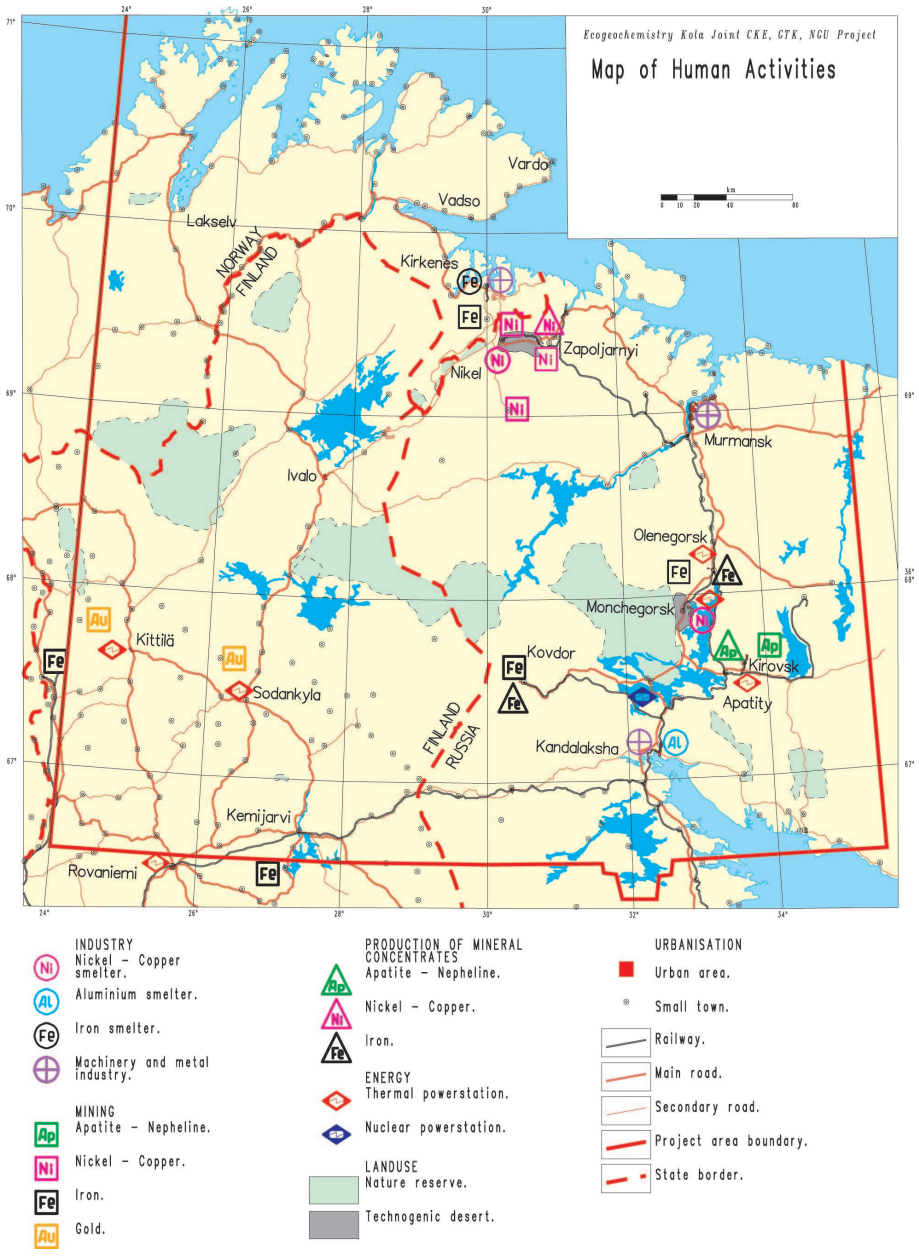


Figure 3.3: Map of human activities

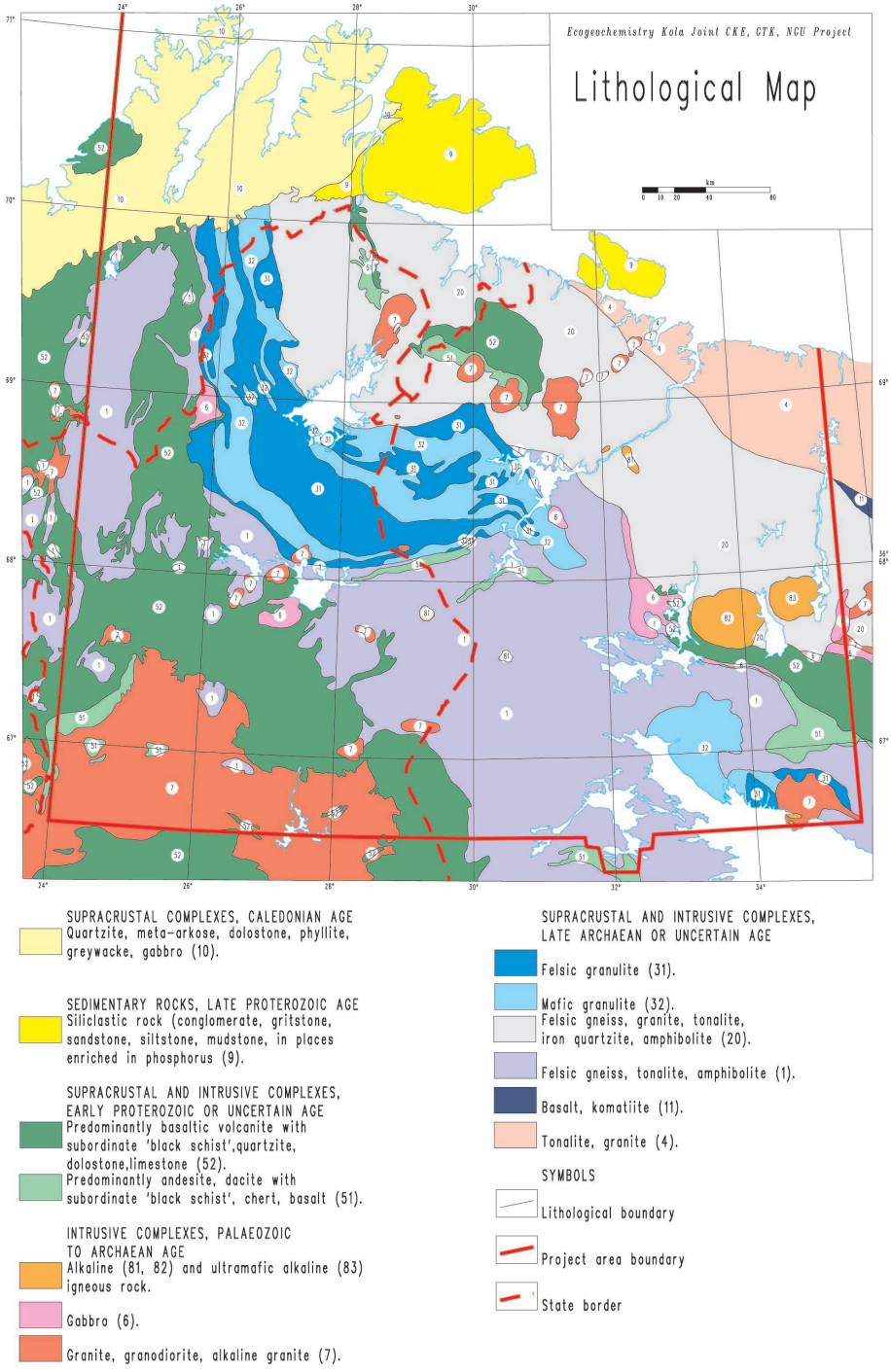


Figure 3.4: Lithological map of the investigated area

Table 3.6: Statistical summary of the mainly used elements of the O-horizon

Element	Unit	DL	%<DL	min	max	mean	med	stddev	mad
As	mg/kg	0.05	0	0.364	43.5	1.6	1.16	0.23	0.46
Ca	mg/kg	5	0	460	25400	3120.3	2960	35.09	785.78
Co	mg/kg	0.03	0	0.21	96	3.12	1.57	18.44	1.11
Cu	mg/kg	0.01	0	2.69	4080	43.69	9.69	14552	5.14
Fe	mg/kg	10	0	430	44800	2878.39	1970	4815	1245.38
Mg	mg/kg	10	0	240	5830	905.43	750	180	296.52
Mn	mg/kg	1	0	11.1	5470	199.42	126	21.09	108.23
Na	mg/kg	10	3.4	5	2350	87.67	60	1.809	29.65
Ni	mg/kg	0.3	0	1.5	2880	50.98	9.18	515	7.74
Pb	mg/kg	0.04	0	4.07	1110	24.12	18.8	19.65	7.41
Sr	mg/kg	0.2	0	6.09	1430	40.87	28.8	515	13.64
V	mg/kg	0.02	0	1.08	48.9	6.36	4.86	19.65	2.39

3.2.2 O-horizon

A big number of various combinations was experimentally examined in order to make the interpretation of the results as unambiguous as possible. The statistical summary of the elements mainly used is given in Table 3.6.

3.3 Geology of the Walchen Data

In the Walchen region south of the town Öblarn in Styria, Austria, extensive zones of ore can be found. They mainly consist of iron as well as copper pyrities, galenite, zinc blende, magnetic pyrities, argentiferous ore, arsenic pyrities, antimonit, pyrargirit and gundmundit. The massiveness of the deposits amounts to several metres in some points. The tunnels (galeries) are located between 1100 m and 1550 m over the sea level and the processing plants and smelting furnaces are located on the floor of the Walchen rift about 980 metres over the sea level. The rich deposit was mined, and the ore was dressed and smelted over centuries - for the extraction of copper, lead, silver, gold, sulphur and vitriol. In those days the economical basis of Öblarn was mining. In the Walchen rift there are also significant marble deposits. Quarrying them was stopped only a few decades ago. For further information visit the internet site *http://www.argis.at/argis_neu/walchen.html*. Recent scientific examinations and papers are treated as a preparation for later conservation of this modern industrial monuments and as a help in creating a display for the Öblarner copper trail (see e.g. the internet site *http://www.argis.at/argis_neu/walchen.html*).

The samples were taken from a forest stretching over an area of 90 km^2. Altogether 773 soil samples were gathered. The density of the soil samples was eight samples per square kilometre.

With the following selection of elements the various rock types should fall into different

Table 3.7: Statistical summary of the used elements of the Walchen data

Element	Unit	min	max	mean	med	stddev	mad
Na	wt.-%	0.15	5.59	1.58	1.5	0.52	0.44
K	wt.-%	0.08	9.9	2.69	2.8	0.86	0.62
Ca	wt.-%	0	18.9	0.58	0.14	1.79	0.12
Mg	wt.-%	0.03	11	1.43	1.19	1.09	0.50
Al	wt.-%	0.81	16	10.03	10.15	1.69	1.26
Fe	wt.-%	0.4	13	6.52	6.4	1.83	1.48
Si	wt.-%	1.1	32.1	21.52	21.75	3.15	2.44
P	wt.-%	0	0.52	0.1	0.09	0.059	0.04

Table 3.8: Statistical summary of the main elements of the Rock data

Element	Unit	min	max	mean	med	stddev	mad
SiO2	wt.-%	0.29	90.06	60.73	61.66	12.09	11.12
AlO3	wt.-%	0.09	21.12	14.15	14.89	3.2	2.21
Fe2O3	wt.-%	0.06	18.8	5.69	5.48	3.29	3.53
TiO2	wt.-%	0.006	4.12	0.70	0.65	4.49	0.34
MgO	wt.-%	0.03	16.49	3.17	2.77	2.45	2.67
CaO	wt.-%	0.11	56.04	5.77	4.04	6.61	3.61
Na2O	wt.-%	0.05	8.92	2.91	2.84	1.42	1.53
K2O	wt.-%	0.02	7.17	2.43	2.23	1.59	1.53
MnO	wt.-%	0.002	0.58	0.09	0.08	0.06	0.04
P2O5	wt.-%	0.005	0.85	0.14	0.13	0.098	0.06

clusters: Na, Mg, Cr, Co, Ni_AAS (AAS: analysed by atom absorption spectralphotometrics), Ba_INAA, Cu, Sr. The statistical summary of the elements used is given in Table 3.6.

3.4 Rock Data

The Rock data consist of 500 rock samples coming from an area around Trondheim, Norway. It can be assumed that this data set shows a good cluster structure. This assumption will be confirmed in Chapter 6. For this reason we used this data set with the elements Al2O3, Fe2O3, TiO2, MgO, CaO, Na2O, K2O, MnO, P2O5, LOI comparing particular validity measures.

Chapter 4

Validity Measures

4.1 Introduction

The procedure of evaluating the results of a clustering algorithm is known under the term *cluster validity*.
Clustering is an unsupervised process since there are no predefined classes and no examples which show what kind of relations should be regarded as valid among the data. As a consequence, the clustering output of the data set requires some sort of evaluation.
The focus of this chapter is to give an overview over the methodology of drawing inferences about the number of clusters in a data set, of improving the accuracies of an arbitrary clustering algorithm and of evaluating cluster results.

There are some problems to be looked at:

- Applying more than one algorithm results in varying classification. Which allocation is the best?
- Which cluster size is correct?
- To each structure of the data a special algorithm can be applied. Which structure is characteristic for the data and which algorithm is the optimal choice for these data?

These problems require a clustering validation.
The following examples should provide a deeper insight.

In Figure 4.1 it becomes obvious that the assignment into four groups is incorrect. If we applied various validity criteria this would produce very bad results for this classification. If the clustering is examined on the basis of a criterion which considers only the density of a cluster, "good" results will be produced by this validity measure. If, however, the clustering is examined on the basis of a criterion aiming at a separation of the clustered objects, "bad" results will be produced for this classification.

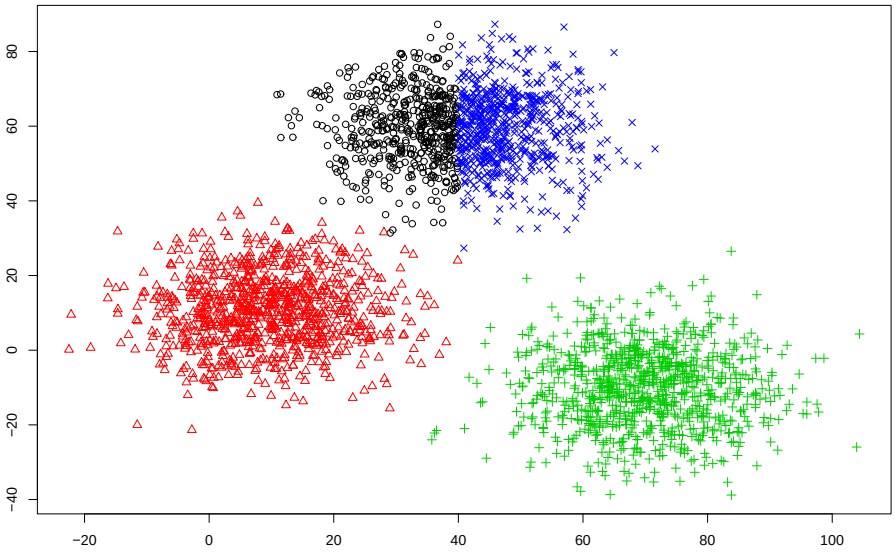


Figure 4.1: Data xclara from R; incorrect cluster size

However, there are very different kinds of validity criteria, which are strongly aligned after the intended purpose. Which validity criteria actually are suitable can not be answered in a general way. This depends on the existing data and on the taste of the viewer.

A test data set consisting of two groups of different size should display various problems of the applied algorithms. With the help of the algorithms PAM and AGNES (Kaufman and Rousseeuw, 1990) this data set was correctly divided into one cluster with four data points and one cluster with 25 data points (see Figure 4.2). Figure 4.3 illustrates the problem of cluster-splitting with the C-means (kmeans) algorithm. The algorithm FANNY of Kaufman and Rousseeuw (1990) does not work with this kind of constructed data set in a satisfactory manner. Figure 4.5 shows that the FCM algorithm classifies the objects correctly, however, it ignores the different sizes of the clusters. The different lines explained in the legend refer to the membership coefficients. We can see that with the help of the Malahanobis distance the GK algorithm paid attention to the different sizes of the clusters, producing, however, two wrong prototypes (see Figure 4.6). The GG algorithm does achieve the right classification, but the memberships for the objects are almost identical in cluster 1 and 2 (see Figure 4.7. This algorithm, in apportion to the FCM algorithm, puts the prototypes exactly in the middle of these two clusters.

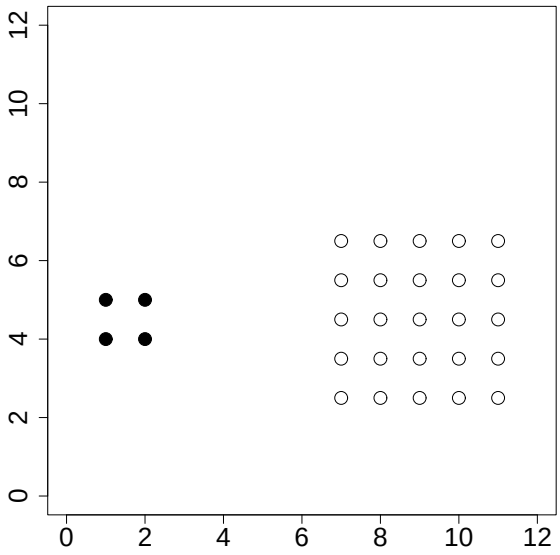


Figure 4.2: Correct classification, found with AGNES and PAM

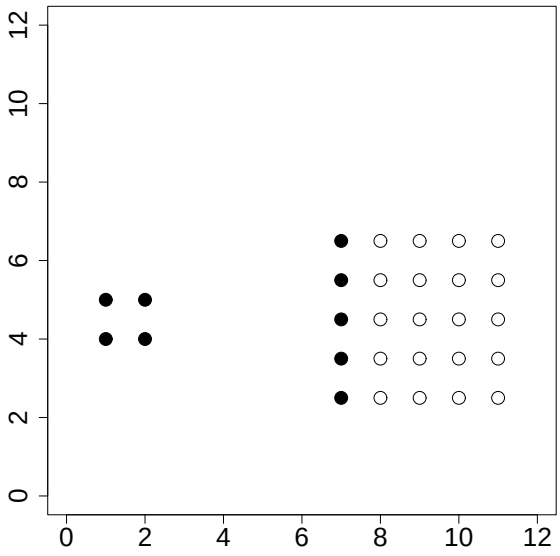


Figure 4.3: Incorrect classification, found with kmeans

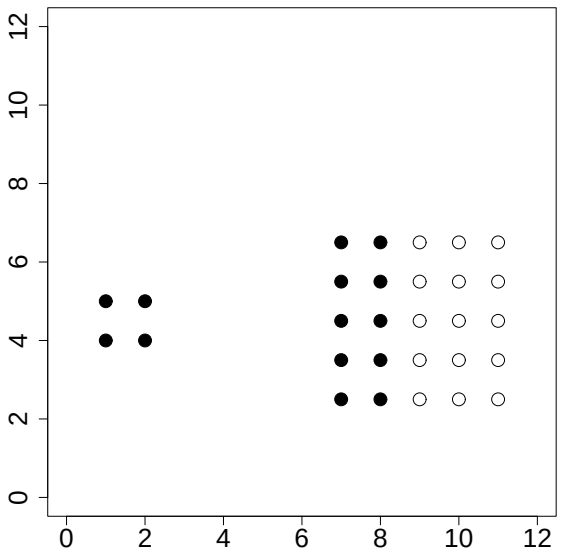


Figure 4.4: Incorrect classification, found with FANNY

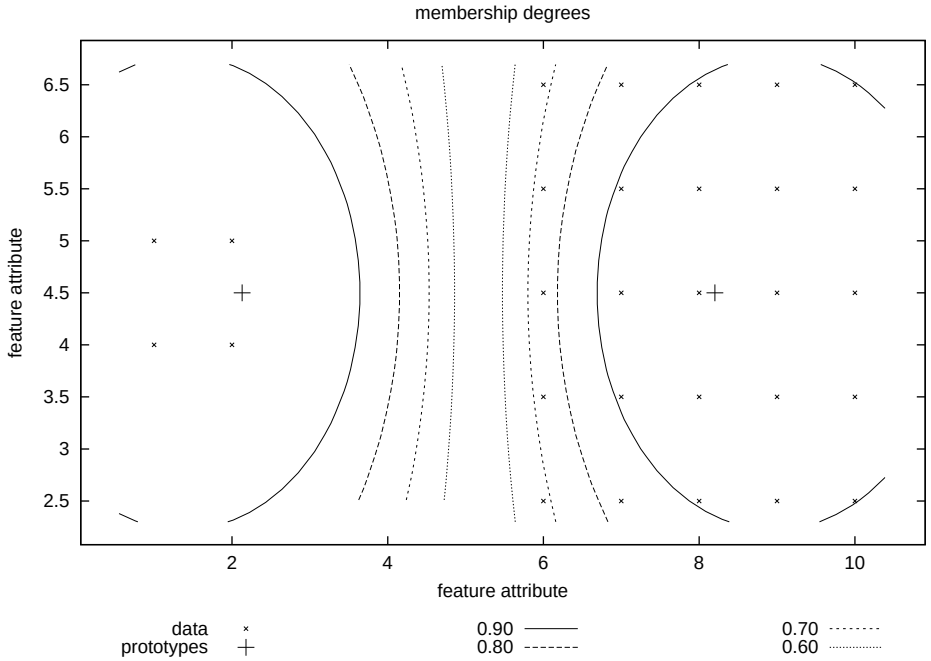


Figure 4.5: Correct classification, found with the FCM algorithm

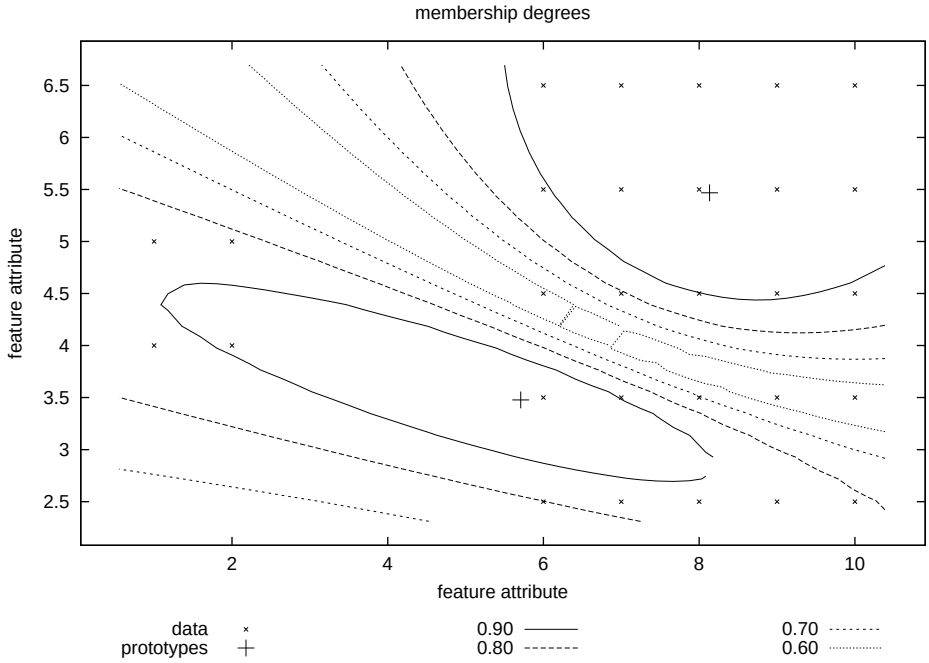


Figure 4.6: Incorrect classification of the GK algorithm

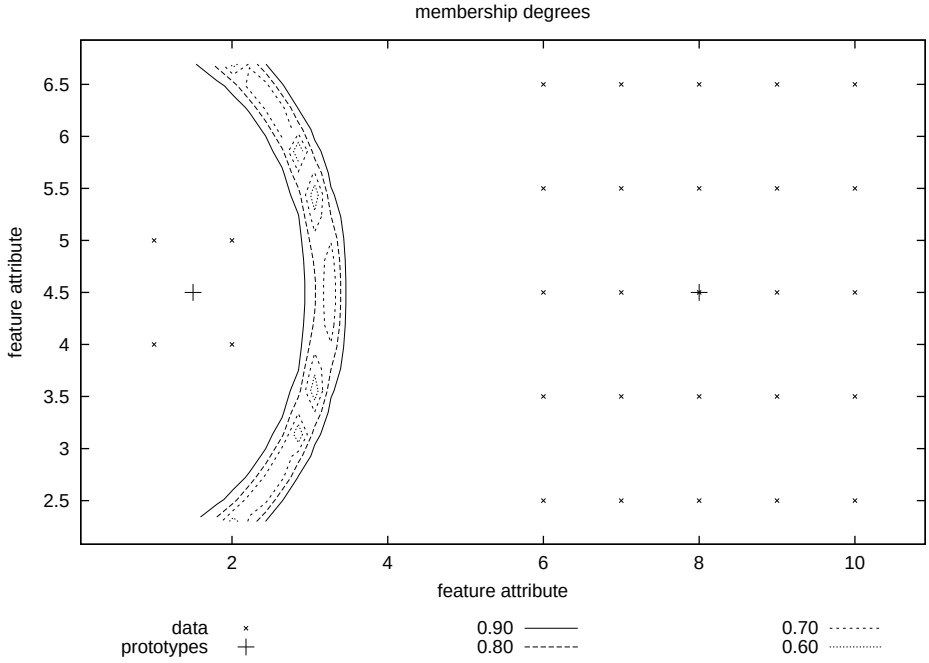


Figure 4.7: Correct classification, found with the GG algorithm

Generally speaking there are three approaches in determining cluster validity.
The term "validation of a clustering procedure" is known under the name *external criteria*, which measures how well the clustering results match some prior knowledge about the data. It is assumed that this information is not, in general, computable from the data set. They are not used to estimate the number of clusters, they are used to compare different partitions.
Internal criteria evaluate the results of a clustering algorithm to estimate the number of clusters and they are computed from the same observations that are used to create a partition.

The third approach is the *relative criteria*, and the basic idea is the evaluation of a clustering structure by comparing it to other clustering schemes.
Important literature on these validity indices are e.g. Andenberg (1973); Bezdek and Pal (1998); Dom (2001); Jain et al. (1999); Milligan (1996); Fridlyand (2001); Fraley and Raftery (1998); Halkidi and Vazirgiannis (2001a,b); Halkidi et al. (2001).

Internal criteria raise the question why not using the objective function to evaluate the clustering? The answer is that we only use the objective function when we know exactly what is desired in a particular application and when there is a feasible algorithm for finding the optimal cluster size. The objective function (e.g. in case of C-means) cannot provide a prediction of the optimal cluster size, since with increasing number of clusters naturally the distance of the data to the prototypes becomes smaller.

4.2 External Criteria

In this approach of the external criteria validity measures the basic idea is to test whether the points of the data set are randomly structured (hypothesis H_0) or not. If the points of the data set are randomly structured, cluster analysis does not produce "good" results. Statistical tests, like Monte Carlo techniques, are needed to test this hypothesis. We can evaluate the clustering structure C, by comparing it to an independent partition of the data P built according to our intuition about the clustering structure of the data set.

The Monte Carlo Process:
The first phase involves the generation of a special data set. This data set may include alternative variance/covariance matrices, controlling the number of clusters, the spacing between clusters, the data dimension, the relative cluster size, or the introduction of error into the data.
The second phase involves analysing the generated data with selected clustering methods.
The primary of the third phase is to compare the true cluster structure in the generated data with the partition which was produced by the clustering procedure.

Consider $C = \{C_1, \ldots, C_{n_c}\}$ to be a clustering structure of a data set X and P is a defined partition of the data. For a couple of two different points it provides the following possibilities:

	S	**D**
S	a	b
D	c	d

Table 4.1: Table for paired comparsion between partitions

1. **SS:** both points belong to C_i and P_i
2. **SD:** both points belong to C_i and different groups of P
3. **DS:** both points belong to P_i and different cluster of C
4. **DD:** both points belong to different clusters of C and to different groups of P_i

These possibilities can be shown in a general form of a 2×2 table (Table 4.1) in which all combinations of these data units are classified.

Thus a, b, c and d are the numbers of **SS**, **SD**, **DS** and **DD** pairs respectivitly, and $a + b + c + d = L$.
Now we can define the following indices to measure the degree of similarity between C and P.

4.2.1 Rand, Jaccard, Folkes and Mallows, and the Hubert Indices

In terms of the entities in Table 4.1, *Rand's* measure of similarity between partitions is defined as

$$R = \frac{a+d}{L} \quad . \tag{4.1}$$

A similar index is the *Jaccard Coefficient*

$$J = \frac{a}{a+b+c} \quad . \tag{4.2}$$

These two indices take values between 0 and 1.
Other indices are the *Folkes and Mallows index*

$$FM = \frac{a}{\sqrt{(a+b)(a+c)}} \tag{4.3}$$

and the *Hubert's* Γ *statistic*

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij} y_{ij} \tag{4.4}$$

where $M = n(n-1)/2$, n is the number of points in the data set and x_{ij} and y_{ij} are elements of the matrices P and C that we want to compare.

In the case of these four indices a high value indicates a great similarity between C and P. The *normalised* Γ *statistic* is defined as

$$\hat{\Gamma} = \frac{1}{M} \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (x_{ij} - \mu_x)(y_{ij} - \mu_y)}{\sigma_x \sigma_y} \tag{4.5}$$

μ_x, μ_y, σ_x and σ_y are the means and standard deviations of P and C. All these statistics have right tailed probability density functions under the random hypothesis. However, the calculation of the probability density function of these indices is difficult. A solution to this problem is to use Monte Carlo techniques. The procedure is as follows:

FOR $i = 1$ to r
Generate a data set X_i with the same dimension as X.
Assign each vector of X_i according to the partition P.
Run the same algorithm used to produce structure C, for each X_i, and let C_i the resulting structure.
Compute $q(C_i)$, the value of the defined index q for P and C_i.
END FOR
Reject the hypothesis H_0 if q's value for our data set is greater than $(1-\rho) \cdot r$ of q_i's values, of the respective synthetic data sets X_i, where ρ is the significant level.

Example:
Assume a given data set, X, containing 100 three-dimensional points. The points of X form four clusters of 25 points each. Each cluster is generated by a normal distribution. The covariance matrices of these distributions are all equal. We independently group data set X in four groups according to the partition P for which the first 25 points belong to the first group P_1, the next 25 belong to the second group P_2, etc. We run the C-means clustering algorithm for $n_c = 4$ clusters and we assume that C is the resulting clustering structure. We compute the values of the indices for the clustering C and the partition P and get the values of the indices of R, J, FM and Γ. Next we generate 100 data sets $X_i, i = 1, \ldots, 100$, and each one of them consists of 100 random points using the uniform distribution. According to the partition P defined earlier for each X_i we assign the first 25 of its points to P_1 and the second, third and fourth groups of 25 points to P_2, P_3 and P_4 respectively. Then we run C-means i-times, one time for each X_i, so as to define the respective clustering structures of the data sets, denoted by C_i. For each of them we compute the values of the indices R_i, J_i, FM_i and Γ_i, $i = 1, \ldots, 100$. We set the significance level $\rho = 0.05$ and we compare these values to the R, J, FM and Γ values corresponding to X. We accept or reject the null hypothesis whether $(1-\rho) \cdot r = (1-0.05) \cdot 100 = 95$ values of R_i, J_i, FM_i and Γ_i are greater or smaller than the corresponding values of R, J, FM and Γ.

4.3 Internal Criteria

Internal criteria are cluster validity measures which evaluate the clustering result of an algorithm by using only quantities and features inherent in the data set.

We can define

$$B_{n_c} = \sum_{i=1}^{n_c} \|v_i - \overline{v}\|^2 \tag{4.6}$$

where $\|\cdot\|$ denotes the Euclidean norm and where

$$\overline{v} = \frac{1}{n_c} \sum_{i=1}^{n_c} v_i \tag{4.7}$$

and

$$W_{n_c} = \sum_{i=1}^{n_c} \sum_{j \in C_i} \|x_j - v_i\|^2 \tag{4.8}$$

to be the matrices of between and within the n_c-clusters sum of squares and crossproducts. Another approach is the pooled within sum of squares and can be found in (Tibshirani et al., 2000).
These statistics measure the dispersion of the data points in a cluster and between the clusters, respectively. Now the following indices can be defined:

4.3.1 Calinski-Harabasz and Hartigan's Indices

Calinski and Harabasz:

$$CH = \frac{B_{n_c}/(n_c - 1)}{W_{n_c}/(n - n_c)} \tag{4.9}$$

where n is the number of data points and n_c is the number of clusters.
Hartigan:

$$H = \log \frac{B_{n_c}}{W_{n_c}} \tag{4.10}$$

The minimum value for these two indices is taken as the proposed number of clusters.

4.3.2 The Average Silhouette Width

Another approach from Kaufman and Rousseeuw (1990) is the *Average Silhouette Width*. They suggest selecting the number of clusters n_c which gives the largest average silhouette width. Silhouette plots visualise the quality of the clustering for every object separately with a horizontal bar which is determined by the silhouette value s_i of the object. The calculation of the silhouette value proceeds as follows: First, the average dissimilarity a_i of an object i to the other members of the same class C_k is given by

$$a_i = \frac{1}{|C_k| - 1} \sum_{i,j \in C_k, j \neq i} d(i,j) \quad . \tag{4.11}$$

Now consider any cluster $C_l \neq C_k$ and define the average dissimilarity $d_{i,C}$ of object i to C_l

$$d_{i,C} = \frac{1}{|C_l|} \sum_{j \in C_l} d(i,j) \quad . \tag{4.12}$$

The cluster C_l for which this average linkage distance $d_{i,C}$ attains a minimum b_i is called the neighbour of object i.
Finally, the silhouette value is defined as

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad . \tag{4.13}$$

The values of s_i lies in the interval $[-1, 1]$. If the value of s_i is close to -1, then there is a bad classification of object i. s_i close to 0 indicates that the object s lies between two clusters. A value of s_i close to 1 means that object i is well classified.

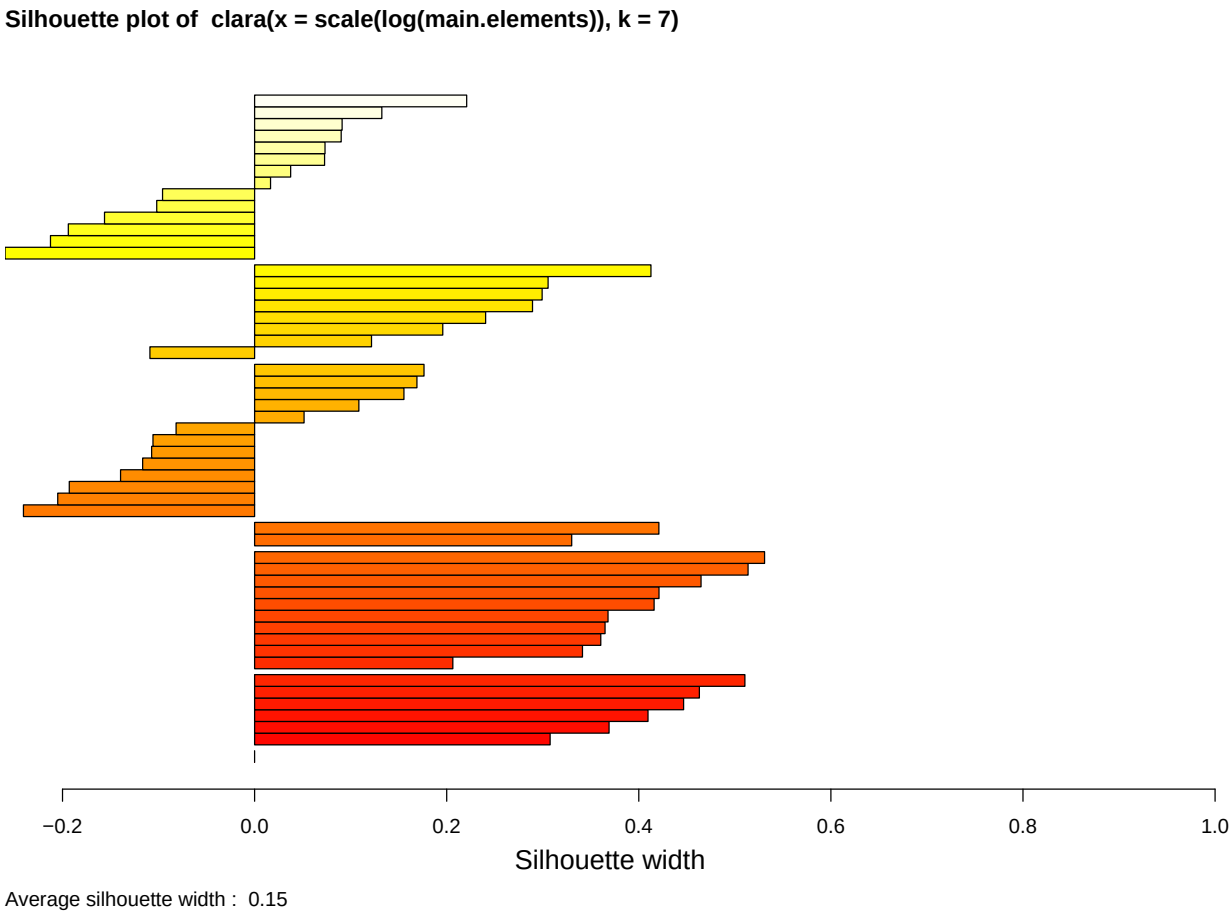


Figure 4.8: Graphical output for the clustering of the main elements of the C-horizon with CLARA into seven clusters.

The silhouettes for $n_c = 7$ clusters (produced with CLARA) of the main elements of the C-horizon are given in Figure 4.8. Only cluster 5 and 6 have a reasonable silhouette width.

However, the average silhouette width can be computed for all possible numbers of clusters. The resulting value is called silhouette coefficient. Its maximum can be used for determining an approximate number of clusters. Experience with the silhouette coefficient has led Kaufman and Rousseeuw (1990) to a rather subjective interpretation, which is summarised in Table 4.2.

Table 4.2: Subjective interpretation of the silhouette coefficient, defined as the maximal average silhouette width for the entire data set.

Silhouette Coefficient	Proposed Interpretation
$0.71 - 1.00$	A strong structure has been found.
$0.51 - 0.70$	A reasonable structure has been found.
$0.26 - 0.50$	The structure is weak and could be artificial.
≤ 0.25	No substantial structure has been found.

4.3.3 Validity Indices for Hierarchical Clustering

Internal cluster criteria for hierarchical clustering can be described as follows.
The *cophenetic similarity* c_{ij}, introduced by Legendre and Legendre (1998), of two objects i and j is defined as the similarity level at which objects i and j become members of the same cluster during the course of clustering.
We may define a statistical index to measure the degree of similarity between the *cophenetic matrix* P_c with the elements c_{ij}, i.e. $P_c = [(c_{ij})]$ and the dissimilarity matrix $P = [(d_{ij})]$.

This index is called *cophenetic correlation coefficient* and is defined as:

$$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}c_{ij} - \mu_p\mu_c}{\left[\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2 - \mu_p^2\right)\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^2 - \mu_c^2\right)\right]^{\frac{1}{2}}} \tag{4.14}$$

where $M = n(n-1)/2$ and n is the number of points in the dataset and μ_p and μ_c are the means of matrices P and P_c respectively. The value of CPCC lies between -1 and 1, if a value is close to 0 it is an indication of a significant similarity between the two indices.

Another approach is the *agglomerative coefficient* and accordingly the *banner plot*. A hierarchy of clusters can be represented with a dendrogram or with a banner plot proposed by Kaufman and Rousseeuw (1990), p. 211.

The overall width of the banner is very important because it gives an idea of the amount of structure that has been found by the algorithm. When the data possess a clear cluster

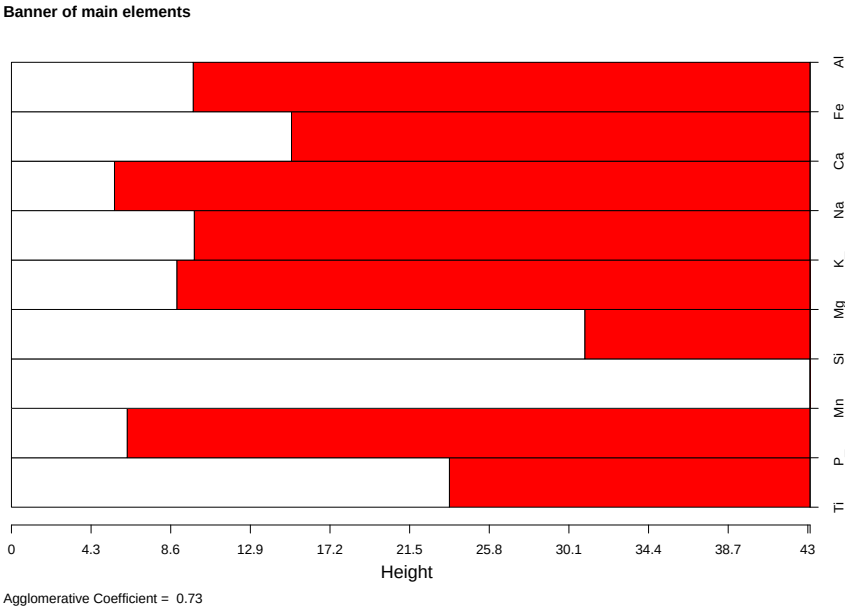


Figure 4.9: Banner of the variables of the main elements of the C-horizon

structure, the between-cluster dissimilarities will become much larger than the within-cluster dissimilarities, and as a consequence the black lines in the banner will become longer. For each object i, we look at the line containing its label and measure its lenght $l(i)$ on the $0-1$ scale above or below the banner. The banner plots distances at which observations and clusters are merged. The observations are listed in the order found by the AGNES algorithm, and the numbers in the height vector are represented as bars between the observations.

The agglomerative coefficient of the data set is then defined as

$$AC = \frac{1}{n}\sum_{i=1}^{n}(1 - l(i)) \quad . \tag{4.15}$$

The agglomerative coefficient lie in $[0, 1]$ and it is simply the average width of the banner. In Figure (4.9), $AC = 0.73$, which refer a strong clustering structure. The agglomerative coefficient tends to grow with the number of objects. Therefore it should only be used to compare data sets with similar numbers of objects.

4.4 Relative Criteria

The major difference to external and internal validity criteria is, that statistical tests are not involved. There are two main approaches to find the best number of clusters. The first one investigates the data set for numbers of clusters within a certain range $[c_{min}, c_{max}]$. The minimum and the maximum values have to be defined in advance by the user.
For each partition obtained the validity is measured by a *global validity* measure. The partition with the best value becomes the optimal partition. However it can not be said that

this partition is the best partition. Some of these criteria evaluate only the allocation of the data to the clusters. Other criteria evaluate the form of the cluster or how well the clusters are separated. It is only the best partition concerning the used validity measure.

The second approach starts with a high number of clusters and performs operations like merging of similar clusters or deleting tiny clusters to approach the final partition with an optimal number of clusters. For these operations, *local validity* measures are used.

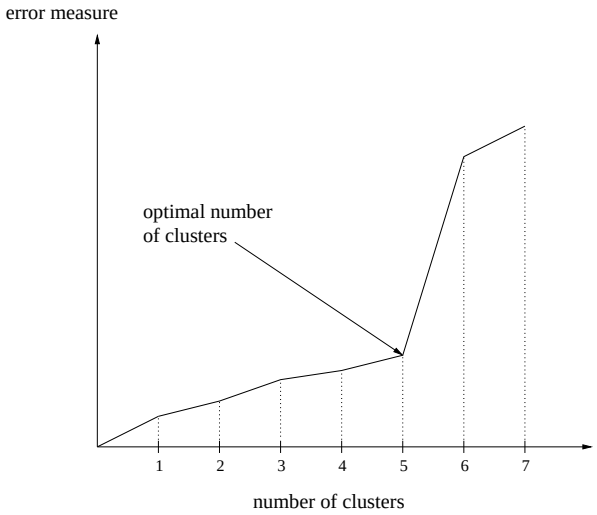


Figure 4.10: Plot of an error measure versus the number of clusters

Figure 4.10 nach Figure shows a typical plot of the error measure for a clustering procedure versus the number of clusters n_c employed: The error measure decreases monotonically as the number of clusters decreases, but from some n_c on the decrease flattens markedly. This location of a knee indicates the optimal number of clusters.

4.4.1 Indices for Non-Overlapping Partitions

In this section the validity indices suitable for non-overlapping partitions (crisp clustering) are discussed.

Davies Bouldin Index

This index is defined as follows:

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} \max_{j=1,\ldots,n_c, j \neq i} d_{ij} \tag{4.16}$$

where

$$d_{ij} = \frac{a_i + a_j}{d(i,j)} \tag{4.17}$$

n_c is the number of clusters, a_i and a_j are defined in (4.11) and $d(v_i, v_j)$ is the distance of the cluster centres v_i and v_j, and $d(i,j)$ is the distance between object i and object j.

This distance is small if cluster i and cluster j are well separated and each of the clusters is compact.

Dunn's Index

The index of Dunn is a very popular cluster validity index, which aims at the identification of compact and well separated clusters.
Let $d_{\min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$ denote the single linkage between two elements from different clusters C_i and C_j, and $d_{\max}(C_k) = \max_{x,y \in C_k} d(x, y)$ the largest distance of two objects from the same cluster. Then the Dunn index is given by

$$D = \min_{1 \le i \le n_c} \left\{ \min_{1 \le j \le n_c, j \neq i} \left\{ \frac{d_{\min}(C_i, C_j)}{\max_{1 \le k \le n_c} d_{\max}(C_k)} \right\} \right\} \quad . \tag{4.18}$$

n_c is the number of clusters.

Modified Hubert Γ Statistic

The definition of the modified Hubert Γ statistic is given by the equation

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d(i, j) \cdot Q(i, j) \tag{4.19}$$

where n is the number of objects in the data set, $d(i, j)$ is the (i, j)-th element of the dissimilarity matrix of the data set, and Q is an $n \times n$ matrix whose (i, j)-th element $Q(i, j)$ is equal to the distance between the representative points v_i and v_j of the clusters where the objects x_i and x_j belong. A high value of Γ indicates the existence of compact clusters. In practice one should see in the plot of Γ versus cluster number a significant knee, which is an indicator of the correct cluster size.

RMSSTD and RS indices

Other indices, also based on the approach of looking for a knee, are the RMSSTD (root-mean-square standard deviations) index and the RS (R-squared) index (see e.g. in Salazar et al., 2002).

The RMSSTD index is defined by

$$RMSSTD = \sqrt{\frac{W_{n_c}}{n - n_c}} \tag{4.20}$$

where W_{n_c} is defined in (4.8), and n_c is the number of clusters.

Validity Index	Name	Variables	Criterion
Davis Bouldin Index	DB	(X, C)	minimise
Dunn's Index	D	(X, C)	maximise
Modified Hubert Statistic	Γ	(X, C, v)	look for knees
Root-Mean-Square stand. dev.	RMSSTD	(X, C, v)	look for knees
R-Squared	RS	(X, C, v)	look for knees
SD Index	SD	(X, C, v)	look for knees
Inter-cluster Density	ID	(X, C, v)	minimise

Table 4.3: Seven crisp cluster validity indices

Inter-Cluster Density

The Inter-Cluster Density (ID) evaluates the average density in the region among clusters in relation to the density of the clusters and is defined as

$$ID = \frac{1}{n_c(n_c-1)} \sum_{i=1}^{n_c} \Big(\sum_{j=1, i \neq j}^{n_c} \frac{density(u_{ij})}{\max\{density(v_i), density(v_j)\}} \Big) \tag{4.21}$$

where v_i, v_j are the centres of clusters C_i, C_j and u_{ij} is the middle point of the line segment defined by the cluster centres v_i and v_j. The term density(u) is defined as

$$density(u) = \sum_{l=1}^{n_{(ij)}} f(x_l, u) \tag{4.22}$$

where $n_{(ij)}$ is the number of tuples that belong to the cluster c_i and c_j. It represents the number of points in the neighbourhood of u. The neighbourhood of a data point, u, is defined to be a hyper-sphere with center u and radius the average standard deviation of the cluster. The function $f(x, u)$ is defined as

$$f(x,u) = \begin{cases} 0, & \text{if } d(x,u) > \text{the average standard deviation of the clusters} \\ 1, & \text{otherwise} \end{cases}$$

A point belongs to the neighbourhood of u if its distance from u is smaller than the average standard deviation of clusters.

In Table 4.3 we can see seven crisp cluster validity indices with the necessary input (variables: X ... data set, C ... cluster result and v ... cluster centres) and criteria for the optimal cluster size.

4.4.2 Global Validity Measures for Fuzzy Clustering

In contrast to the previous sections we focus on the memberships now. To perform fuzzy clustering with global validity measures we need

- an initialisation algorithm for every number of clusters within the allowed range in order to initialise the prototype,
- a fuzzy clustering algorithm,
- an algorithm that calculates one or more global validity measures for the final partition,
- an algorithm which selects the best partition (with respect to the validity measure) from the results.

We can search for an appropriate value of n_c in points where a significant local change (knee) in value of the index occurs. This may be an indication for the optimal cluster size.

Partition Coefficient

The *partition coefficient* (Bezdek, 1981) is a very simple validity criterion, which is based on the idea that, with a good classification, the data can be assigned clearly to the clusters. The membership degrees should be close to 1 or close to 0.
The partition coefficient is defined as

$$PC(U) = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij}^2 \quad . \tag{4.23}$$

The optimal choice of n_c is given by

$$\min_{n_c} PC(U) \tag{4.24}$$

where the range of the PC values lie in the interval $[-1, -\frac{1}{n_c}]$. Figure 4.11 shows the partition coefficient in relation to the number of clusters of the main elements from the C-horizon. The data have been log-transformed and standardised, and were clustered with the Gustafson-Kessel algorithm.

Partition Entropy

Similar to the partition coefficient, the *partition entropy* (Bezdek, 1981) is only using memberships of the data to the clusters.
The partition entropy is defined as

$$PE(U) = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij}\ln(u_{ij}) \tag{4.25}$$

The values of PE are in the interval $[0, \ln(n_c)]$. The partition entropy is to be minimised. The graphical output of the partition entropy is similar to Figure 4.11 and is shown in Figure 4.12.

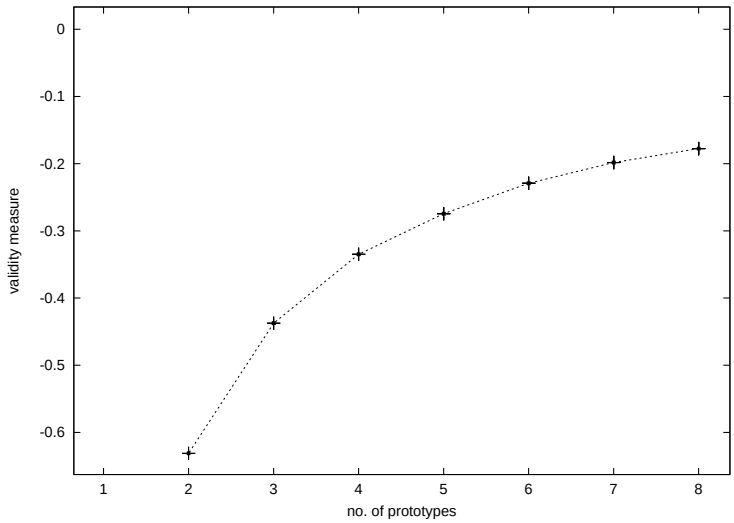


Figure 4.11: Partition coefficients of the main elements from the C-horizon clustered with the Gustafson-Kessel algorithm.

Compactness and Separation

This validity measure, published by Xie and Beni (1991), becomes minimal for compact, well-separated clusters. The distance of the data to the clusters is compared with the distance between the clusters.
The measure is defined as

$$S(U,X) = \frac{\sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij}^2 \cdot d^2(x_j, v_i)}{n \cdot \min\left\{d^2(v_i, v_j) | i,j \in \{1,\ldots,n_c\}, i \neq j\right\}} \quad . \tag{4.26}$$

The numerator term represents the homogeneity of the data in a cluster, and therefore it should be as small as possible. The denominator evaluates the heterogeneity of the data from different clusters. With this validity measure a small value indicates a good classification. Figure 4.13 shows a knee at a number of four clusters. This is an indication for the optimal cluster size.

Fukuyama Sugeno Index

The *Fukuyama-Sugeno index* (Fukuyama and Sugeno, 1989) is defined as

$$FS(U,X) = \sum_{i=1}^{n_c}\sum_{j=1}^{n} u_{ij}^m \Big(d^2(x_j, v_i) - d^2(v_i, v)\Big) \tag{4.27}$$

where v is the mean vector of X.
The fist term in the brackets measures the compactness of the cluster. The second term measures the distance between the cluster prototypes. Small values for FS imply compact and well separated clusters.

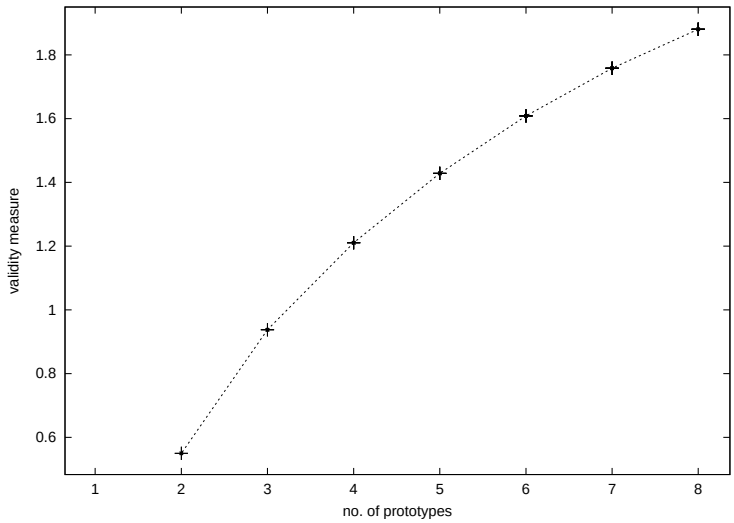


Figure 4.12: Partition entropy of the main elements from the C-horizon clustered with the Gustafson-Kessel algorithm.

Hyper-volume and Partition Density

The *fuzzy hyper-volume* (Gath and Geva, 1989) becomes minimal for small clusters and is defined as

$$FHV(U) = \sum_{i=1}^{n_c} \sqrt{det(S_i)} \tag{4.28}$$

where S_i is given by (2.21) . The fuzzy hyper-volume applied to the clustering result of the log-transformed and standardised main elements from the C-horizon is shown in Figure 4.14.

Let $X_i \supset X$ denote a subset of the data set with a distance less than 1 to the prototype v_i of cluster C_i, that is,

$$X_i = \{x_j \in X | (x_j - v_i)^T A_i^{-1} (x_j - v_i) < 1\} \quad , \tag{4.29}$$

with A_i defined by (2.20). The sum of all membership degrees of elements in X_i to cluster C_i is denoted by u_i. This value provides an estimation on the density of the region near the cluster centre occupied by data objects. Thus the *average partition density* is defined as

$$APD(U) = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{u_i}{\sqrt{A_i}} \tag{4.30}$$

and the *partition density* as

$$PD(U) = \frac{\sum_{i=1}^{n_c} u_i}{FHV(U)} \quad . \tag{4.31}$$

The partition density measures the density of the complete partition whereas the average partition density measures the cluster density. Both indices (all from Gath and Geva, 1989)

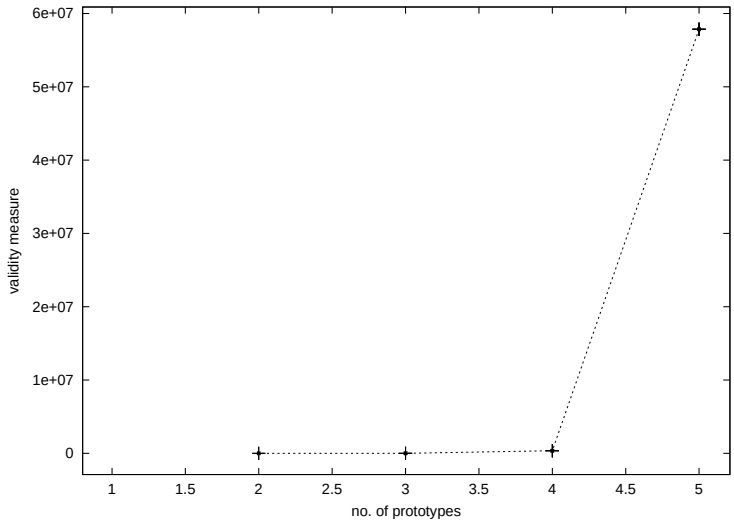


Figure 4.13: Compactness and separation of the main elements of the C-horizon clustered with the Gustafson-Kessel algorithm.

need to be maximised for a good partition (Höppner, 2000, stores their negative values). The average partition density in Figure 4.15 and the partition density in Figure 4.16 show that two clusters are enough to produce a "good" clustering result.

Contractive Properties of FCM

This measure provides an upper bound for the norm of the derivative of the FCM iteration mapping, which can be used to decide whether an FCM solution is a contractive fixed point and thus a minimum of (2.11). If the norm drops below 1, the fixed point is contractive and represents a local minimum for sure. For details see Höppner (2000). In Figure 4.17 one can not appropriate determine the optimal number of clusters.

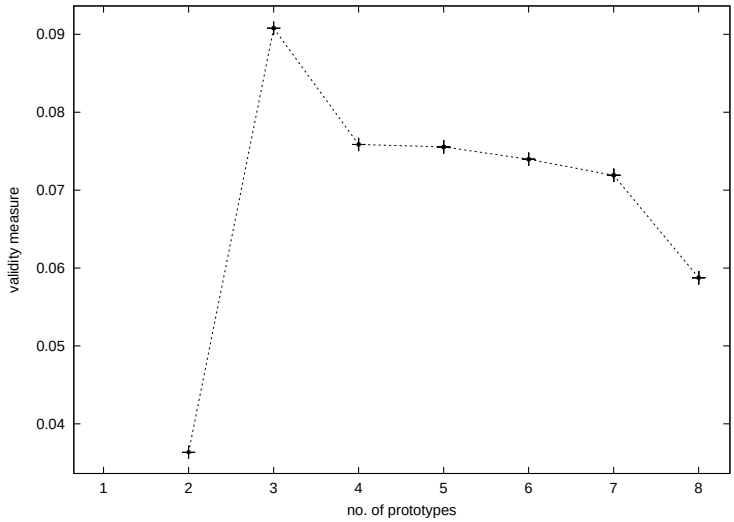


Figure 4.14: Fuzzy hyper-volume of the main elements from the C-horizon clustered with the Gustafson-Kessel algorithm.

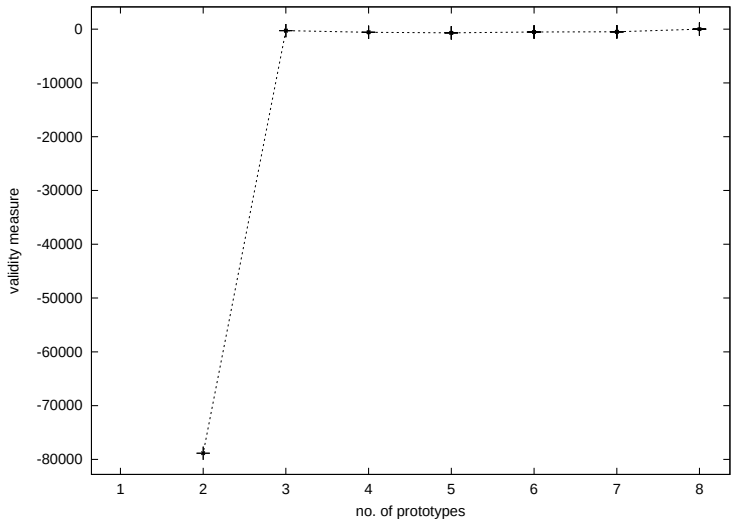


Figure 4.15: Average partition density of the main elements from the C-horizon clustered with the Gustafson-Kessel algorithm.

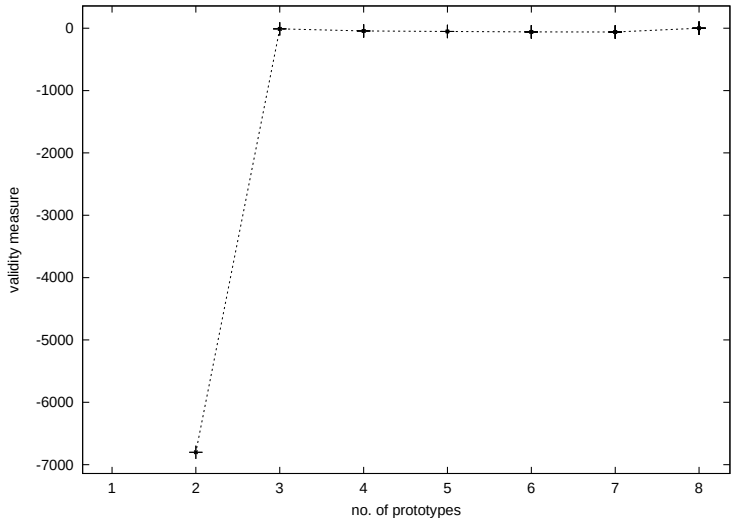


Figure 4.16: Partition density of the main elements from the C-horizon clustered with the Gustafson-Kessel algorithm.

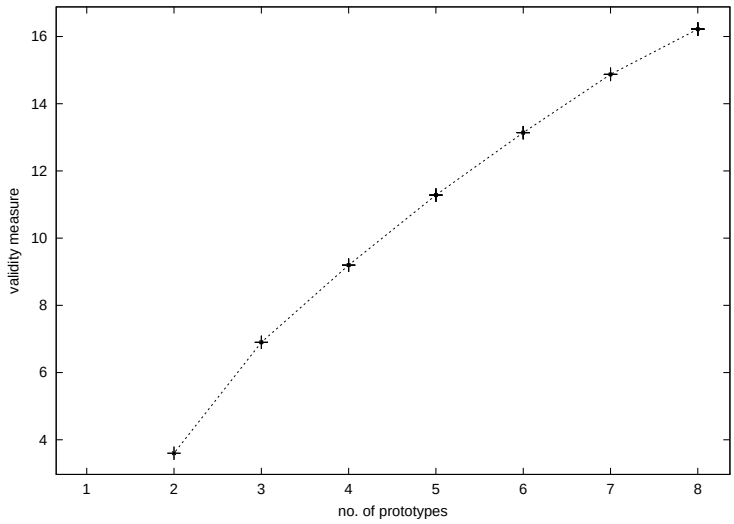


Figure 4.17: Contractive index of the main elements from the C-horizon clustered with the FCM algorithm.

4.4.3 Local Validity Measures for Fuzzy Clustering

To perform clustering with local validity measures, the following algorithms are needed.

- an initialisation algorithm,
- a fuzzy clustering algorithm,
- an algorithm that calculates local validity measure for each cluster in order to classify them,
- an algorithm that modifies the set of clusters according to local validity measures.

Local validity measures are used to decide cluster properties like **tiny**, **good**, and **compatible**. A tiny cluster has only a small number of features compared to other clusters. A good cluster has small distances to the assigned features and almost hard memberships. And two clusters are compatible if they approximate the same model in the data and it is possible to approximate the union of assigned features with a single instead of two clusters.
In order to approach an optimal number of clusters, tiny cluster will be deleted and compatible clusters are merged. To reduce the complexity of the data set it can be helpful to extract a well-separated cluster and redo the fuzzy clustering without the extracted data.

Tiny clusters are often characterised by

$$\sum_{j=1}^{n} u_{ij}^{m} < C_{tiny} \tag{4.32}$$

where m is the fuzzyfier.

If this sum drops below a certain threshold C_{tiny}, the cluster is considered to be tiny and marked for deletion.

There are many other local validity measures. In most cases these measures are used to detect lines and circles (shell-clustering).
But in the geochemical data sets of the C-horizon and O-horizon there are no clear line or circle structures. Lines and circle structures appear mainly in pattern recognition.

Chapter 5

Results

The O-horizon and the C-horizon of the Kola data and the Walchen data were analysed with the help of the algorithms of hierarchical clustering, the CLARA algorithm of Kaufman and Rousseeuw (1990) and the fuzzy cluster algorithm of Höppner (2000). Kaufman and Rousseeuw (1990) developed the FANNY algorithm for the use in fuzzy clustering. This algorithm, applied to standardised large data sets, constantly gives same-sized memberships of the objects to all clusters. Modifications of the Fortran programme (e.g. raising the limit of the number of iterations) did not solve the problem.

With the algorithms applied and with various selection of elements for the clustering, hundreds of results were produced. Some of the most important ones are presented and described in this thesis.

The number of clusters can be defined with the help of validity measures when doing fuzzy clustering. Some of the values of the validity measures are shown in tables and are analysed.

5.1 O-horizon

In the humus samples (O-horizon; 0-3 cm under the surface) the effects of environmental pollution and weather conditions should be discernible. In the western half of the Kola Peninsula, in north-western Russia the pollution of the environment is enourmous. Nickel mining, roasting and smelting lasting over a period of more than 60 years have brought about severe damage to the ecosystems in the surroundings of the three major industrial sites Nikel (nickel smelting), Zapoljarnij (nickel mining and roasting) and Monchegorsk (nickel smelting). Further emission sources in the area include iron-ore mines and mills, a large open-cast apatite mine and processing plant, several coal-fired power plants, an aluminum smelter and the cities and harbours of Murmansk and Severomorsk. In contrast, large parts of the Kola Peninsula and the neighbouring regions of Finland and Norway represent some of Europe's most pristine wilderness areas. This contrast, provided that the choice of elements is appropriate, should become very clear with the help of cluster analysis.

The choice of elements is of crucial importance here. Elements which are not suitable for

discerning environmental pollution, bring undesired influences (e.g. from weather conditions) into the clustering. It is therefore very difficult to separate and interpret these various kinds of influences in the clustering result. Masking variables, i.e. variables which show no or another group stucture, can also make an interpretation of the results impossible.

The chemical elements Co, Cu, and Ni are typical elements reflecting pollution. The selection of these elements and application of the algorithm CLARA with 4 clusters leads to the result presented in Figure 5.1. Objects (samples) belonging to particular cluster are in black. Since the interpretation of the resulting clusters is also of interest, we additionally present Figure 5.2, which shows the mean values of the elements in the four clusters. The dashed line has no meaning, but it is helpful when comparing different results. The most serious pollution (highest values of Co, Cu, Ni) is visible in 5.2 for cluster 4. Figure 5.1 shows that cluster 4 consists of samples around Nikel, Zapoljarnij and Monchegorsk. Cluster 3 shows strong pollution in the form of a ring surrounding cluster 4. The difference between East and West (the western area is lightly polluted, the eastern area is strongly polluted) is discernable in cluster 2. The influence of Co, Cu and Ni, shown in Figure 5.2, on cluster 2 is very low. Cluster 1 shows samples without cluster 3 and 4. In Figure 5.2 also the cluster size is printed, which is the fraction of samples in the cluster on all samples.

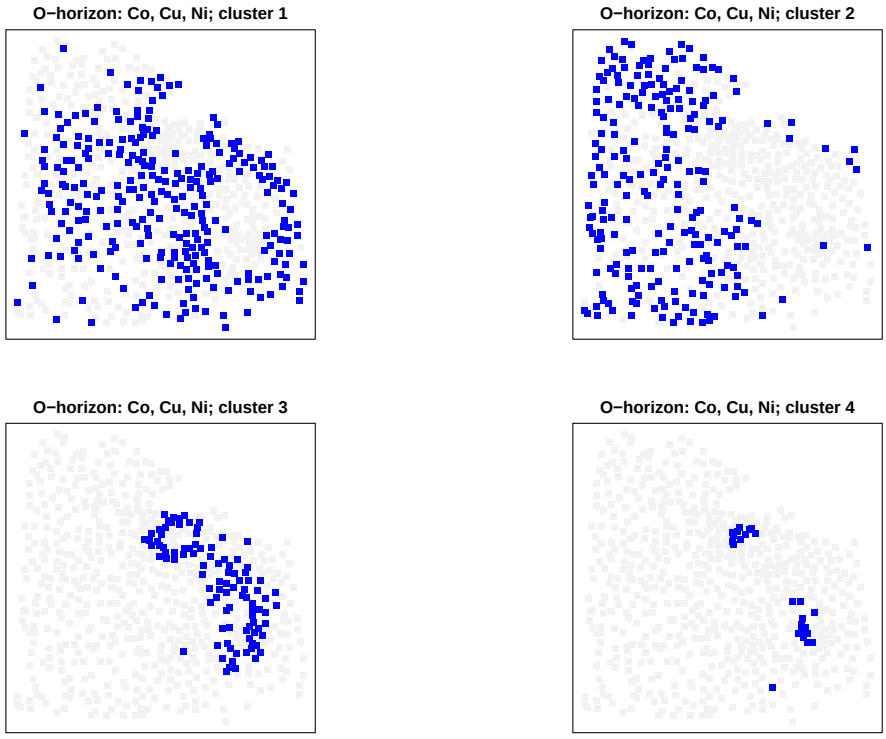


Figure 5.1: Elements Co, Cu, Ni from the O-horizon clustered with CLARA

A more detailed insight is provided by increasing the number of clusters to 8. In Figure 5.3 five levels of pollution can be recognised. Cluster 7 shows the strongest pollution in

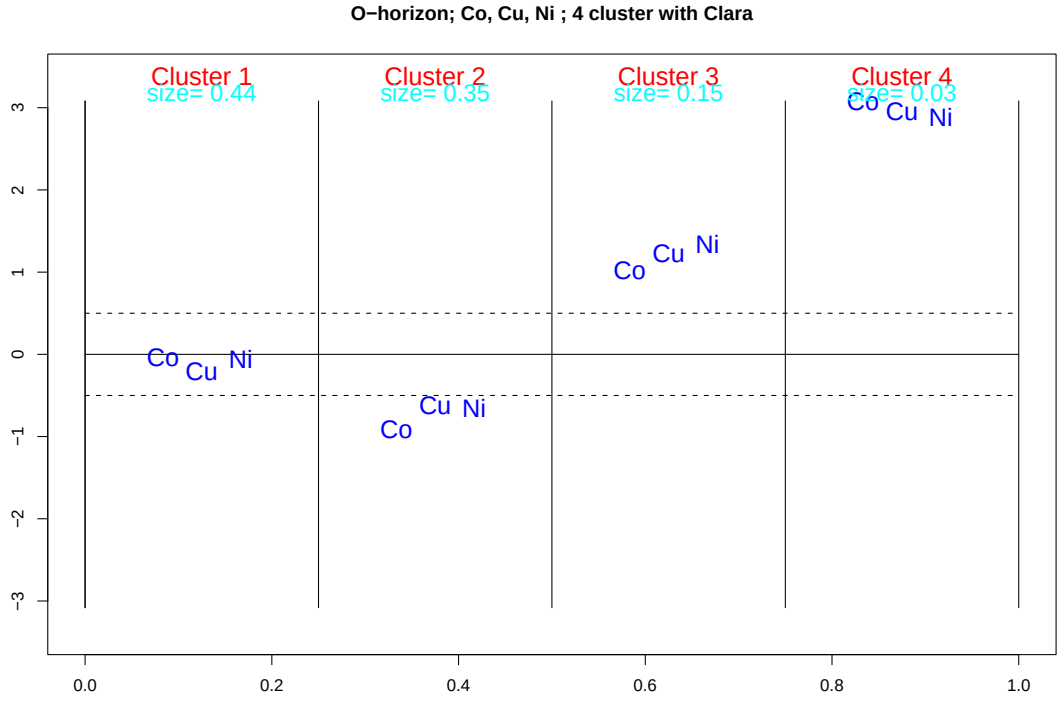


Figure 5.2: Influence of the elements Co, Cu, Ni from the O-horizon clustered with CLARA

the region of Nikel, Zapoljarnij and Monchegorsk. Cluster 8, 4, 2 and 1 show decreasing pollution in the form of a ring around cluster 7. In all other clusters the difference between East and West is visible.

It will be of interest now to compare the previous results with fuzzy clustering. We apply the GK algorithm for six clusters. The resulting memberships coefficients include a lot of new information: In the previous analysis each object was clearly assigned to one cluster (0/1 memberships), whereas now we receive memberships of each object to all clusters. These coefficients can be visualised by using a grey scale: the higher the membership coefficient of a sample in a cluster, the darker the point in the map. Figure 5.5 shows the resulting maps, each map represents one cluster. The East-West difference is also visible in Figure 5.5 in clusters 1 and 3. The pollution can be seen in cluster 6 and the outer ring of the pollution is visible in cluster 4.

The clustering of the elements As, Co, Cu, Ni and V with the help of CLARA is shown in Figure 5.6. In cluster 3, 5 and 6 the pollution is again clearly visible. However, the difference between East and West is not visible anymore. Due to the larger number of elements the interpretation of the results has become more difficult.

In Figure 5.7 the results are presented in a different way. The first illustration shows the result of CLARA with two clusters, the black cluster consisting of 448 data points and the red one consisting of 169 data points. In each of the following four illustrations the number of clusters has been raised by one. This way of presenting the results is not as clear as the

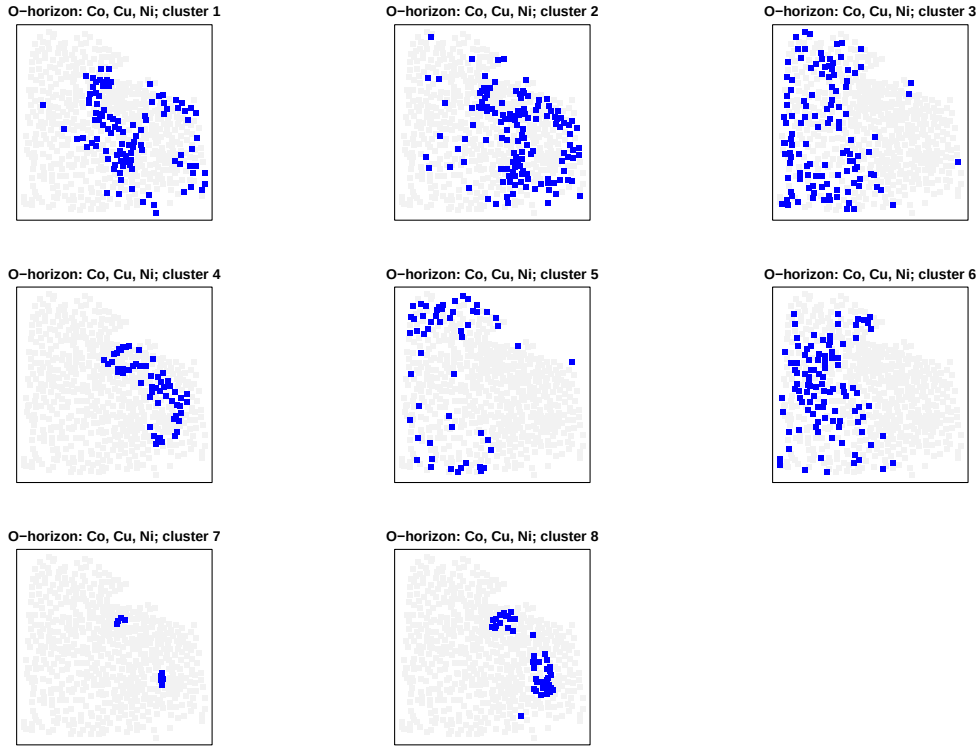


Figure 5.3: Elements Co, Cu, Ni from the O-horizon clustered with CLARA

former plots, but it gives information on how the clusters split and change when its number is raised.
The pollution of the red cluster from the first illustration disperses into smaller segments in the following illustrations, while the black points from illustration no. 1 disperse in a seemingly arbitrary manner into unseparated clusters.

The clustering of the elements Ca, Fe, Mg, Na and Sr should make the influence of the sea visible in Figure 5.8. Cluster 3 shows the strongest influence of the sea, with Mg and Na having the strongest influence here (see Figure 5.9). In cluster 6 the influence of Ca, Mg, Na and Sr is strong, and it illustrates the second belt of the sea spray. Cluster 1, 4 and 7 show a distinction between the area influenced by sea spray and the interior. Cluster 9 is roughly similar to the univariate map of the high radiotoxic Strontium.

The clustering of the elements Co, Cu, Mn, Na, Ni, Pb and Sr in Figure 5.10 is not so easy to interpret. The sea spray is clearly visible in cluster 2, with Na and Sr having the strongest impact (see Figure 5.11). In cluster 1 one can still see the influence of the sea, while cluster 4 makes not only the influence of the sea visible but it also mirrors the pollution by Co and Cu. In cluster 3 the differences between East and West and between the area influenced by the sea and the interior can not be clearly discerned anymore, and it is difficult to say which of the two factors (sea spray or pollution) has the bigger influence. Cluster 5 approximately mirrors the univariate map of Manganese with Na, for example, having very little influence. Therefore this cluster can be interpreted as a counterpart to sea spray (distinction of sea

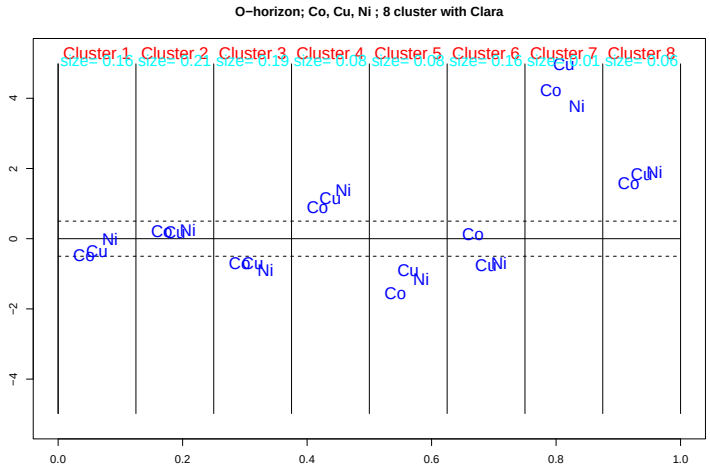


Figure 5.4: Influence of the elements Co, Cu, Ni from the O-horizon clustered with CLARA

spray - interior). Cluster 6 shows a strong lead pollution in certain points. Cluster 7 is the cluster which is more or less left over when all samples which showed an influence of sea spray or pollution have been crossed out. In cluster 8 the distinction between sea spray and the interior is again visible. The pollution becomes clear in cluster 9 and 10 with cluster 9 mainly showing the pollution by Strontium and with cluster 10 showing the pollution by Co, Cu and Ni. Cluster 11 and 12 are very difficult to interpret, since both the sea spray and the pollution have their impact here.

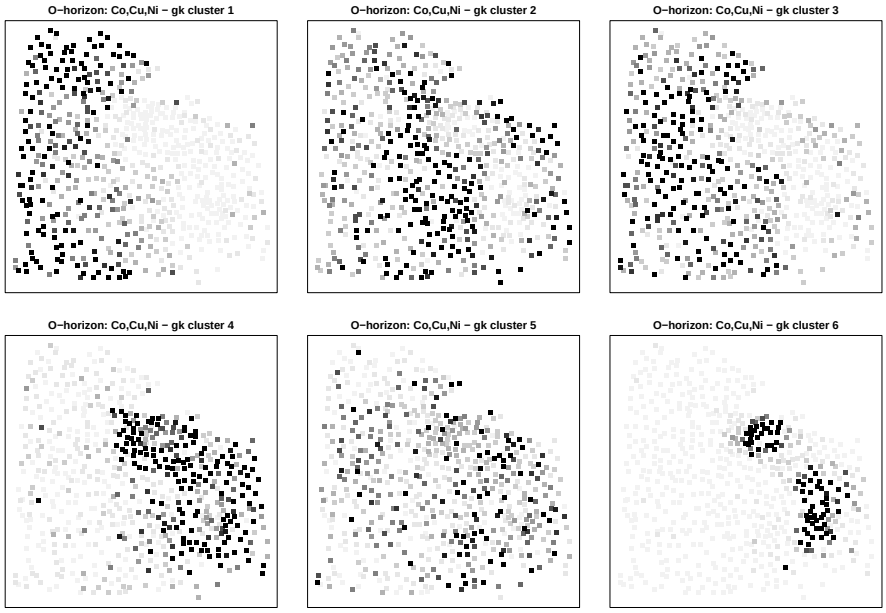


Figure 5.5: Elements Co, Cu, Ni from the O-horizon clustered with the GK algorithm

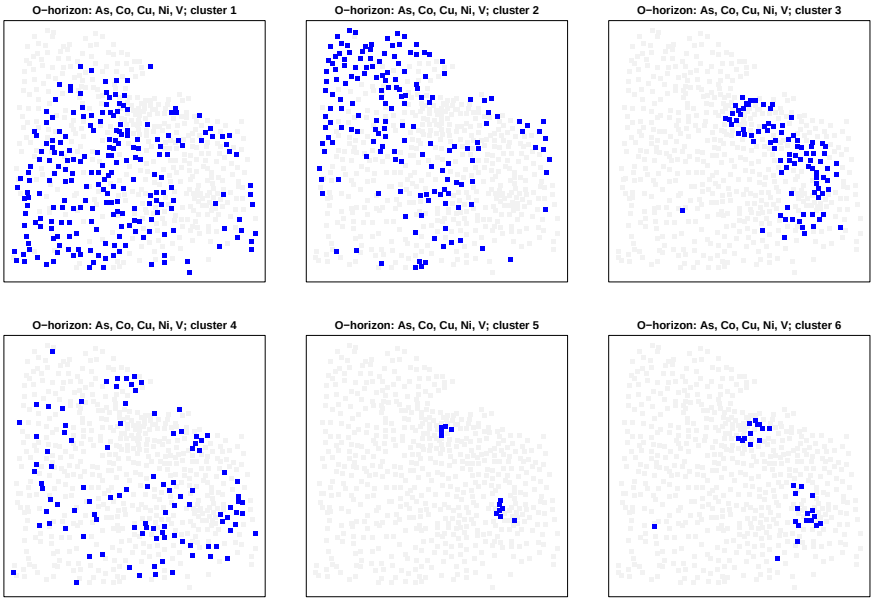


Figure 5.6: Elements As, Co, Cu, Ni, V from the O-horizon clustered with CLARA

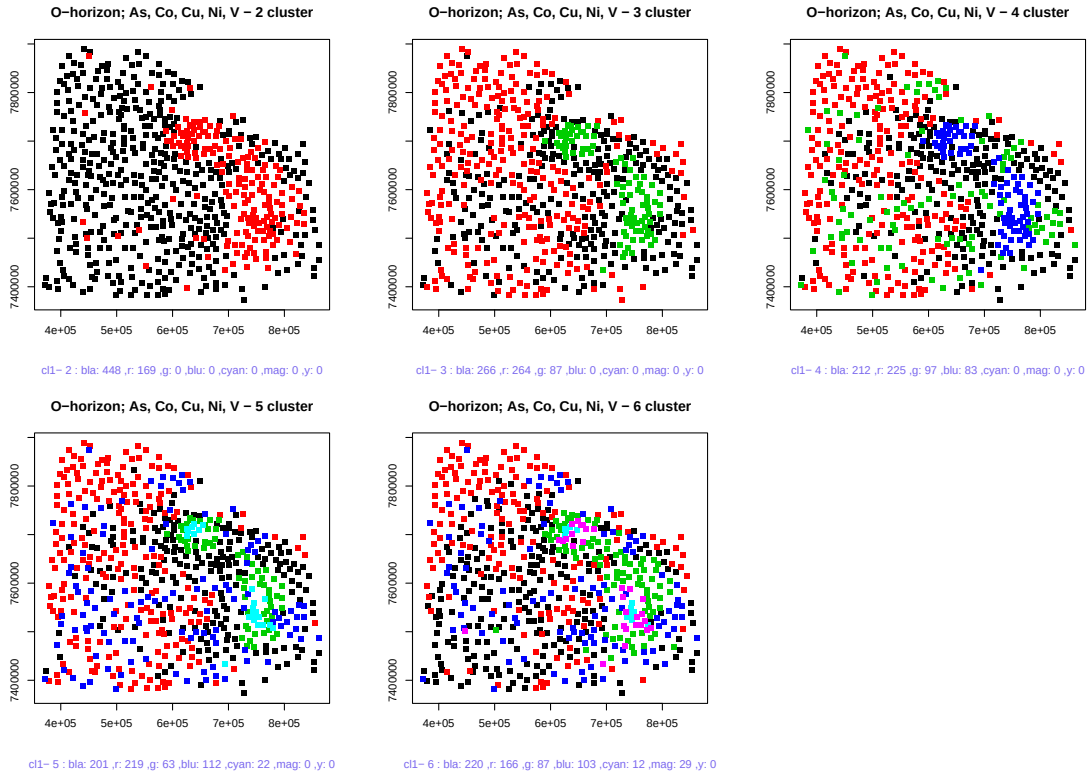


Figure 5.7: Elements As, Co, Cu, Ni, V from the O-horizon clustered with CLARA

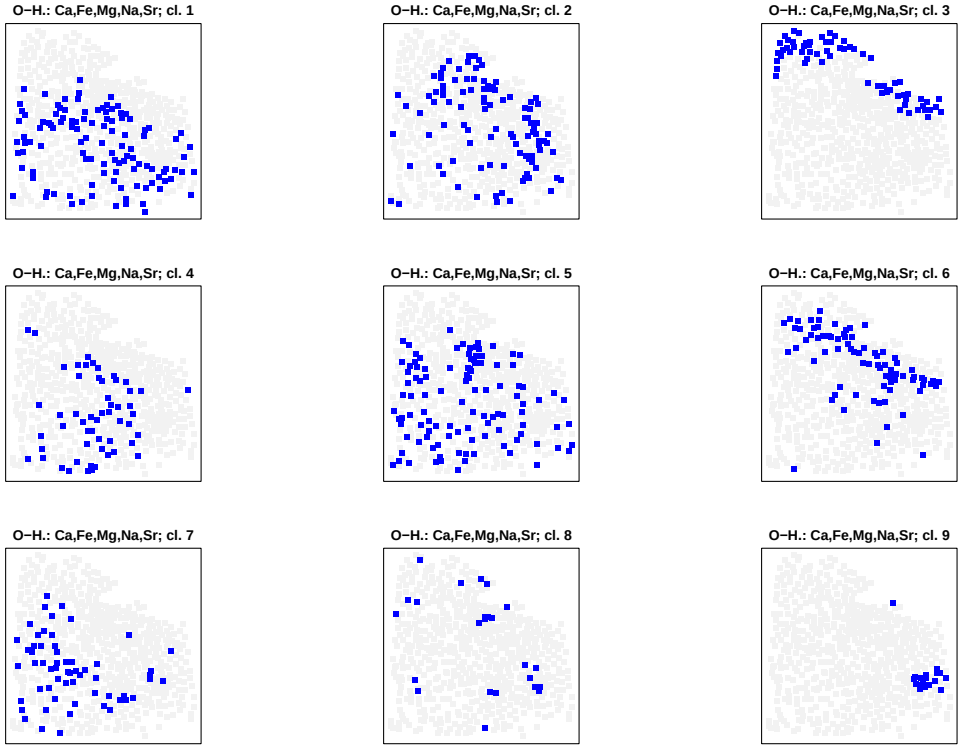


Figure 5.8: Elements Ca, Fe, Mg, Na, Sr from the O-horizon clustered with CLARA

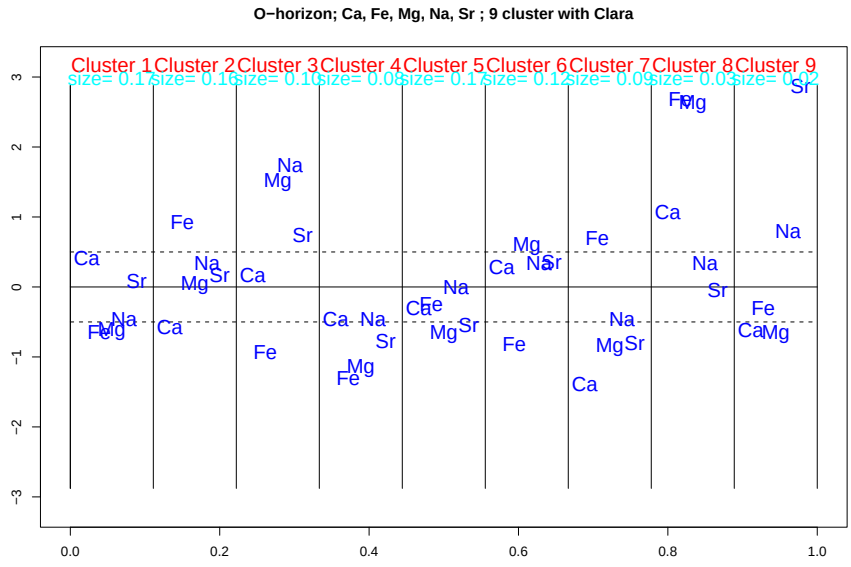


Figure 5.9: Influence of the elements Ca, Fe, Mg, Na, Sr from the O-horizon clustered with CLARA

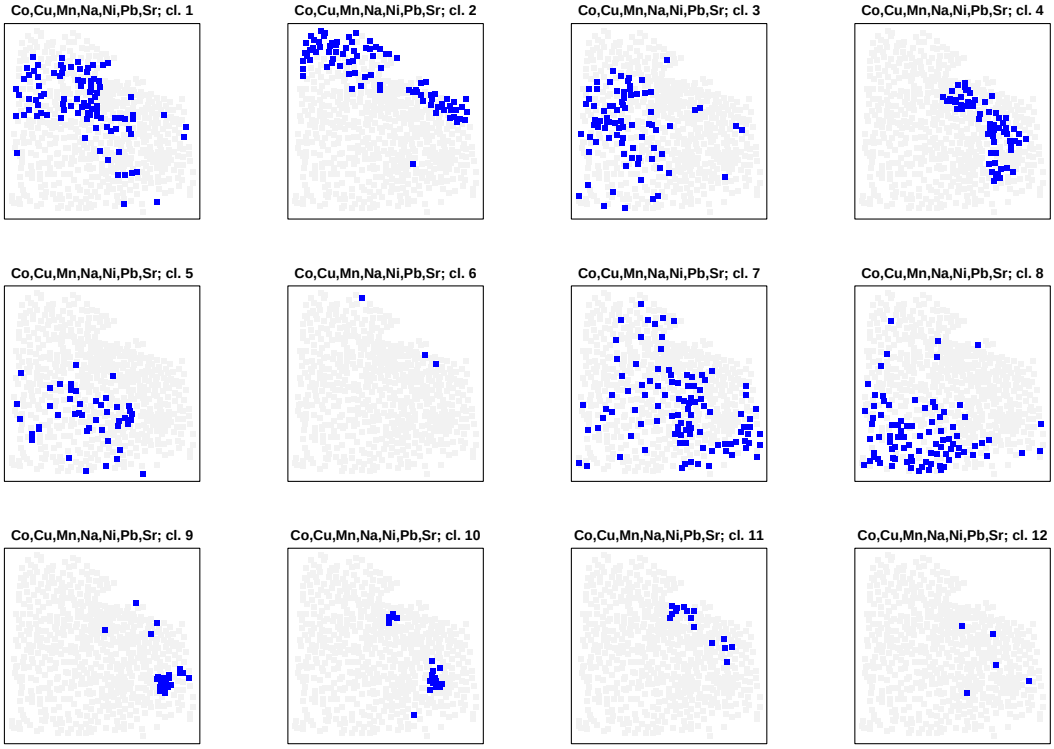


Figure 5.10: Elements Co, Cu, Mn, Na, Ni, Pb, Sr from the O-horizon clustered with CLARA

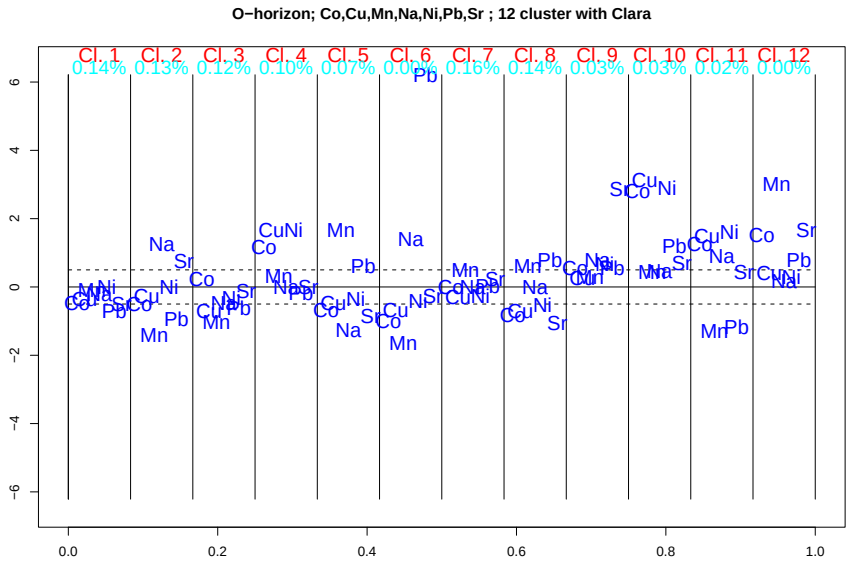


Figure 5.11: Influence of the elements Co, Cu, Mn, Na, Ni, Pb, Sr from the O-horizon clustered with CLARA

5.2 C-horizon

The samples of the C-horizon were taken from a layer 30 cm under the surface. Therefore environmental influence should be small. After having clustered some of the elements from the C-horizon the types of rock should be visible. Figure 3.4 shows a separation into groups of rocks. The clustering should not only contribute to clarity by putting together the information at several univariate maps, but it should raise the question if the clustering mirrors the lithological map or not.

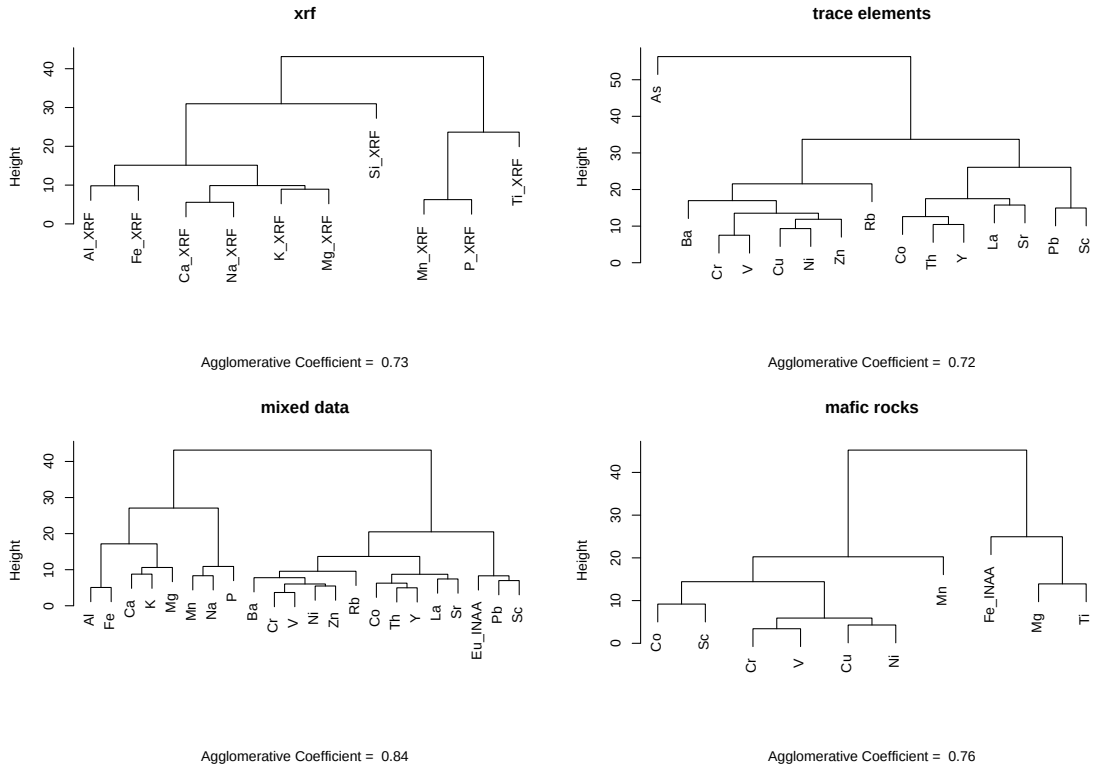


Figure 5.12: Clustering of elements from the C-horizon with AGNES

The variable cluster produced with the help of hierarchical clustering is presented in Figure 5.12. It shows the similarities of the elements in each of four selections of elements (main elements, trace elements, mixed elements and mafic rock elements). In the hierarchical clustering produced by AGNES and the *average method*, one can detect from the xrf data (main elements of the C-horizon) both the outliers Si and Ti and define Al-Fe, Ca-Na, K-Mg and Mn-P as very similar elements in the sense of this clustering. Later we will see that, if we cluster the data of the main elements without these outliers, we can reach better results. If one element of each of the similar pairs is crossed out and clustered with Si and Ti the results will be even better.

Figure 5.13 is the result of fuzzy clustering produced with the Gustafson-Kessel algorithm. The memberships to cluster 1 visualised in the first illustration clearly show the sedimentary rocks and the supracrustal complexes in the North together with the alkalines in the East.

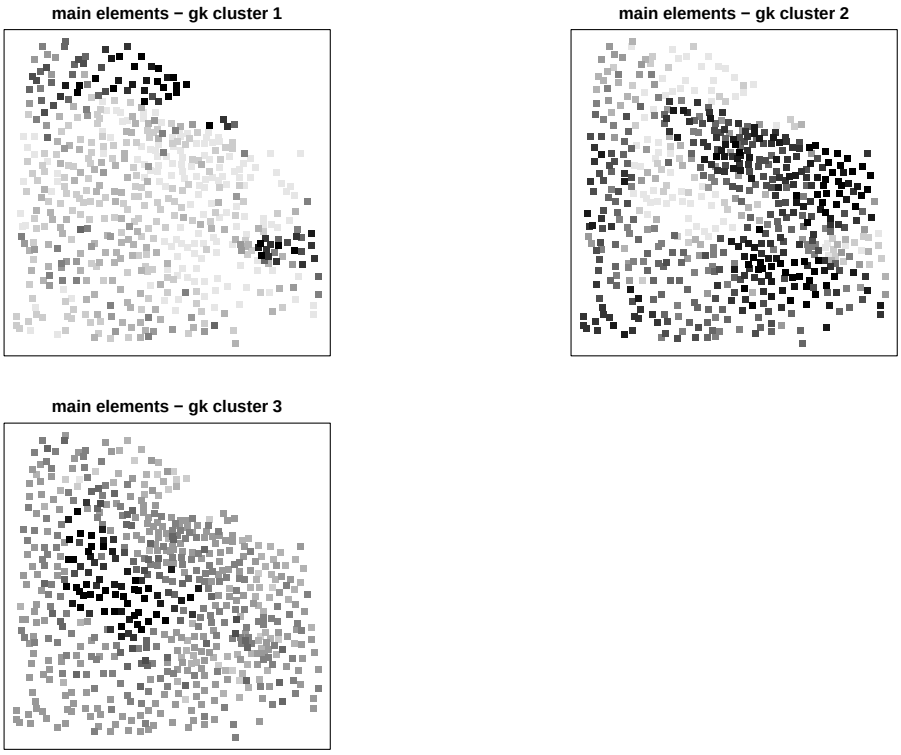


Figure 5.13: Main elements of the C-horizon clustered with the Gustafson-Kessel algorithm

The elements K and Ti have the biggest influence in this cluster (see Figure 5.14). The alkaline intrusions and the sediments should differ from each other, but geochemically they are very similar (in the clustering of the main elements without Si and Ti the alkaline intrusions disintegrate into two clusters). Cluster 2 shows a clear regional demarcation. The granulite belt is clearly visible in cluster 3. Basically, more clusters can be produced by the FCM and the GK algorithm by putting together similar elements. Eight different clusters can be produced from the elements Al, Ca, K, Mn, Si and Ti with the help of the FCM algorithm, and they are shown in Figure 5.15. Cluster 3 shows group no. 1 and no. 4 of the lithological map in Figure 3.4. The geology 31 (granulite belt) and the geologies 9 and 10 (sedimentaries) fall into one cluster. Cluster 8 shows a typical half-ring made by the geology 7 (granite) near Nikel and Zapoljarnij and geology 1 (felsic gneiss) in the West.

The clustering of the data into six clusters with the help of the algorithm CLARA is shown in Figure 5.16. In cluster 2 geology 1 in the south and geology 4 in the North-East form one group. One can observe this kind of constellation again and again. Cluster 3 mainly contains geology 52 (see lithological map in Figure 3.4. Cluster 4 shows the typical half-ring consisting of geology 7 (compare Reimann et al., 2000) and the alkaline intrusions of geology 82 south-east of Olenogorsk. In cluster 5 the geologies 31, 9 and 10 fall once again into one group. The elements potassium and silicon (see Figure 5.17) are of prime importance for cluster 6. Most of geology 7 (granite) is reflected here. Cluster 1 contains the points which are not to be found in any of the other clusters.

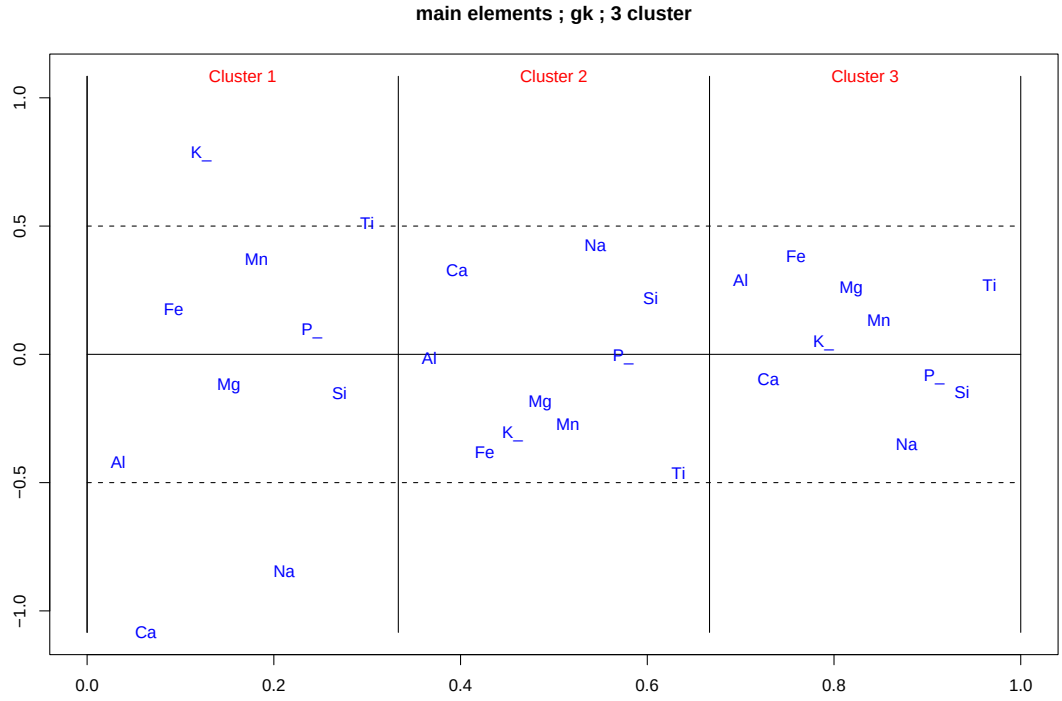


Figure 5.14: Influence of the main elements of the C-horizon clustered with the Gustafson-Kessel algorithm

Clustering of the elements Al, Ca, K, Mn, Si and Ti (variable selection of the main elements) produces clusters which are easier to discern (see Figure 5.18) than the ones with the main elements in Figure 5.16. Cluster 2 and cluster 4 swell into bigger groups. Unexpectedly, cluster 6 shows several lineal rock formations together with a peculiar line underneath of the Varanger Peninsula which contradicts the lithological map in which no lineal arrangement of geology 7 is visible.

When clustering the first seven principal components (see Section 1.4) of the trace elements (Figure 5.20) with the help of CLARA the geologies 9 and 10 are visible in cluster 4. Cluster 5 shows geology 82 and 7 in the East. Cluster 5 consists all in all of geology 7, while the peculiar line underneath of the Varanger Peninsula, which is not visible in the lithological map, is visible in this cluster.

Figure 5.21 shows the result of the clustering of the selected mixed elements with the help of CLARA. In cluster 1 La and P are most influential (see Figure 5.22). Cluster 2 generally shows geology 7 with the elements Rb, Ca, P and Th having the largest influence in this cluster. In cluster 3 Ca has the largest influence. For this reason this cluster is very similar to the univariate map of Ca. In Cluster 4 nearly all elements have a very large influence. This cluster shows the alkaline intrusions in the East with the sediments in the North-West. Geology 51 in the South is also visible in this cluster. Cluster 5 gives a very good visualisation of the granulite belt (geology 31) together with geology 9 in the North. It is

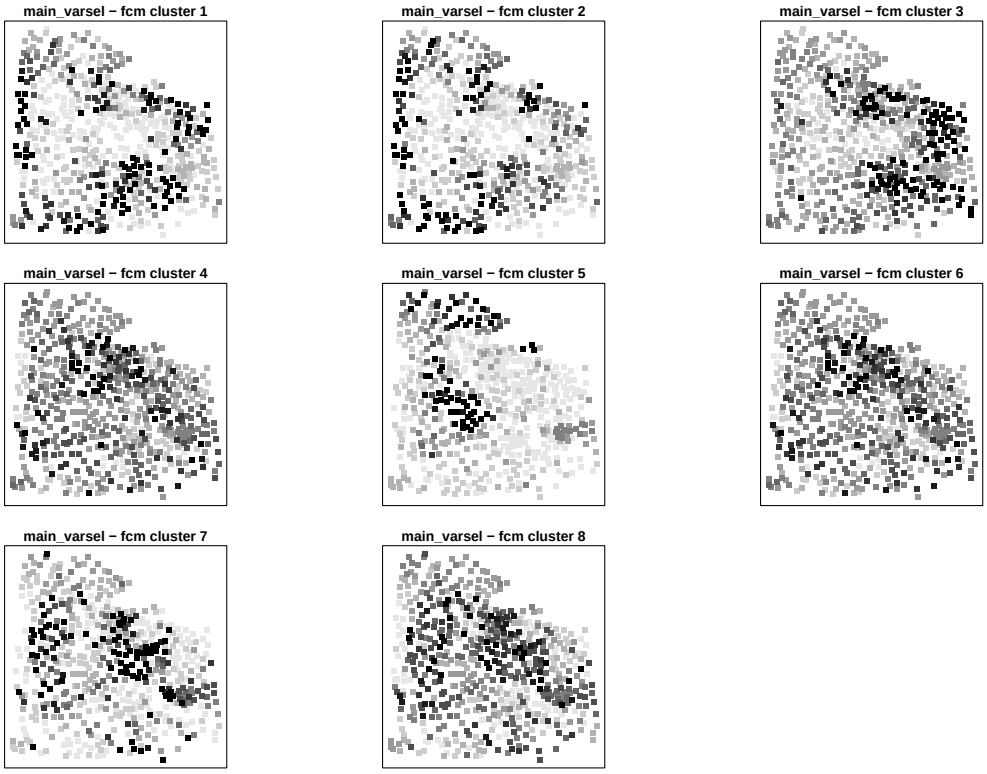


Figure 5.15: Selection of the main elements of the C-horizon clustered with the FCM algorithm

interesting to note that the geology 9 on the Rhybachi Peninsula together with the geology 9 on the Varanger Peninsula is not visible. Geology 9 on the Rhybachi Peninsula is to be found in cluster 4. The elements Ca, P and Rb have the largest influence in cluster 6.

In Figure 5.23 cluster 6 displays an interesting division between the sediments in the North. Here geology 9 forms a clear group without geology 10. Cluster 4 basically shows geology 52.

The fuzzy clustering of the trace elements and the mixed elements produces similar groups. The alkaline intrusions in the East fall into one cluster when the GK algorithm is applied. Furthermore, a half-ring formed by geology 7 near Nikel and Zapoljarnij which is always visible when the FCM algorithm is applied. Generally speaking, clustering the trace elements and the mixed elements is more difficult to interpret than the main elements.

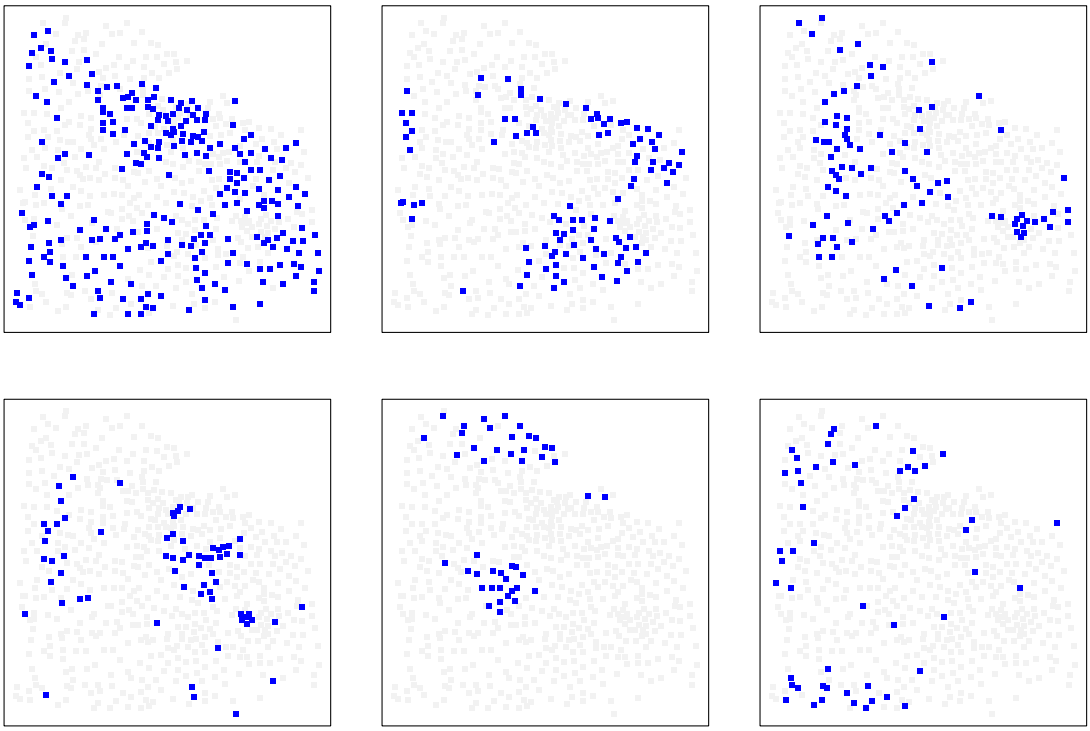


Figure 5.16: Main elements of the C-horizon clustered with CLARA

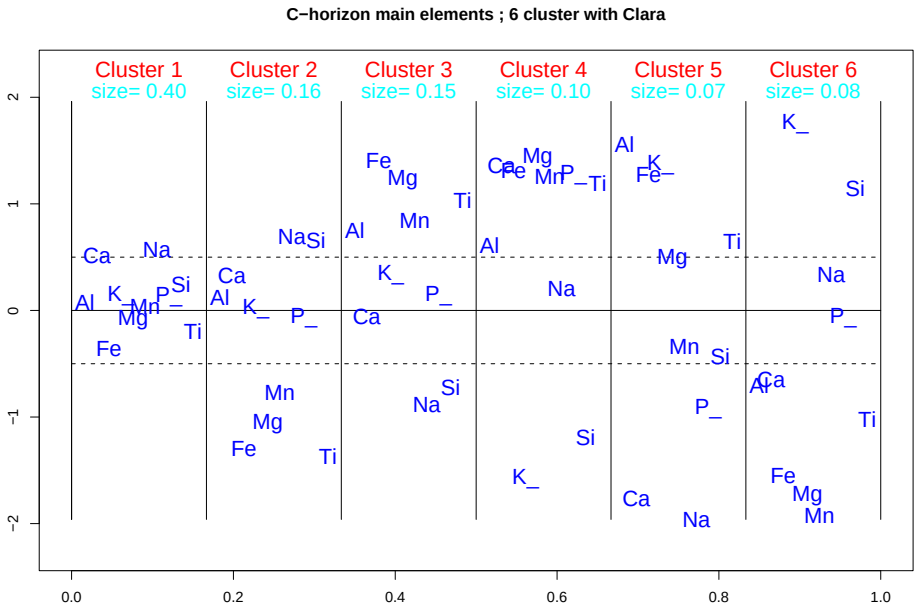


Figure 5.17: Influence of the main elements of the C-horizon clustered with CLARA

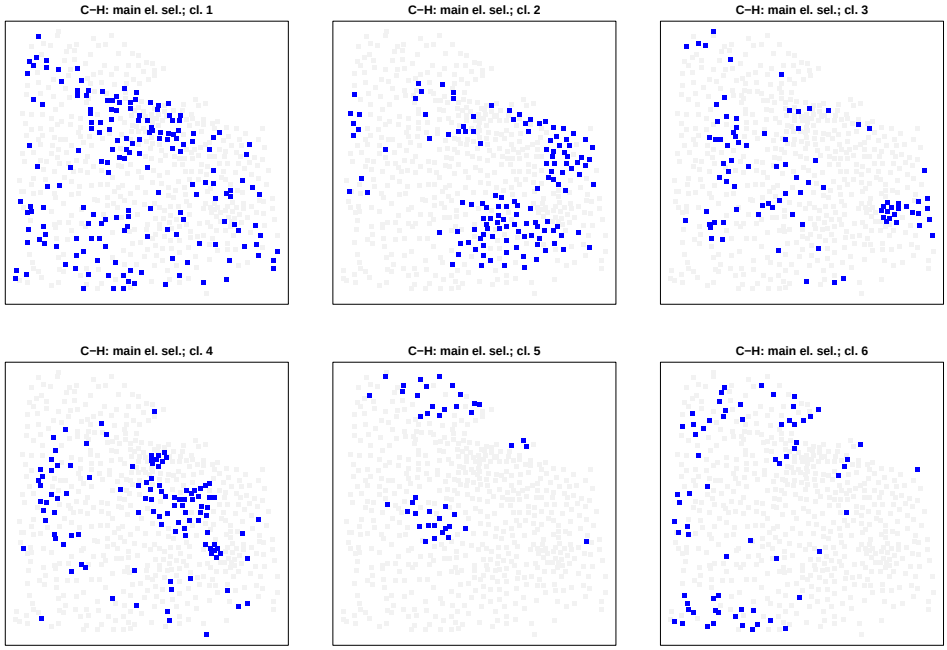


Figure 5.18: Selection of the main elements of the C-horizon clustered with CLARA

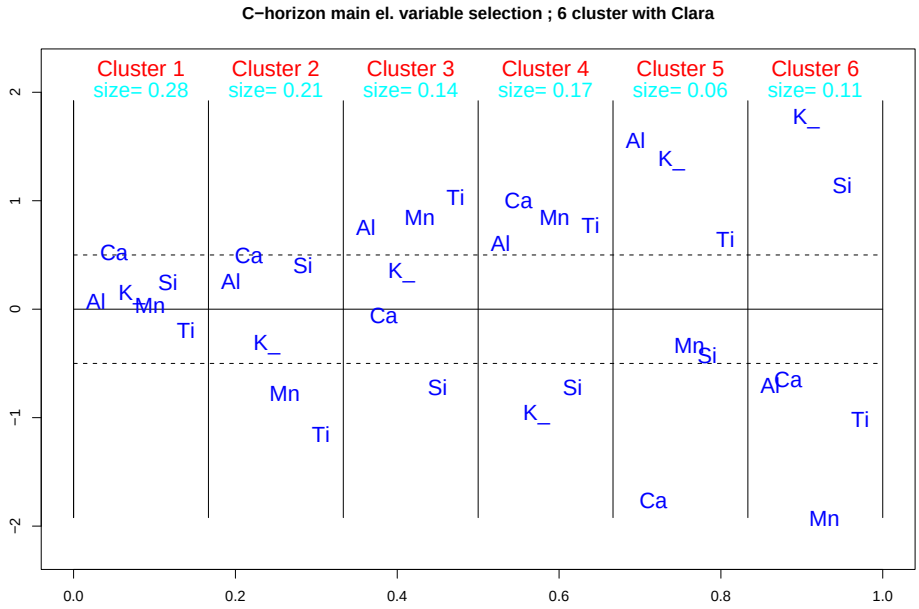


Figure 5.19: Influence of the selected main elements of the C-horizon clustered with CLARA

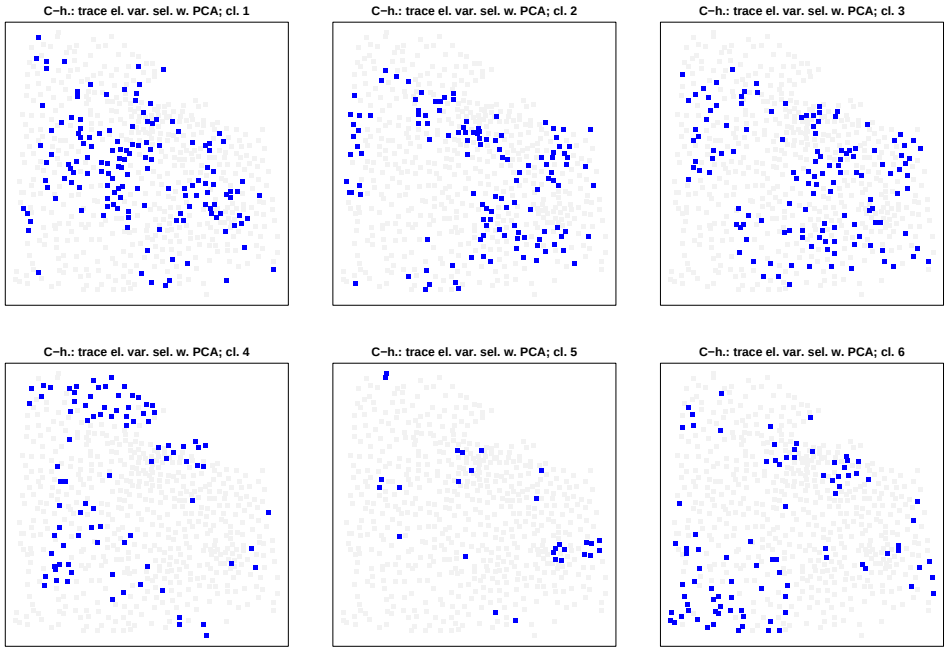


Figure 5.20: First seven principal components of the selection of the variables of the trace elements of the C-horizon clustered with CLARA

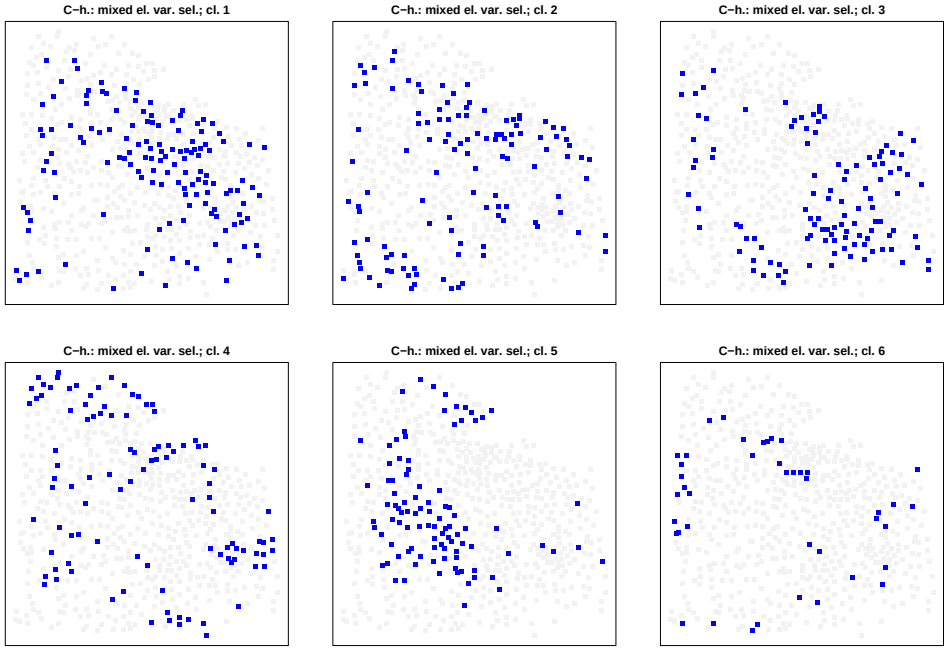


Figure 5.21: Selection of the mixed elements of the C-horizon clustered with CLARA

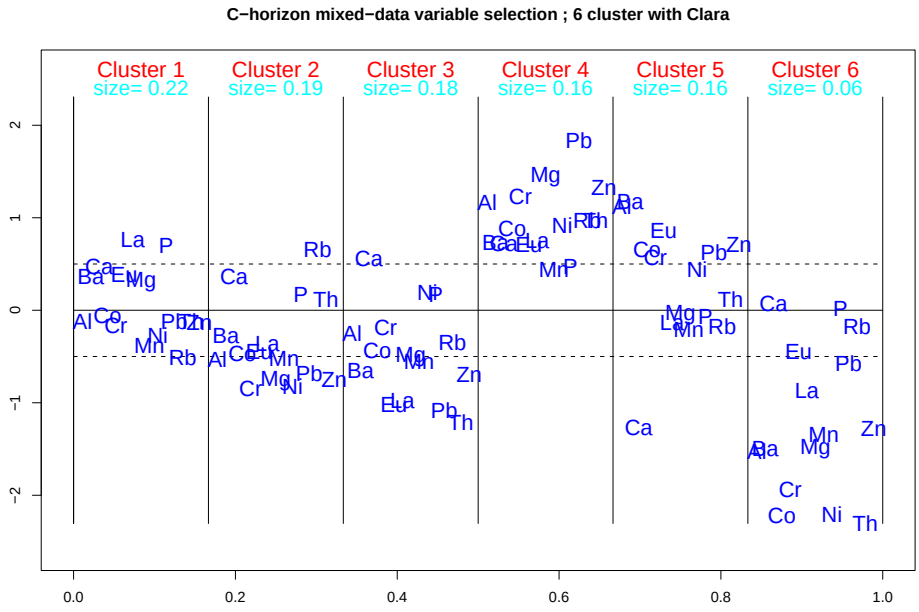


Figure 5.22: Influence of the selected mixed elements of the C-horizon clustered with CLARA

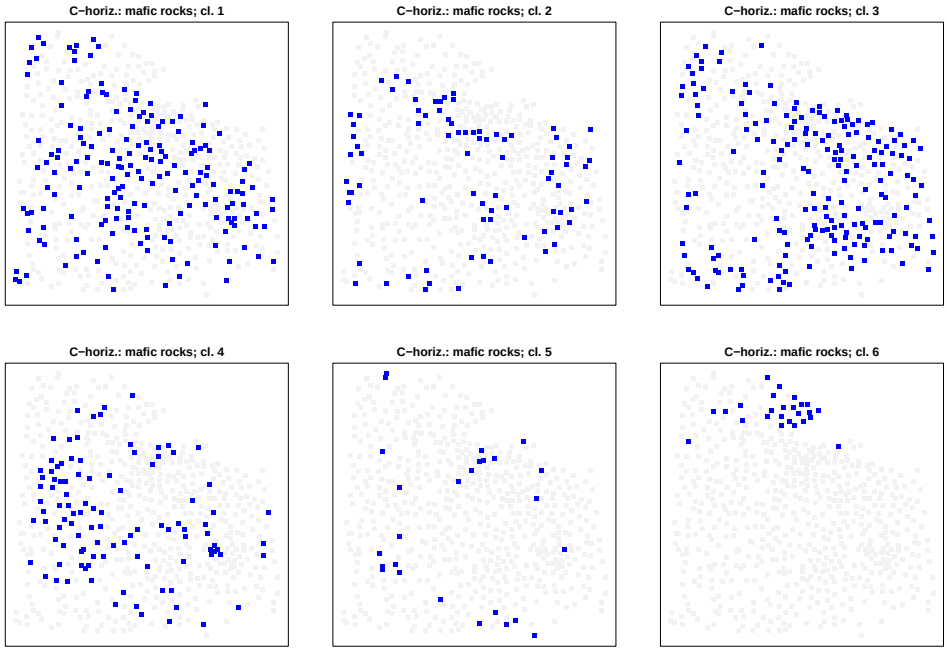


Figure 5.23: Mafic rock elements of the C-horizon clustered with CLARA

5.3 Walchen Data

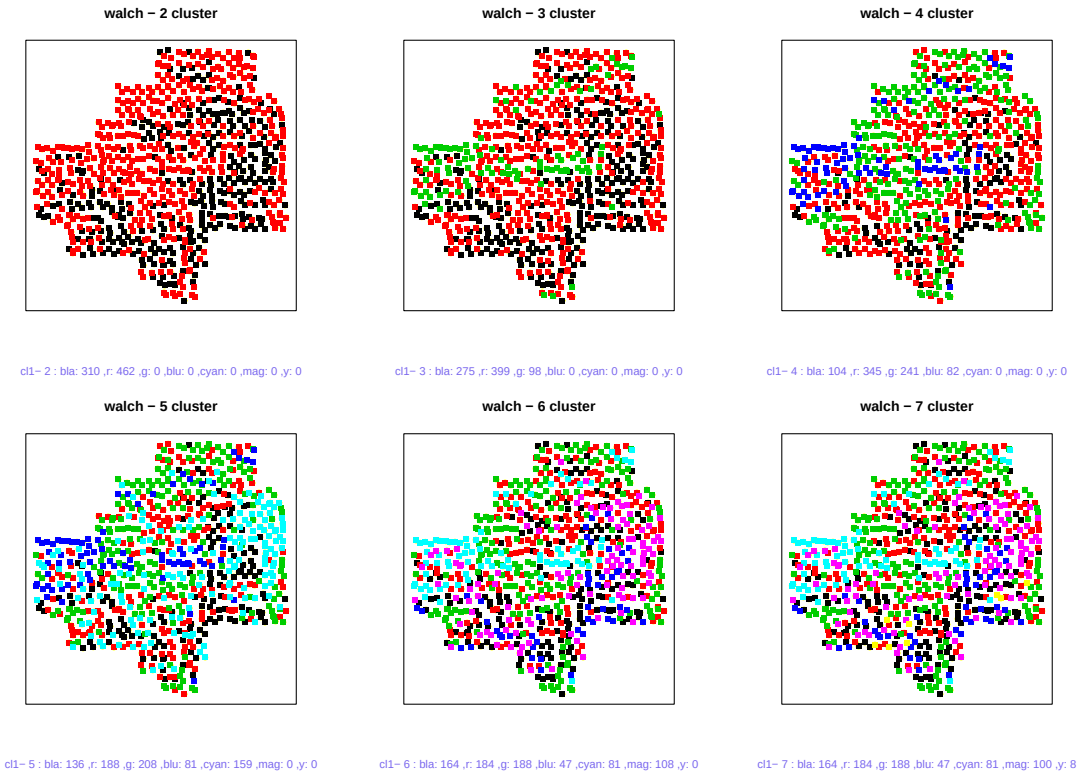


Figure 5.24: Walchen data clustered with PAM

Figure 5.24 shows the Walchen data clustered with the help of the PAM algorithm. The number of clusters was varied from 2 to 7, the resulting pictures are printed in two rows in Figure 5.24, and referred to in the text as pictures no. 1 to 6. In the red cluster (cluster 2) in picture no. 1 many elements have high influence (see Figure 5.25). This red cluster falls apart in picture no. 2. The green cluster (cluster 3) in picture no. 2 shows very clearly the green rocks. This cluster consists of the ore deposits in the centre of the map and of the green rocks in the West and in the North. The geologist cannot say exactly where the demarcation line between the Ennstaler phyllites in the North and the Wölzer mica schist in the centre and in the South lies. This distinction is just more or less visible in the North of the green cluster in the middle of the map. The green cluster of picture no. 2 is almost completely visible in picture no. 3 as a blue cluster. The black-marked samples of cluster 1 have the smallest average element concentrations. The adjoining red cluster (cluster 2) has also quite low concentrations in general. In picture no. 4 the blue points (cluster 4) stand for a big influence, the red ones (cluster 2) for a medium influence and the black ones (cluster 1) together with the magenta-coloured (cluster 5) samples stand for a small influence (5.26). At the same time the influence of the elements in the black cluster and in the cluster coloured in magenta is of a contrary character. This means that in the black cluster the influence of Co, Cu, Mg and Ni almost completely disappears while Ba, Cr, Na and Sr still display an influence, and in the magenta-coloured cluster there is clearly opposite relation

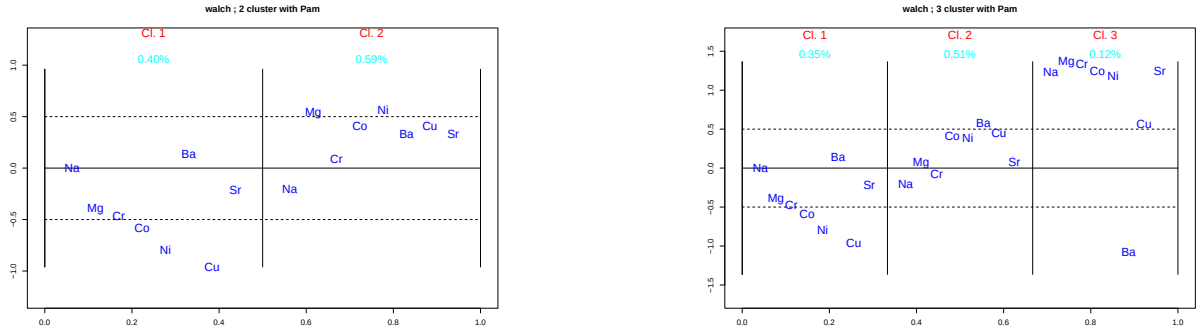


Figure 5.25: Walchen data clustered with PAM; influence of the elements

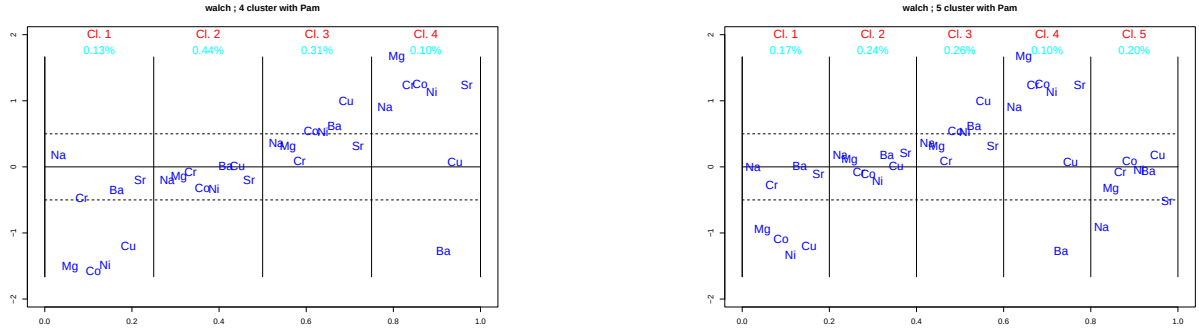


Figure 5.26: Walchen data clustered with PAM; influence of the elements

in the influence of these elements. In picture no. 5 and 6 the ore deposits in the centre are visible together with the green rocks in the West (this time marked in magenta colour). In Figure 5.24 one could also see that the green rocks, especially the ore deposits in the centre together with the green rocks in the West form a very homogeneous cluster, because this cluster did not really fall apart even after raising the number of clusters.

Figure 5.28 presents the results after application of the algorithm CLARA. The number of clusters was fixed with 7, and the results are almost the same as before (Figure 5.24). Cluster 1, 2, 6 and especially cluster 4 have low average concentrations of the elements (see Figure 5.29). In cluster 2 and 4 one may see the distinction between the Ennstaler phylittes and the Wölzer mica schist. Cluster 4 shows again the green rocks, and cluster 7 shows rich magnesium deposits.

In Figure 5.30 and 5.31 we can once again compare the results of the fuzzy clustering with the help of the FCM algorithm and the GK algorithm. By doing clustering with the FCM algorithm four individuals clusters could be created. However, with the GK algorithm only three clusters could be created. The green rocks are more visible when the clustering was done with the GK algorithm than with the FCM algorithm (see Figure 5.30 picture no. 3 and Figure 5.31 picture no. 4). The samples, however, which are little influenced by the elements are more clearly visible when they were clustered with the FCM algorithm (see Figure 5.30 picture no. 1 and 3).

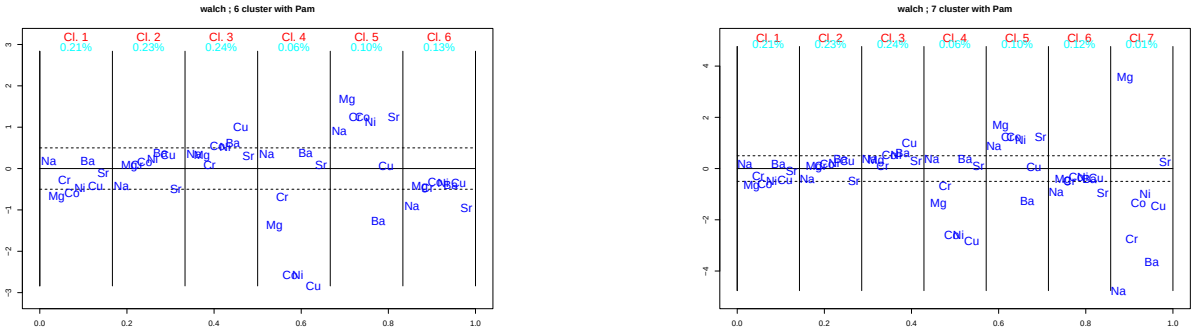


Figure 5.27: Walchen data clustered with PAM; influence of the elements

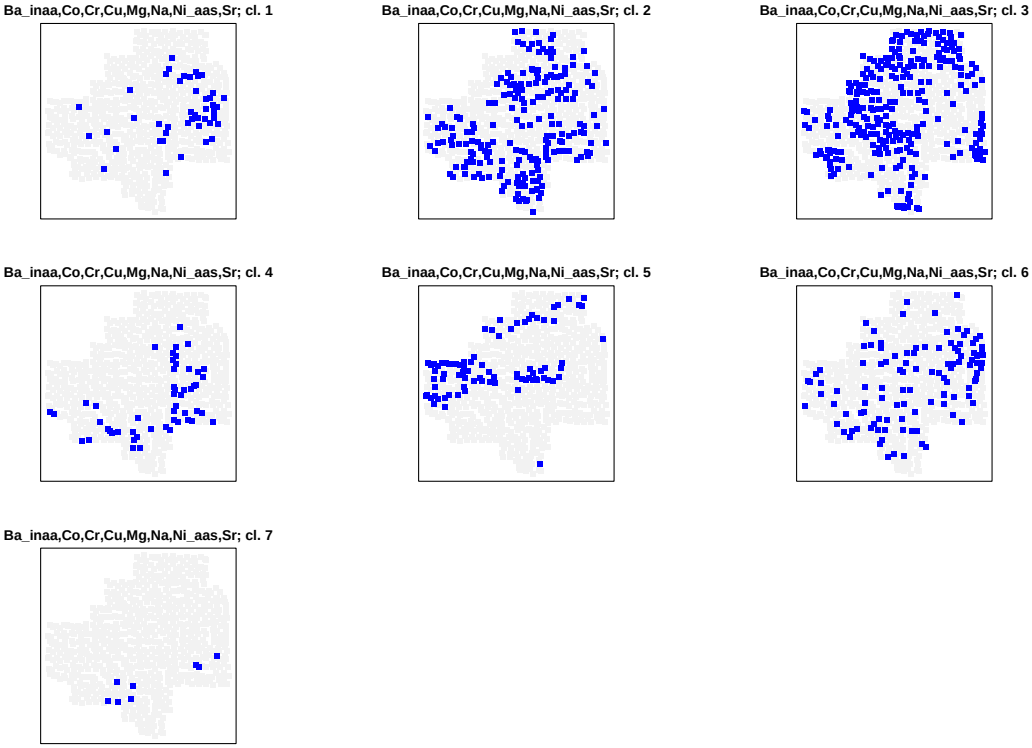


Figure 5.28: Walchen data clustered with CLARA

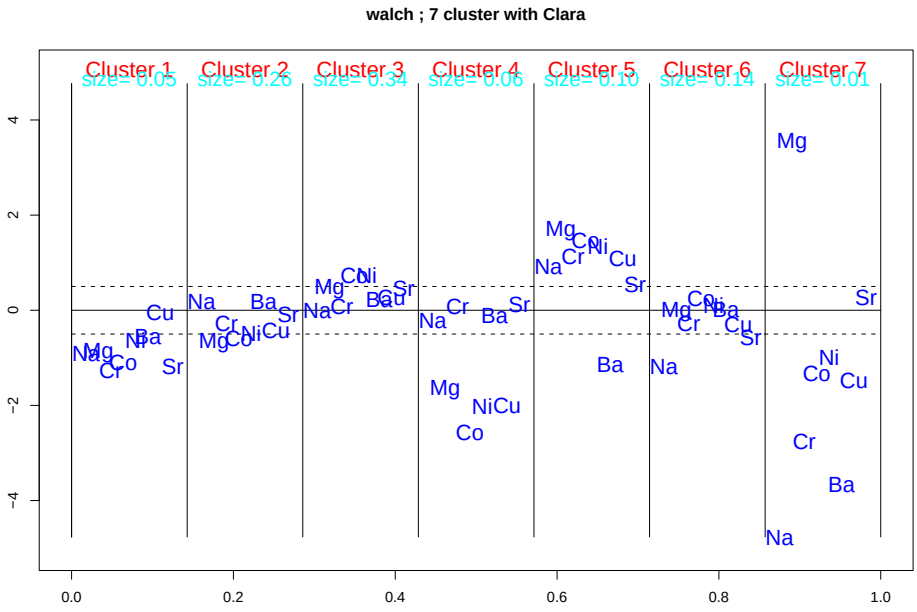


Figure 5.29: Walchen data clustered with CLARA; influence of the elements

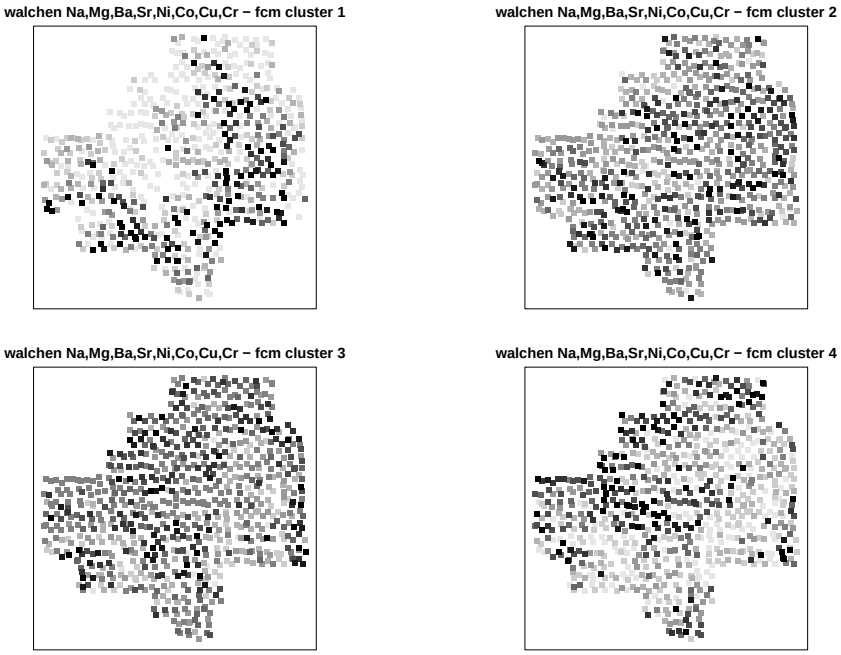


Figure 5.30: Walchen data clustered with the FCM algorithm

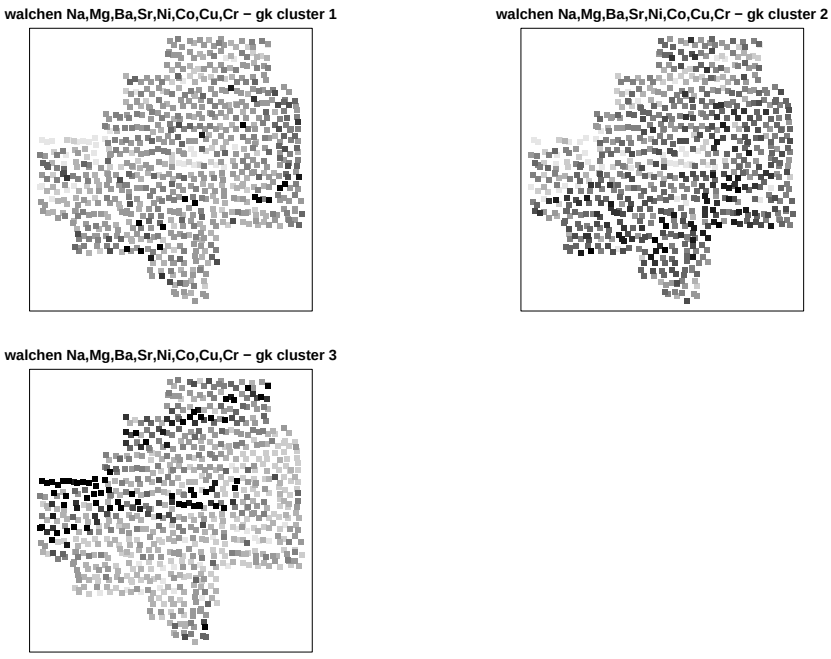


Figure 5.31: Walchen data clustered with the GK algorithm

5.4 Results of the Validity Measures for Fuzzy Clustering

The following charts should give more information about the distribution of the data. The quality criteria described in Section 4.4.2 also provide further information concerning the question which of the selections of the elements lead to the best cluster quality, which of the algorithms (FCM, GK or GG) is the best one for each of the clustering groups of elements and which number of clusters is optimal.

The values of the various validity measures were evaluated by the fuzzy clustering package of Höppner (2000) for a certain number of clusters and for the assignment of the objects with several algorithms. This particular number of clusters was defined in the following way: Beginning with the assignment of the objects into two clusters, the number of clusters was increasing until a significant rise in one or more validity measures for these and for the next numbers of clusters was discernible. This significant rise (knee) is an indicator for the optimal number of clusters. n_c^{opt} was defined with the help of the pick tool with $(v_1, v_2, \ldots, v_6)^T$ being the values of the voting vectors (see Appendix B 2.2). $n_c^{(2)}$ is the second best choice for the number of clusters when the pick tool is applied. The number of clusters on the chart in bold print stands for the best choice in the number of clusters according to the criteria recommended in literature (see Chapter 4). If, for a particular problem, only the density of the cluster is of some importance then the viewer should look for a jump with the indizes APD and PD only.

In this thesis when doing fuzzy clustering some experiments were made with different fuzzifiers (see Section 2.3.1). Table 5.2 should show that a rise of the fuzzifiers results in a decline in quality of the clustering for all values (compare Table 5.1 with Table 5.2). The validity measures for the clustering of the main elements from the C-horizon with the GK algorithm (Table 5.3) at the number of two clusters behave in a similar way as with a clustering with the FCM algorithm (Table 5.1). The first jump occurs no sooner than at $n_c = 5$ when clustering with the FCM algorithm, while in Table 5.3 it is already visible at $n_c = 3$ for the validity measure separation. This means that when the main elements from the C-horizon are clustered with the FCM algorithm the optimal number of clusters is reached at $n_c = 5$. Furthermore, we can read in Table 5.1 that, because of the validity measures PC, S, APD and PD, the division in four clusters is better than the division into three clusters.

Table 5.4 also tells us that the optimal number of clusters for the main elements from the C-horizon clustered with the GG algorithm definitely lies at 5. $n_c = 6$ provides values for PE, S, CI and FHV significantly higher than the values at $n_c = 5$.

In Table 5.5 only the separation index(s) gives a hint that the trace elements are more suitable for clustering than the main elements. Here it is difficult to say what is the optimal number of clusters. On the one hand a small jump is visible at $n_c = 3$ and a second at $n_c = 4$ (both for separation), on the other hand the first big jump occurs at $n_c = 5$ (also for separation). With all the other validity measures no significant jumps are visible at all.

Table 5.6 shows that the trace elements of the C-horizon clustered with the GK algorithm clearly provide worse values than with the FCM algorithm (Table 5.5).

When comparing Tables 5.7, 5.5 and 5.1 we can see that the mixed elements of the C-horizon are less suitable for clustering (all values of the validity measures lie above the ones of the trace elements and the main elements). The optimal number of clusters for the mixed elements clustered with the FCM algorithm should lie at either $n_c = 3$ or $n_c = 5$.

By comparing Table 5.8 with Table 5.7 we can see that here the FCM algorithm is more suitable than the GK algorithm. When the mixed elements are clustered with the GK algorithm the optimal number of clusters lies at 2. Already at $n_c = 3$ no useful results are produced.

When comparing Table 5.9 with Table 5.1 we can see that the selection of variables of the main elements contributed to an improvement of the clustering result. While the values for PC, PE, S and CI are almost identical until $n_c = 5$, the values for APD and PD are much better at the selection of variables, the optimal number of clusters lies at either $n_c = 22$ or $n_c = 24$ with a selection of variables.

The optimal number of clusters lies at three for the elements of the C-horizon clustered with the GK algorithm (see Table 5.3), with the clustering of the selected main elements, however, it lies at seven (see Table 5.10). This again confirms the assumption that the selection of variables provides better results than the ones without prior selection.

When comparing the validity measures for all remaining compositions of the elements (main elements without Si and Ti, selection of the trace elements, selection of the mixed elements and various compositions of the C-horizon) clustered with all algorithms (FCM, GK and GG), we can see that the selection of the variables produces the best results just followed by a selection of variables which are no outliers. The FCM algorithm is mostly to be prefered to all the other algorithms.

The evaluation of the clustering of the raw data provides the lowest validity measures. The clustering of the standardised data provides lower validity measures than the ones coming from the clustering of the transformed and standardised data. It is interesting to see that the O-horizon is less suitable for clustering than the C-horizon. By comparing Table 5.1, Table 5.11, Table 5.13 and Table 5.14 we can work out the following hierarchy of the best cluster structure in the data:

1. Rock data
2. Walchen data
3. C-horizon
4. O-horizon

Table 5.1: Global validity measures for main elements of C-horizon with FCM

n_c	PC	PE	S	CI	FHV	APD	PD
2	−0.625291	0.556759	0.658435	3.20757	2	-7.28671	-7.28671
3	−0.443408	0.931866	1.57331	6.33103	3	-7.61955	-7.61955
4	−0.370977	1.16595	0.682314	8.14834	4	-9.41773	-9.41773
5	−0.308632	1.36905	0.908516	10.0283	5	-6.93104	-6.93104
6	−0.260384	1.5465	12.7239	12.145	6	-5.43258	-5.43258
7	−0.225863	1.69646	4.21551	13.9575	7	-3.06478	-3.06478
8	−0.200061	1.82627	376.086	15.648	8	-8.323	-8.323
9	−0.178654	1.94465	2183.92	17.2646	9	-2.63137	-2.63137
10	−0.161882	2.04707	2964.68	18.7191	10	-2.89289	-2.89289
11	−0.148564	2.14036	3071.92	20.0811	11	-8.32674	-8.32674
12	−0.137768	2.22421	28859.7	21.3045	12	-7.41043	-7.41043
13	−0.128505	2.29995	1801.28	22.5114	13	-2.39404	-2.39404
14	−0.122	2.36723	43929.9	23.5091	14	-8.51139	-8.51139
15	−0.116038	2.42882	15391.4	24.4044	15	-7.91491	-7.91491
16	−0.10976	2.49192	504299	25.4464	16	-7.8168	-7.8168
17	−0.10456	2.54837	844975	26.411	17	-3.80348	-3.80348
18	−0.0993869	2.60545	1.01702e+07	27.4419	18	-8.17271	-8.17271
19	−0.0946139	2.66068	1.26215e+07	28.4576	19	-8.26481	-8.26481
20	−0.0906569	2.71009	368685	29.3868	20	-3.66314	-3.66314
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (4)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (4)	0.4	0.5	1.0	0	0	0	0

Table 5.2: Global validity measures for main elements of C-horizon with FCM, fuzzifier=2.5

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.55	0.63	0.94	3.42	2	-3.98	-3.98
3	-0.37	1.03	4.87	6.38	3	-9.30	-9.30
4	-0.28	1.32	18.64	8.74	4	-6.99	-6.99
5	-0.22	1.54	3250.52	10.67	5	-5.88	-5.88
6	-0.18	1.72	11462.6	12.40	6	-3.67	-3.67
7	-0.16	1.88	3.2296e+06	13.95	7	-4.27	-4.27
8	-0.14	2.01	2.47749e+09	15.31	8	-3.46	-3.46
9	-0.12	2.13	7.52823e+13	16.57	9	-2.33	-2.33
10	-0.11	2.23	1.2427e+07	17.85	10	-3.12	-3.12
11	-0.10	2.33	1.09513e+08	18.96	11	-2.41	-2.41
12	-0.09	2.41	7.55707e+11	19.97	12	-1.98	-1.98
13	-0.08	2.49	8.33847e+09	20.99	13	-1.62	-1.62
14	-0.08	2.57	1.96001e+07	22.03	14	-1.99	-1.99
15	-0.07	2.64	3.87314e+09	22.90	15	-1.58	-1.58
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (3)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (3)	0.4	0.5	1.0	0	0	0	0
2 (3)	1.0	0.5	1.0	0	0	0.7	0.7

Table 5.3: Global validity measures for main elements of C-horizon with GK

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.63	0.54	1.13	3.59	0.03	-78855.3	-6801.68
3	-0.43	0.93	1.53	6.90	0.09	-265.63	-10.20
4	-0.33	1.21	362044	9.19	0.07	-581.79	-43.32
5	-0.27	1.42	5.78554e+07	11.28	0.07	-678.22	-51.46
6	-0.22	1.60	109221	13.14	0.07	-526.86	-58.35
7	-0.19	1.75	227225	14.87	0.07	-487.75	-60.77
8	-0.17	1.88	1.9238e+12	16.21	0.05	-0	-0
9	-0.16	1.98	1.69398e+08	17.31	0.03	449.65	-130.75
10	error	error	error	error	error	error	
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (3)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (3)	0.4	0.5	1.0	0	0	0	0

Table 5.4: Global validity measures for main elements of C-horizon with GG

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.96	0.05	1.60	0.20	0.124627	-0	-0
3	-0.93	0.11	2.64	0.39	0.242817	-0	-0
4	-0.95	0.07	5.52	0.45	0.315184	-0	-0
5	-0.94	0.09	1.10	0.36	0.0341557	-0	-0
6	-0.85	0.23	1e+20	10.98	3.77	-0	-0
7	-0.84	0.29	1e+20	12.11	0.74	-0	-0
8	-0.88	0.19	1e+20	5.94	0.02	-0	-0
9	nan	-0	1e+20	nan	0.02	-0	-0
10	-0.92	0.14	1e+20	12.37	0.08	-0	-0
11	-0.90	0.21	1e+20	21.57	32.02	-0	-0
12	nan	6-0	nan	nan	0.05	-0	-0
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (5)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (5)	0.4	0.5	1.0	0	0	0	0

Table 5.5: Global validity measures for trace elements of C-horizon with FCM

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.62	0.56	0.55	3.25	2	-0	-0
3	-0.44	0.93	0.92	6.38	3	-1.74	-1.74
4	-0.33	1.21	3.16	9.23	4	-1.60	-1.60
5	-0.27	1.43	24.95	11.62	5	-0.74	-0.74
6	-0.22	1.62	535.52	13.72	6	-1.37	-1.37
7	-0.19	1.77	6313.52	15.60	7	-0.65	-0.65
8	-0.17	1.90	6564.73	17.32	8	-0.59	-0.59
9	-0.15	2.02	194223	18.91	9	-0.53	-0.53
10	-0.13	2.12	2.58152e+08	20.32	10	-0	-0
11	-0.12	2.22	84369.3	21.72	11	-0	-0
12	-0.11	2.31	263935	23.04	12	-0	-0
13	-0.10	2.38	1.77698e+06	24.25	13	-0	-0
14	-0.09	2.46	738259	25.43	14	-0	-0
15	-0.09	2.53	8.31085e+08	26.56	15	-0	-0
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (3)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (3)	0.4	0.5	1.0	0	0	0	0

Table 5.6: Global validity measures for trace elements of C-horizon with GK

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.50	0.68	15.05	5.45	0.0009	-4469.98	-3343.9
3	-0.34	1.08	421498	8.86	0.0013	-3367.24	-2362.26
4	-0.26	1.36	8.88467e+06	11.49	0.0016	-1760.65	-1269.84
5	-0.20	1.58	1.87547e+08	13.70	0.0018	-1466.74	-1134.41
6	-0.17	1.76	5.68983e+09	15.63	0.0020	-1259.69	-1063.17
7	-0.15	1.92	1.53592e+12	17.35	0.0020	-1113.38	-1025.13
8	-0.13	2.05	9.16642e+12	18.92	0.0021	-1014.49	-997.48
9	-0.11	2.16	2.19962e+10	20.36	0.0022	-966.90	-956.35
10	-0.10	2.26	1.39993e+13	21.68	0.0023	-993.56	-932.86
11	-0.09	2.36	2.0691e+12	22.94	0.0024	-1202.81	-891.53
12	-0.09	2.44	1.85501e+14	24.13	0.0025	-2238.58	-843.48
13	-0.08	2.51	3.52548e+15	24.93	0.0024	-4250.18	-904.97
14	-0.08	2.58	1.0385e+13	26.01	0.0025	-13823.6	-846.56
15	-0.07	2.65	6.54847e+12	27.08	0.0027	-51796.5	-787.99
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (3)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (3)	0.4	0.5	1.0	0	0	0	0

Table 5.7: Global validity measures for mixed elements from the C-horizon clustered with FCM

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.60	0.58	0.67	3.56	2	-0	-0
3	-0.42	0.96	1.52	6.94	3	-0	-0
4	-0.31	1.25018	6.62	9.80	4	-0.87	-0.87
5	-0.25	1.46986	15.32	12.14	5	-0.84	-0.84
6	-0.21	1.65022	1624.37	14.23	6	-0.79	-0.79
7	-0.18	1.80451	47874.8	616.11	7	-0.72	-0.72
$n_c^{opt}(n_c^{(2)})$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
2 (3)	0.4	0.5	1.0	0	0.8	0.7	0.7
2 (3)	0.4	0.5	1.0	0	0	0.7	0.7

Table 5.8: Global validity measures for mixed elements from the C-horizon clustered with GK

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.51	0.67	9.12	5.34	1.96465e-06	-1.80048e+06	-553941
3	-0.35	1.07	2.33616e+06	8.57	2.52548e-06	-1.46248e+06	-434753
4	-0.26	1.35	4.31129e+08	11.02	1.99042e-06	-1.20128e+06	-553116
5	-0.21	1.56	2.26419e+10	13.10	1.3105e-06	-1.03687e+06	-839980
6	-0.18	1.74	5.66829e+13	14.98	1.09156e-06	-1.01246e+06	-1.00628e+06
7	-0.16	1.89	4.50536e+12	16.73	1.18359e-06	-1.45539e+06	-924940
8	-0.14	2.02	3.64997e+13	18.37	1.37893e-06	-4.56412e+06	-791026

Table 5.9: Global validity measures for the selected main elements from the C-horizon clustered with the FCM algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.62	0.55	0.70	3.20	2	-33.17	-33.17
3	-0.46	0.90	0.87	5.75	3	-25.76	-25.76
4	-0.40	1.11	0.52	7.11	4	-20.34	-20.34
5	-0.35	1.28	0.82	8.54	5	-14.64	-14.64
6	-0.31	1.44	0.84	9.88	6	-14.94	-14.94
7	-0.27	1.58	0.75	11.49	7	-14.69	-14.69
8	-0.25	1.70	0.77	13.04	8	-12.72	-12.72
9	-0.23	1.80	0.71	13.88	9	-13.49	-13.49
10	-0.23	1.85	0.95	13.87	10	-14.91	-14.91
11	-0.21	1.93	1.04	15.05	11	-12.35	-12.35
12	-0.20	2.01	1.73	16.21	12	-11.42	-11.42
13	-0.18	2.09	1.84	17.17	13	-11.08	-11.08
14	-0.17	2.16	10.25	18.34	14	-9.93	-9.93
15	-0.17	2.21	1.97	18.61	15	-9.18	-9.18
16	-0.16	2.27	2.51	19.50	16	-8.66	-8.66
17	-0.15	2.33	2.51	19.99	17	-8.24	-8.24
18	-0.15	2.38	2.45	20.69	18	-8.16	-8.16
19	-0.14	2.43	6.19	21.73	19	-7.89	-7.89
20	-0.14	2.48	6.85	22.48	20	-7.62	-7.62
21	-0.14	2.51	2.15	22.61	21	-7.96	-7.96
22	-0.13	2.56	1.50	22.7741	22	-7.76	-7.76
23	-0.13	2.60	2.75	23.78	23	-7.25	-7.25
24	-0.12	2.65	426.38	25.3092	24	-6.50	-6.50
25	-0.12	2.68	8.18	25.026	25	-6.36	-6.36
26	-0.12	2.72	11584.1	26.30	26	-5.92	-5.92
27	-0.12	2.74	1.45132e+09	26.43	27	-6.18	-6.18

Table 5.10: Global validity measures for the selected main elements from the C-horizon clustered with the GK algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.66	0.50	1.03	3.05	0.27	-3085.16	-1217.33
3	-0.47	0.88	0.99	5.97	0.28	-288.94	-131.67
4	-0.35	1.16	1.85	8.79	0.34	-227.43	-106.69
5	-0.30	1.36	1.94	10.20	0.34	-146.63	-72.17
6	-0.26	1.52	1.49	11.78	0.35	-201.62	-97.28
7	-0.23	1.67	3.21	13.34	0.36	-217.859	-120.83
8	-0.20	1.80	6.33	14.82	0.38	-204.44	-114.43
9	-0.18	1.91	1095.04	16.20	0.38	-220.92	-134.71
10	-0.17	2.01	17112.3	17.45	0.41	-187.20	-113.08

Table 5.11: Global validity measures for the mainly used elements (Al, Ca, Co, Cu, Fe, Mg, Mn, Na, Ni, Pb, Sr and V) without transformation of the O-horizon clustered with the FCM algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.689628	0.480512	0.461393	2.63413	2	-6.43919	-6.43919
3	-0.497024	0.842514	1.35836	5.22751	3	-3.66717	-3.66717
4	-0.438102	1.05702	0.721534	6.36334	5	-0	-0
5	-0.361849	1.25762	7.52899	8.59393	6	-0	-0
6	-0.302143	1.44557	12.3877	11.0047	7	-0	-0
7	-0.265428	1.58695	16.347	12.6643	8	-0	-0
8	-0.239615	1.70797	142152	14.2676	9	-0	-0
9	-0.224048	1.80411	1101.77	15.0455	10	-0	-0
10	-0.227732	1.81158	1684.54	14.9039	11	-0.972882	-0.972882
11	-0.207604	1.91525	3.79849e+07	16.4712	12	-0.971263	-0.971263

Table 5.12: Global validity measures for the mainly used elements (Al, Ca, Co, Cu, Fe, Mg, Mn, Na, Ni, Pb, Sr and V) of the O-horizon clustered with the FCM algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.555828	0.634794	1.81014	4.60454	2	-13.289	-13.289
3	-0.389091	1.01062	5.73387	7.6676	3	-10.5828	-10.5828
4	-0.307881	1.26886	26254.3	9.88433	4	-7.16118	-7.16118
5	-0.257045	1.4705	666.565	11.859	5	-5.63721	-5.63721
6	-0.221073	1.63893	118041	13.7423	6	-4.76453	-4.76453

Table 5.13: Global validity measures for the Walchen data custered with the FCM algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.6165	0.566507	0.842476	3.38704	2	-27.943	-27.943
3	-0.438113	0.938461	1.55702	6.38227	3	-16.8219	-16.8219
4	-0.338542	1.20869	2.13142	8.98192	4	-6.80781	-6.80781
5	-0.322285	1.32393	2.23021	9.55637	5	-3.03532	-3.03532
6	-0.27333	1.49672	2.74455	11.4891	6	-17.65	-17.65
7	-0.23371	1.65614	5.00656	13.4781	7	-12.8642	-12.8642
8	-0.202959	1.79778	3.27163e+06	15.4103	8	-11.0676	-11.0676
9	-0.179598	1.92303	15692.4	17.1245	9	-9.15687	-9.15687

Table 5.14: Global validity measures for the main elements of the Rock data clustered with the FCM algorithm

n_c	PC	PE	S	CI	FHV	APD	PD
2	-0.689628	0.480512	0.461393	2.63413	2	-6.43919	-6.43919
3	-0.497024	0.842514	1.35836	5.22751	3	-3.66717	-3.66717
4	-0.497497	0.870652	0.966742	5.33836	4	-0	-0
5	-0.438102	1.05702	0.721534	6.36334	5	-0	-0
6	-0.361849	1.25762	7.52899	8.59393	6	-0	-0
7	-0.302143	1.44557	12.3877	11.0047	7	-0	-0
8	-0.265428	1.58695	16.347	12.6643	8	-0	-0
9	-0.239615	1.70797	142152	14.2676	9	-0	-0
10	-0.224048	1.80411	1101.77	15.0455	10	-0	-0
11	-0.227732	1.81158	1684.54	14.9039	11	-0.972882	-0.972882
12	-0.207604	1.91525	3.79849e+07	16.4712	12	-0.971263	-0.971263
13	-0.190252	2.01216	3.014e+08	18.0242	13	-1.93486	-1.93486

Table 5.15: Silhouette Coefficients for the C-horizon; non-standardised data[1]

n_c	m.el	m.el.woSiTi	m.el.varsel	tr.el	tr.el.woAs	tr.el.varsel	mix.el.	mix.el.varsel
2	0.39	0.39	0.43	0.38	0.38	0.39	0.37	0.44
3	0.17	0.17	0.38	0.47	0.47	0.41	0.35	0.32
4	0.25	0.25	0.28	0.16	0.16	0.26	0.36	0.36
5	0.33	0.33	0.22	0.11	0.27	0.26	0.36	0.47
6	0.27	0.27	0.30	0.24	0.16	0.24	0.41	0.46
7	0.20	0.20	0.26	0.17	0.17	0.15	0.44	0.47

[1]m.el: main elements, m.el.woSiTi: main elements without Si and Ti, m.el.varsel: selection of the main elements, tr.el: trace elements, tr.el.woAs: trace elements without As, tr.el.varsel: selection of the trace elements, mix.el.: mixed elements, mix.el.varsel: selection of the mixed elements

Table 5.16: Silhouette Coefficients for the C-horizon; log-transformed and standardised data

n_c	m.el	m.el.woSiTi	m.el.varsel	tr.el	tr.el.woAs	tr.el.varsel	mix.el.	mix.el.varsel
2	0.24	0.23	0.30	0.23	0.32	0.28	0.23	0.26
3	0.25	0.30	0.27	0.14	0.21	0.23	0.19	0.15
4	0.11	0.23	0.18	0.17	0.19	0.11	0.21	0.15
5	0.23	0.21	0.17	0.17	0.15	0.17	0.22	0.11
6	0.15	0.23	0.19	0.14	0.18	0.17	0.22	0.17
7	0.15	0.14	0.20	0.15	0.18	0.16	0.14	0.14

5.5 Results for the Silhouette Coefficient

The silhouette coefficients were calculated for the interval of two till seven clusters and are shown in Tables 5.5 to 5.5. As expected the non-standardised data have higher silhouette coefficients than the standardised data. However, in this case clustering is only done with variables which have a large scale, variables with small range have no contribution. In Table 5.5 we can see that according to Table 4.3.2 for the main elements of the C-horizon no substantial stucture can be found in this data for any number of clusters. In Tables 5.5 and 5.5 very few values show a weak structure. The only reason why the values for standO and $n_c = 2$ of 0.78 in Table 5.5 can be achieved lies in the fact that there are only two objects in one cluster. Basically, the O-horizon data only shows a better clustering structure when they are not logarithmised (see Table 5.5). According to Table 4.3.2 these data show weak or even no clustering structure and therefore we can expect no good clustering results.

Table 5.17: Silhouette Coefficients for the O-horizon, Walchen data and Rock data; standardised data[1]

n_c	standO	standlogO	standW	standlogW	standRock	standlogRock
2	0.78	0.17	0.39	0.27	0.30	0.35
3	0.24	0.09	0.17	0.16	0.26	0.27
4	0.21	0.19	0.15	0.13	0.23	0.22
5	0.16	0.16	0.20	0.18	0.30	0.25
6	0.20	0.14	0.16	0.18	0.21	0.22
7	0.14	0.15	0.14	0.16	0.22	0.21

[1]standO: standardised data from the O-horizon, standlogO: log-transformed and standardised data from the O-horizon, standW: standardised Walchen data, ..., standlogRock: log-transformed and standardised Rock data

Chapter 6

Summary

In Chapter 1 the proof was given that the (underlying) geochemical data must be standardised before clustering.

In Chapter 2 many clustering methods were mentioned. Unfortunately, only a handful of cluster algorithms are included in the statistics package `R`. Especially for fuzzy clustering there is only the algorithm FANNY, which did not produce useful results. For this reason the algorithms of Höppner (2000) for fuzzy clustering of the data were applied in this work. It would have been useful to integrate these algorithms (written in `C++`) into `R` in order to escape the tiresome data set import/export between `R` and the programmes of Höppner.

New kinds of cluster algorithm should also be integrated into `R`. These new algorithms could be applied to the existing data and its results could be evaluated with the help of suitable validity measures. With the help of these suitable validity measures the algorithms could be compared in order to find the best one for the underlying data.

Höppner follows very simple rules initialising the prototypes/memberships. With newer approaches presented, for example, in Chapter 2 the danger of getting bad results by a bad initialisation could be minimised.

In spite of having the worst results with regard to the validity measures the O-horizon (with an appropriate selection of the elements) and the Walchen data basically provided results which were easiest to interpret. The trace elements of the C-horizon, the mixed elements of the C-horizon as well as various choices of these did not show good cluster stuctures and their results were difficult to interpret, while the main elements did display relatively clear structures.

The results of the clustering of the existing data showed that the selection of the elements is very important for the clustering. The clustering of the variables can be helpful.
Clustering of the O-horizon provided a clear indication of the pollution around Nikel, Zapoljarnij and Monchegorsk also the sea spray was clearly visible.
Various kinds of rock formations were made visible by clustering the C-horizon. The distinction between the Ennstaler phyllites and the Wölzer mica schist as well as the localisation of the ore deposits were revealed by the clustering of the Walchen data.

The hierarchical clustering routine AGNES and the partitioning clustering routines PAM and CLARA made it possible to reveal most of the structures. The plots created with the help of fuzzy clustering have the advantage of providing more information by the memberships to the clusters. In most cases, however, they were not able to recognise so many different clusters as PAM or CLARA did. The FCM algorithm is able to recognise more structures than the GK algorithm. However, the clusters which were shown with the help of the GK algorithm displays clearer structures. The GG algorithm and the COSA algorithm were not suitable for clustering of the existing data.

Many validity measures were presented in this thesis. However, for the purpose of the evaluation of results and algorithms the statstics package `R` includes a very insufficient choice of validity measures. Nevertheless numerous evidences were given and numerous assumptions were confirmed by applying the existing algorithms of `R` and by using the validity measures included in Höppner's programme. These validity measures can be of some help in the search for the optimal number of clusters. One should also be careful when evaluating the various groups of elements with the help of validity measures. It can happen that results of the clustering of the selected elements are impossible to interpret. It is more useful to choose the elements from a geochemical point of view, paying little attention to the validity measures, in order to make the results as easy to interpret as possible.

The implementation in the package `R` of all validity measures which were presented in Chapter 4 would be very helpful. With these validity measures one could make more statements about the structure of the existing data, the validity of the gained results and about the validity of the various algorithms.

The groups produced with the help of cluster analysis can serve as the input group(s) for the multivariate method *Discriminant Analysis* (DA). With the help of the popular methods *k-th Nearest Neighbour Classification, Linear Discriminant Analysis* (LDA) and *Quadratic Discriminant Analysis* (QDA) or with the combination of these two (RDA - *Regularized Discriminant Analysis*) or with an extension of LDA (*Logistic and Log-linear Discrimination* and *Reduced Linear Discriminant Analysis*) or with modern methods (like for example MDA - *Mixture Discriminant Analysis*, FDA - *Flexible Discriminant Analysis*, EDDA - *Eigenvalue Decomposition Discriminant Analysis*, DANN - *Discriminant Adaptive Nearest Neighbour* and many others) the memberships of objects which have not been assigned in the input group(s) can be classified. The best method for spatial data is PDA - *Penalized Discriminant Analysis* (Hastie et al., 1995). For large data sets applications of *Neural Network to Discriminant Analysis* are most suitable. Especially the groups which show the pollution around Nikel, Monchegorsk and Zapoljarnij as well as the various clusters which show the rock types of the C-horizon are most suitable for being the input groups for DA.

Appendix A

Software

A.1 R

R is an integrated system of software facilities for data manipulation, calculation and graphical display. It is an environment were many classical and modern statistical techniques have been implemented. Some of these are built into the base R environment, but many are supplied as *packages*. There are about 8 packages supplied with R (standard packages) and many more are available through the CRAN family at the internet site *http://cran.r-project.org*. R can be regarded as an implementation of the S language which was developed at Bell Laboratories in the 1980s and forms the basic of the S-PLUS system. For R, the basic references are Becker et al. (1988), Venables and Smith (2002), Verzani (2002) and Ripley (2002). At most R installations, help is available in HTML format by running

```
> help.start()
```

R can be used interactively, but it is also possible to write own functions in the R language. The recall and editing capabilities under UNIX are highly customisable. The author prefers to work with R with the help of the Emacs text editor which provides more general support mechanisms (via ESS, *Emacs Speaks Statistics*; see Rossini et al., 2001).

A.2 Programs

A.2.1 QQ-plots

```
qqplots <- function(){

hist((gestein[,12]),xlab="",main="Histogram of Mg, rock data",
      cex=3,cex.axis=3,cex.main=3,cex.sub=2,cex.lab=2.5)
```

```
qqnorm(c2mm[,105],cex.main=2,cex.axis=2,main="Mg, C-horizon",xlab="",ylab="")
qqline(c2mm[,105])
```

A.2.2 CLARA

```
"clara1" <-
  function (k,x,y)
{

# clara is fully described in chapter 3 of Kaufman and Rousseeuw (1990)
# Return from clara: sample, medoids, clustering, objective, clusinfo, diss,
# silinfo, call, data

library(cluster)

# k ... number of clusters
# x ... 1 for main elements
#   ... 2 for trace elements
#   ... 3 for mixed data
#   ... 4 for mafic rock
#   ... 5 for main elements without Si and Ti
#   ... 6 for trace elements without As
#   ... 7 for main elements variable selection
#   ... 8 for walchen data, or trace elements variable selection
#   ... 9 for mixed data variable selection
# y ... 0 without PCA
#   ... 1 PCA before clustering


if(k<14){r1_3;r2_4}
if(k<10){r1_3;r2_3}
if(k<8){r1_2;r2_3}
if(k<5){r1_2;r2_2}
if(k<3){r1_1;r2_2}

library(cluster)

# import walchen data, for (x==8)

#w_scan("~/Reimann/walchen.txt",what="")
#w1_matrix(as.numeric(a,ncol=30,byrow=T))
#w2_matrix(w1,ncol=30,byrow=T)
```

```
#colnames(w2)_scan("/nfshome1/dipdis/e9656209/Reimann/walchen.var", what="")
#w3_w2[,2:30]

if(y==0){

  if(x==1){data_c2mm[,101:110];name_"main elements"}
  if(x==2){data_c2mm[,c(8,12,24,27,30,42,58,63,66,70,79,85,90,92,94)];
         name_"trace elements"}
  if(x==3){data_c2mm[,c(6,12,18,24,27,31,33,40,42,48,50,54,58,61,63,66,70,
     79,85,90,92,94)];name_"mixed-data"}
  if(x==4){data_c2mm[,c(24,27,30,34,48,50,58,70,87,90)];name_"mafic rock"}
  if(x==5){data_c2mm[,101:109];name_"xrf wo Si,Ti"}
  if(x==6){data_c2mm[,c(12,24,27,30,42,58,63,66,70,79,85,90,92,94)];
         name_"trace el. wo As"}
  #if(x==8){data_w3[,c(1,4,10,11,12,15,22,24)]}
  if(x==7){data_c2mm[,c(101,102,104,106,109,110)];
           name_"main el. var.selection"}
  if(x==8){data_c2mm[,c(8,12,24,27,30,42,63,66,85,94)];name_"trace el select."}
  if(x==9){data_c2mm[,c(6,12,18,24,27,31,42,48,50,58,61,63,66,85,94)];
         name_"mixed-data sel."}
       }

if(y==1){

  if(x==1){data_scale(log(c2mm[,101:110]))
         pc_princomp(data)
         data_pc$scores[,1:6]
         name_"main el. with pca"}
  if(x==2){data_scale(log(c2mm[,c(8,12,24,27,30,42,58,63,66,70,79,85,
            90,92,94)]));
         pc_princomp(data)
         data_pc$scores[,1:5]
         name_"trace el. with pca"}
  if(x==3){data_scale(log(c2mm[,c(6,12,18,24,27,31,33,40,42,48,50,54,
            58,61,63,66,70,79,85,90,92,94)]));
         pc_princomp(data)
         data_pc$scores[,1:7]
         name_"mixed data with pca"}
  if(x==4){data_scale(log(c2mm[,c(24,27,30,34,48,50,58,70,87,90)]));
         pc_princomp(data)
         data_pc$scores[,1:4]
         name_"mafic rock with pca"}
  if(x==5){data_scale(log(c2mm[,101:109]));
         pc_princomp(data)
```

```
        data_pc$scores[,1:4]
        name_"ma.el. wo. Si,Ti, pca"}
  if(x==6){data_scale(log(c2mm[,c(12,24,27,30,42,58,63,66,70,79,85,
           90,92,94)]));name_"tr.el. wo As w. pca"
        pc_princomp(data)
        data_pc$scores[,1:5]}
  if(x==7){data_scale(log(c2mm[,c(101,102,104,106,109,110)]));
          name_"m.el. sel. pca"
        pc_princomp(data)
        data_pc$scores[,1:3]}
  if(x==8){data_scale(log(c2mm[,c(8,12,24,27,30,42,63,66,85,94)]));
          name_"tr.el sel. pca"
        pc_princomp(data)
        data_pc$scores[,1:7]}
  if(x==9){data_scale(log(c2mm[,c(6,12,18,24,27,31,42,48,50,58,61,
           63,66,85,94)]));name_"mix-data sel. pca"
        pc_princomp(data)
        data_pc$scores[,1:5]}
     }


t3_0;t4_0;t5_0;t6_0;t7_0;m_0

par(mfrow=c(r1,r2),pty="s",xaxt="n",yaxt="n")

for(i in 2:k){
  if(y==0){
  a_clara(scale(log(data)),i)}
  if(y==1){a_clara(data,i)}
  plot(c2mm[,2:3],col=0,mar=c(3,2,2,1),xlab="",ylab="")
    for(j in 1:i){
      points(c2mm[a$clust==j,2:3],pch=15,col=j)}
        m_a$clust==1
        t_0
    for(jj in 1:606){if(m[jj]==TRUE){t_t+1}}
        m_a$clust==2
        t2_0

    for(jj in 1:606){if(m[jj]==TRUE){t2_t2+1}}
  #  for(r in 2:(k-1)){if(i>r){m_a$clust==(r+1)
  #     for(jj in 1:606){if(m[jj]==TRUE){t[r+1]_t[r+1]+1}}}}
    if(i>2){m_a$clust==3
        t3_0
    for(jj in 1:606){if(m[jj]==TRUE){t3_t3+1}}}
```

```
    if(i>3){m_a$clust==4
        t4_0
    for(jj in 1:606){if(m[jj]==TRUE){t4_t4+1}}}
    if(i>4){m_a$clust==5
        t5_0
    for(jj in 1:606){if(m[jj]==TRUE){t5_t5+1}}}
    if(i>5){m_a$clust==6
        t6_0
    for(jj in 1:606){if(m[jj]==TRUE){t6_t6+1}}}
    if(i>6){m_a$clust==7
        t7_0
    for(jj in 1:606){if(m[jj]==TRUE){t7_t7+1}}}

# How many points are in the clusters.

  title(paste(name,"-",i,"cluster"),sub=paste("cl1-",i,":","bla:",t,",r:",
        t2,",g:",t3,",blu:",t4,",cyan:",t5,",mag:",t6,",y:",t7),
        col.sub="slateblue2")
  }


}

o2mmclara <- function(k){

# clara for O-horizon data

library(cluster)

# read the data

o_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.txt",what="")
o1_matrix(as.numeric(o,ncol=43,byrow=T))
o2_matrix(o1,ncol=43,byrow=T)
colnames(o2)_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.var", what="")
data_scale(log(o2[,c(6,13,15,24,36)]));name_"O-horizon; As, Co, Cu, Ni, V"

# identify the shape of par

if(k<13){r1_3;r2_4}
if(k<10){r1_3;r2_3}
if(k<7){r1_2;r2_3}
if(k<5){r1_2;r2_2}
if(k<3){r1_1;r2_2}
```

```
par(mfrow=c(r1,r2))

t3_0;t4_0;t5_0;t6_0;t7_0;m_0

for(i in 2:k){
a_clara(data,i)

plot(o2[,2:3],col=0,mar=c(3,2,2,1),xlab="",ylab="")
      for(j in 1:i){
        points(o2[a$clust==j,2:3],pch=15,col=j)}
        m_a$clust==1
        t_0

      for(jj in 1:617){if(m[jj]==TRUE){t_t+1}}
          m_a$clust==2
          t2_0

      for(jj in 1:617){if(m[jj]==TRUE){t2_t2+1}}
  #   for(r in 2:(k-1)){if(i>r){m_a$clust==(r+1)
  #   for(jj in 1:606){if(m[jj]==TRUE){t[r+1]_t[r+1]+1}}}}
        if(i>2){m_a$clust==3
          t3_0

        for(jj in 1:617){if(m[jj]==TRUE){t3_t3+1}}}

        if(i>3){m_a$clust==4
          t4_0

        for(jj in 1:617){if(m[jj]==TRUE){t4_t4+1}}}

        if(i>4){m_a$clust==5
          t5_0

        for(jj in 1:617){if(m[jj]==TRUE){t5_t5+1}}}

        if(i>5){m_a$clust==6
          t6_0

        for(jj in 1:617){if(m[jj]==TRUE){t6_t6+1}}}

        if(i>6){m_a$clust==7
          t7_0
```

```
        for(jj in 1:617){if(m[jj]==TRUE){t7_t7+1}}}

# How many points are in the clusters.

  title(paste(name,"-",i,"cluster"),sub=paste("cl1-",i,":","bla:",t,",
          r:",t2,",g:",t3,",blu:",t4,",cyan:",t5,",mag:",t6,",y:",t7),
          col.sub="slateblue2")
  }

}


claraspiel <- function(k,x,y){

# Each cluster is plotted in a single picture

# k ... Clusteranzahl
# x ... 1 for main elements (xrf-data)
#   ... 2 for trace elements
#   ... 3 for mixed data
#   ... 4 for mafic rock
#   ... 5 for main elements without Si and Ti
#   ... 6 for trace elements without As
#   ... 7 for main elements variable selection
#   ... 8 for trace elements variable selection
#   ... 9 for mixed data variable selection
# y ... 0 for C-horizon
#   ... 1 for C-horizon PCA
#   ... 2 for O-horizon
#   ... 3 for walchen data

# identify the shape of par

if(k<13){r1_3;r2_4}
if(k<10){r1_3;r2_3}
if(k<7){r1_2;r2_3}
if(k<5){r1_2;r2_2}
if(k<3){r1_1;r2_2}

if(y==0){

  if(x==1){data_scale(log(c2mm[,101:110]));name_"xrf_data"}
  if(x==2){data_scale(log(c2mm[,c(8,12,24,27,30,42,58,63,66,
```

```
          70,79,85,90,92,94)]));name_"trace elements"}
  if(x==3){data_scale(log(c2mm[,c(6,12,18,24,27,31,33,40,42,
          48,50,54,58,61,63,66,70,79,85,90,92,94)]));name_"mixed data"}
  if(x==4){data_scale(log(c2mm[,c(24,27,30,34,48,50,58,70,87,90)]));
          name_"mafic rock"}
  if(x==5){data_scale(log(c2mm[,101:109]));name_"xrf wo Si,Ti"}
  if(x==6){data_scale(log(c2mm[,c(12,24,27,30,42,58,63,66,70,79,85,90,
          92,94)]));name_"trace el. wo As"}
  if(x==7){data_scale(log(c2mm[,c(101,102,104,106,109,110)]));
          name_"main el. var.selection"}
  if(x==8){data_scale(log(c2mm[,c(8,12,24,27,30,42,63,66,85,94)]));
          name_"trace el select."}
  if(x==9){data_scale(log(c2mm[,c(6,12,18,24,27,31,42,48,50,58,61,63,
          66,85,94)]));name_"mixed-data sel."}

      }

if(y==1){

  if(x==1){data_scale(log(c2mm[,101:110]))
          pc_princomp(data)
          data_pc$scores[,1:6]
          name_"main el. with pca"}
  if(x==2){data_scale(log(c2mm[,c(8,12,24,27,30,42,58,63,66,70,79,85,90,
            92,94)]));
          pc_princomp(data)
          data_pc$scores[,1:5]
        name_"trace el. with pca"}
  if(x==3){data_scale(log(c2mm[,c(6,12,18,24,27,31,33,40,42,48,50,54,58,
            61,63,66,70,79,85,90,92,94)]));
          pc_princomp(data)
          data_pc$scores[,1:7]
          name_"mixed data with pca"}
  if(x==4){data_scale(log(c2mm[,c(24,27,30,34,48,50,58,70,87,90)]));
          pc_princomp(data)
          data_pc$scores[,1:4]
          name_"mafic rock with pca"}
  if(x==5){data_scale(log(c2mm[,101:109]));
          pc_princomp(data)
          data_pc$scores[,1:4]
          name_"ma.el. wo. Si,Ti, pca"}
  if(x==6){data_scale(log(c2mm[,c(12,24,27,30,42,58,63,66,70,79,85,90,
            92,94)]));name_"tr.el. woAs w pca"
          pc_princomp(data)
```

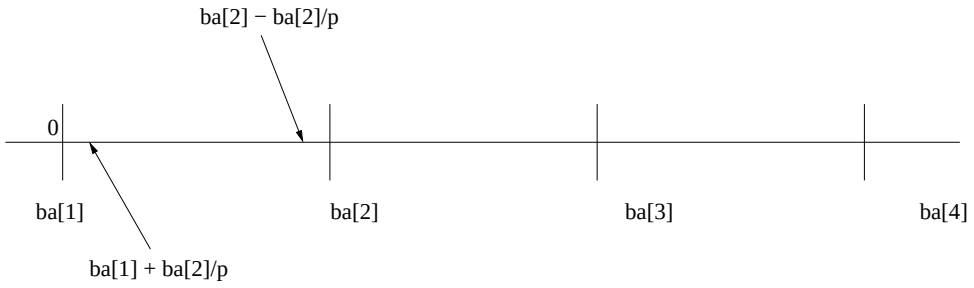
```
          data_pc$scores[,1:5]}
  if(x==7){data_scale(log(c2mm[,c(101,102,104,106,109,110)]));
            name_"m.el. sel. pca"
          pc_princomp(data)
          data_pc$scores[,1:3]}
  if(x==8){data_scale(log(c2mm[,c(8,12,24,27,30,42,63,66,85,94)]));
            name_"tr.el sel. pca"
          pc_princomp(data)
          data_pc$scores[,1:7]}
  if(x==9){data_scale(log(c2mm[,c(6,12,18,24,27,31,42,48,50,58,61,63,66,
            85,94)]));name_"mix-data selection w pca"
          pc_princomp(data)
          data_pc$scores[,1:5]}

    }

if(y==2){

  if(x==1){
     o_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.txt",what="")
     o1_matrix(as.numeric(o,ncol=43,byrow=T))
     o2_matrix(o1,ncol=43,byrow=T)
     colnames(o2)_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.var", what="")
     data_scale(log(o2[,c(6,13,15,24,36)]));name_"main elements"}

    }

if(y==3){

  if(x==1){
     w_scan("~/Reimann/WAL.DAT",what="")
     w1_matrix(as.numeric(w,ncol=30,byrow=T))
     w2_matrix(w1,ncol=30,byrow=T)
     #colnames(w2)_scan("/nfshome1/dipdis/e9656209/Reimann/walchen.var",
     #  what="")
     w3_w2[,2:30]

     ko_scan("~/Reimann/walchen_koord",what="")
     k1_matrix(as.numeric(ko,ncol=9,byrow=T))
     k2_matrix(k1,ncol=9,byrow=T)
     koord_k2[,2:3]
     w_cbind(koord,w3)
     vnames_scan("/nfshome1/dipdis/e9656209/Reimann/walchen.var", what="")
     colnames(w)_vnames[2:32]
```

```
     name_"walch"
   # Data for clustering
     data_scale(log(w[,c(3,6,12,13,15,17,24,26)]))}

    }

d_colnames(data)

par(mfrow=c(r1,r2),pty="s",xaxt="n",yaxt="n",mar=c(3,2,2,1))

a_clara(((data)),k)

  if(y==0 | y==1){

     for(i in 1:k){
        plot(c2mm[,2:3],col=gray(0.95),mar=c(3,2,2,1),xlab="",ylab="",pch=15)
        points(c2mm[a$clust==i,2:3],pch=15,col=4)}
     }

if(y==2){

  for(i in 1:k){
    plot(o2[,2:3],col=gray(0.95),mar=c(3,2,2,1),xlab="",ylab="",pch=15)
    title(paste("O-horizon: As, Co, Cu, Ni, V; cluster",i))
    points(o2[a$clust==i,2:3],pch=15,col=4)}

    }

if(y==3){

  for(i in 1:k){
    plot(w[,1:2],col=gray(0.95),mar=c(3,2,2,1),xlab="",ylab="",pch=15)
    points(w[a$clust==i,1:2],pch=15,col=4)}

   }
}



# Read in the data as in claraspiel

# clustering

  a_clara(((data)),k)
```

```
# read the center

  cent_matrix(a$med,ncol=k,byrow=T)

# names of the variables

  rnam_colnames(data)
  for(j in 1:p){
    rnam[j]_substring(rnam[j],1,2)
               }

  rownames(cent)_rnam

  ma_max(abs(cent))

# create the plot

  par(mfrow=c(1,1),cex=1,cex.axis=1,cex.lab=1.5,xaxt="s",yaxt="s")

  plot(cent[,1],type="n",xlim=range(0,1),ylim=range(-ma-0.3,ma+0.3),
       xlab="",ylab="")

        segments(0, 0, 1, 0)
        segments(0, 0.5, 1, 0.5, lty = 2)
        segments(0, -0.5, 1, -0.5, lty = 2)
  title(paste(name,";",k,"cluster with Clara"))

  bb <- c(0,1)
  bb1 <- bb/k
  ba <- seq(from = bb1[1], by = bb1[2])
  ba1 <- ba[2]/20
  ba2 <- c(0,ba1)

  segments(0,-ma,0,ma)

  for(i in 1:(k+1)){
        segments(ba[i],-ma,ba[i],ma)}

# create weights
  weight_0
  for(i in 1:k){
        weight[i]_a$clusinfo[i,1]
      }
```

```
  sumweight_sum(as.numeric(weight))

# text

   for(j in 1:k){
     text(seq(from=ba[j]+ba[2]/p,to=ba[j+1]-ba[2]/p,
          by=(ba[j+1]-ba[j]-2*ba[2]/p)/(p-1)), cent[,j], rownames(cent),
          col="blue",cex=1.5)
     text(ba[j]+ba[2]/2,ma+0.3,paste("Cluster ",j,sep=""),col="red",cex=1.5)

     text(ba[j]+ba[2]/2,ma+0.1,paste("size= ",
          substring(as.numeric(weight[j])/sumweight,1,4),sep=""),
          col=5,cex=1.4)}

}
```

The sequence `seq(from=ba[j]+ba[2]/p,to=ba[j+1]-ba[2]/p, by=(ba[j+1]-ba[j]-2*ba[2]/p)/(p-1))` is achieved by:



In order to arrange the elements in a constant distance from each other for the first clustering (see e.g. the final result in Figure 5.2) the sequence starts from `ba[1] + ba[2]/p` and ends at `ba[2] - ba[2]/p`. From this we can yield the increment for the horizontal arrangement of the elements for each cluster:

$$\mathtt{by} = \frac{\mathtt{ba[j+1]} - \mathtt{ba[j]} - \frac{2.\mathtt{ba[2]}}{\mathtt{p}}}{\mathtt{p}-1} \tag{A.1}$$

A.2.3 Agnes

```
"agnesplot" <-
function ()
{
```

```
# Computes agglomerative hierarchical clustering of the dataset.
# which.plots = 2 for a dendrogramm.

library(cluster)

par(mfrow=c(2,2))

# Data: Main elements from the C-horizon
b_agnes(scale(log(t(c2mm[,101:110]))),method="average")
plot(b,which.plots = 2,main="xrf")

# Data: Trace elements from the C-horizon
b_agnes(scale(log(t(c2mm[,c(8,12,24,27,30,42,58,63,66,70,79,85,90,92,94)]))),
        method="average")
plot(b,which.plots = 2,main="trace elements")

# Data: Mixed elements from the C-horizon
b_agnes(scale(log(t(c2mm[,c(6,12,18,24,27,31,33,40,42,48,50,54,58,61,63,66,
                            70,79,85,90,92,94)]))),method="average")
plot(b,which.plots = 2,main="mixed data")

# Data: Mafic rocks from the C-horizon
b_agnes(scale(log(t(c2mm[,c(24,27,30,34,48,50,58,70,87,90)]))),
        method="average")
plot(b,which.plots = 2,main="mafic rocks")

# Data: Oxide from Rock Data
b_agnes(scale(log((gestein[1:170,8:17]))),method="average")
plot(b,which.plots = 2,main="trace elements")

}
```

A.3 Hoeppners Fuzzy Clustering Program

This package was written in C++ by Höppner (2000). The programs in this package are free software and they can be modified under the terms of the *General Public Licence* (GNU). The package contains some initialization algorithm, the fuzzy C-means algorithm, the Gustafson-Kessel algorithm, the Gath-Geva algorithm, some algorithms for calculating validity measures and some tools for visualisation.
The Data Description Language (DDL) is used to exchange information between different programs by means of ASCII files. It is used to define data and their relationships to other data. The DDL language was chosen because it provides a simple mechanism to share data

analysis software with others. One can write own extensions in a different programming language and add these extensions only by sticking to the DDL format. On the web-site *www.fuzzy-clustering.de* operated by Höppner (2000), the package can be downloaded. Install the package for Unix in a subdirectory fc-X.Y.Z. Type the command gunzip to unzip the package. With the commands ./configure and make the package will be installed.
For displaying data you can use the gnuplot program from Williams and Kelley (1998) or OpenGl (see e.g. Fullagar, 1994; Glaeser and Stachel, 1999). For gnuplot use the gsv tool, otherwise the xsv tool for visualising data.

A.3.1 Data Set Import

To export data from R to the fuzzy-clustering package from Höppner only the data observed in a file must be stored. Once the data are inside the DDL-stream, subsets can not be selected.
To store e.g. the main elements of the C-horizon saved in the R-object (c2mm[,101:110]) in a file one should use the command

```
write.table(c2mm[,101:110], file = "~/ .../fc-X.Y.Z/data/xrfdata.dat", quote
                    = F, row.names = F, col.names = F)
```

The file xrfdata.dat contains only numbers.

Having done this, convert the data into the DDL format. Put in the directory .../fc-X.Y.Z/src the command

```
ddlimport ../data/xrfdata.dat > ../data/xrfdata.ddl
```

Now the data are prepared for the fuzzy clustering package.

A.3.2 Examples

Type all your commands in the ../fc-X.Y.Z/src directory.
The data are displayed by

```
gsv -q -Szero < ../data/xrfdata.ddl
```

-q (quiet mode) produces no ANALYSIS symbol in the DDL output and the interim output should be zero (-Szero). Each ANALYSIS symbol in the DDL-stream represents a clustering task or result. For each ANALYSIS symbol that has been read, the program does its job and writes the result into the output stream.
The FCM algorithm tries to identify a fixed number of prototypes and assigns all objects

from the data to these prototypes. Additionally the GK and the GG algorithm calculate the covariance matrix and the norm matrix. The FCM, GK and GG algorithms need an initialisation algorithm to set the prototypes. Try

```
cini -r4 < ../data/xrfdata.ddl | fcm | gsv -q
```

to apply the FCM algorithm with four clusters (`-r4`) and display the result with `gsv`.

Available options of the `gsv` tool:
-h or –list-options
-n or –analysis-name arg
-r or –cluster-range arg
-c or –constants arg
-a or –dist-shift arg
-f or –fuzzifier arg
-p or –postscript
-g or –graphics-mode arg
-i or –init-by-membership
-d or –inter-data-distance arg
-s or –interim-output arg
-x or –maximum-features arg
-q or –quiet-mode
-o or –output [arg]
-k or –select arg
-X or –select-x-axis arg
-Y or –select-y-axis arg
-Z or –select-z-axis arg
-T or –select-t-axis arg
-C or –select-c-axis arg
-A or –select-a-axis arg
-I or –select-i-axis arg
-S or –select-s-axis arg
-U or –select-dx-axis arg
-V or –select-dy-axis arg
-W or –select-dz-axis arg
-u or –substitute-features
-m or –max-iterations arg
-e or –epsilon arg
-v or –unsupervised [arg]
-w or –weight-factor arg
-t or –input-transformation arg
-R or –output-transformation arg
-D or –data-scale arg
-l or –prefer-local [arg]
-H or –sphere-hierarchy arg

-L or –connect arg
-N or –compare-name arg
-Q or –window arg
-F or –filter arg

The graph of the membership degrees can be displayed by

```
cini -r4 < ../data/xrfdata.ddl | fcm | mesh | fcm -o | gsv -g3 -q -Zmemb
```

whereas `-o` outputs the memberships. To diplay a 3-d graph type `-g3`, for the membership degrees use `-Zmemb`.
In this plot no structure can be recognised and no other options to display the data is available. To be display the results on a map with the help of coordinates (each object is assigned to coordinates) the results have to be exported to `R`.
A useful tool in this package is the pick tool. If e.g. the GG algorithm is used in an unsupervised mode, the data set is evaluated for multiple numbers of clusters automatically by using the pick tool. The pick tool returns only the best partition which is selected according to a votion vector (see below). By typing

```
cini -r2:10 < ../data/xrfdata.ddl | gk -m100 | pick | gsv -q
```

the best partition out of a number of clusters from 2 to 10 can be found. The algorithm of Gustafson-Kessel is very complex and when calculating ten clusters the algorithm can break down, because the covariance matrix must be inverted which is computorially expensive and moreover the inverted matrix may become singular. `-m` stands for maximum iteration and it is better to demand e.g. a maximum of 100 iterations.
In the file `xrfdata.ddl` the *voting vector* can be typed in. The voting vector contains the rating of seven cluster validity measures: the partition coefficient, the partition entropy, the separation, the contraction index, the fuzzy hyper-volume, the average partition density and the partition density (default $0.4, 0.5, 1.0, 0, 0.8, 0.7, 0.7$).

The validity measures for each cluster can be plotted by

```
cini -r2:20 < ../data/xrfdata.ddl | fcm | gsv -q -Xnpro -Yvalm:2 -Tone
                                    -Lseqc
```

To save results type

```
cini -r5 < ../data/xrfdata.ddl | gg -m1000 | save
```

to get the resulting file `xrfdata.cini.gg.tmp` .
To display the clusters on the map of the region, export the data to `R`. The command

```
ddlexport ../data/xrfdata.ddl > ../data/xrffinal.dat
```

deletes most of the non-numeric characters. Now the data can be read by `R`.

A.3.3 Implementation in R

A.3.4 Fuzzy-programs

```
landplot <- function(k,x,m)
  {

# k ... number of clusters

# x ... 1 for main elements
#   ... 2 for trace elements
#   ... 3 for mixed data
#   ... 4 for mafic rock
#   ... 5 for main elements without Si and Ti
#   ... 6 for trace elements without As
#   ... 7 for main elements variable selection
#   ... 8 for O-horizon main elements
#   ... 9 for Walchen data
#   ...10 for trace elements variable selection
#   ...11 for mixed data variable selection
#   ...12 for Rock data

# m ... 1 Fuzzy C-means algorithm
#   ... 2 Gustafson-Kessel algorithm
#   ... 3 Gath-Geva algorithm

if(x==1){xname_"main elements"}
if(x==2){xname_"trace elements"}
if(x==3){xname_"mixed-data"}
if(x==4){xname_"mafic rock"}
if(x==5){xname_"main wo SiTi"}
if(x==6){xname_"trace el. wo As"}
if(x==7){xname_"main_varsel"}
if(x==8){xname_"O-horizon main el."}
if(x==9){xname_"walchen Na,Mg,Ba,Sr,Ni,Co,Cu,Cr"}
if(x==10){xname_"trace el. varsel"}
if(x==11){xname_"mixed-data varsel"}
if(x==12){xname_"rocks oxid"}
if(m==1){methodname_"fcm"}
if(m==2){methodname_"gk"}
if(m==3){methodname_"gg"}
```

```
if(x==1 | x==2 | x==3 | x==4 | x==5 | x==6 | x==7 | x==10 | x==11){
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
  a1_a[(length(a)-606*k*3+1):length(a)]
  a2_matrix(as.numeric(a1[seq(from=1,by=3,to=606*k*3)]),ncol=k,byrow=T)
  source("~/public_html/loadc2mm")
  c2mm_loadc2mm(path="~/public_html/")
}

if(x==8){
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
  a1_a[(length(a)-617*k*3+1):length(a)]
  a2_matrix(as.numeric(a1[seq(from=1,by=3,to=617*k*3)]),ncol=k,byrow=T)

  o_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.txt",what="")
  o1_matrix(as.numeric(o,ncol=43,byrow=T))
  o2_matrix(o1,ncol=43,byrow=T)
  colnames(o2)_scan("/nfshome1/dipdis/e9656209/Reimann/o2mm.var", what="")

}

if(x==9){
# a_scan("/nfshome1/dipdis/e9656209/public.html/fc-0.3.7/data/1",what="")
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
  a1_a[(length(a)-772*k*3+1):length(a)]
  a2_matrix(as.numeric(a1[seq(from=1,by=3,to=772*k*3)]),ncol=k,byrow=T)

  w_scan("~/Reimann/WAL.DAT",what="")
  w1_matrix(as.numeric(w,ncol=30,byrow=T))
  w2_matrix(w1,ncol=30,byrow=T)
  #colnames(w2)_scan("/nfshome1/dipdis/e9656209/Reimann/walchen.var", what="")
  w3_w2[,2:30]

  ko_scan("~/Reimann/walchen_koord",what="")
  k1_matrix(as.numeric(ko,ncol=9,byrow=T))
  k2_matrix(k1,ncol=9,byrow=T)
  k2[423,8]_8
  k2[423,9]_0.20
  koord_k2[,2:3]
  w_cbind(koord,w3)
  vnames_scan("/nfshome1/dipdis/e9656209/Reimann/walchen.var", what="")
  colnames(w)_vnames[2:32]
}

if(x==12){
```

```
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
  a1_a[(length(a)-500*k*3+1):length(a)]
  a2_matrix(as.numeric(a1[seq(from=1,by=3,to=500*k*3)]),ncol=k,byrow=T)
}

if(k<13){r1_3;r2_4}
if(k<9){r1_3;r2_3}
if(k<7){r1_2;r2_3}
if(k<5){r1_2;r2_2}
if(k<3){r1_1;r2_2}

par(mfrow=c(r1,r2),pty="s",xaxt="n",yaxt="n",mar=c(4,2,4,2))

if(x==1 | x==2 | x==3 | x==4 | x==5 | x==6 | x==7 | x==10 | x==11){
  for(i in 1:k){
    plot(c2mm[,2:3],col=0,xlab="",ylab="")
    title(paste(xname,"-",methodname,"cluster",i))

    for(j in 1:10){
      points(c2mm[a2[,i]>2*j/(10*k),2:3],pch=15,col=gray(1-j/10))
              }
}
}

if(x==8){
  for(i in 1:k){
    plot(o2[,2:3],col=gray(0.95),xlab="",ylab="",pch=15)
    title(paste(xname,"-",methodname,"cluster",i))

    for(j in 1:10){
      points(o2[a2[,i]>2*j/(10*k),2:3],pch=15,col=gray(1-j/10))
              }
}
}

if(x==9){
  for(i in 1:k){
    plot(w[,1:2],col=0,xlab="",ylab="")
    title(paste(xname,"-",methodname,"cluster",i))

    for(j in 1:10){
      points(w[a2[,i]>2*j/(10*k),1:2],pch=15,col=gray(1-j/10))
              }
}
```

```
}

if(x==12){
  for(i in 1:k){
    plot(gestein[,3:4],col=gray(0.90),xlab="",ylab="")
    title(paste(xname,"-",methodname,"cluster",i))

    for(j in 1:10){
      points(gestein[a2[,i]>2*j/(10*k),3:4],pch=15,col=gray(1-j/10))
              }
}
}


}


elementplot <- function(p,k,x,m){

# p ... Anzahl der Variablen

# k ... Anzahl der Cluster

# x ... 1 for main elements
#       2 for trace elements
#       3 for mixed-data
#   ... 4 for main elements without SiTi

# m ... 1 for fcm
#       2 for gk
#       3 for gg

# Calculate from R

#if(x==1){
#  comline_paste("cini -r",k," < ~/public_html/fc-0.3.7/data/xrfscale.ddl |
#     fcm -o | save",  system(comline)
#  system("~/public_html/fc-0.3.7/src/ddlexport
#     ~/public_html/fc-0.3.7/data/xrfscale.dat.ci}

#if(x==2){
#  comline_paste("cini -r",k," < ~/public_html/fc-0.3.7/data/speldaten.ddl |
#    fcm -o | save"  system(comline)
#  system("~/public_html/fc-0.3.7/src/ddlexport
```

```
#    ~/public_html/fc-0.3.7/data/spel daten.dat.c}

#if(x==3){
#  comline_paste("cini -r",k," < ~/public_html/fc-0.3.7/data/gem.ddl |
#    fcm -o | save",sep="  system(comline)
#  system("~/public_html/fc-0.3.7/src/ddlexport
#    ~/public_html/fc-0.3.7/data/gem. dat.cini.fc}


# Identification of data

if(x==1){rnam_colnames(c2mm[,101:110])
         xname_"main elements"}
if(x==2){rnam_colnames(c2mm[,c(8,12,24,27,30,42,58,63,66,70,79,85,90,
                               92,94)])
         xname_"trace elements"}
if(x==3){rnam_colnames(c2mm[,c(6,12,18,24,27,31,33,40,42,48,50,54,58,
                               61,63,66,70,79,85,90,92,94)])
         xname_"mixed-data"}
if(x==4){rnam_colnames(c2mm[,101:109])
         xname_"main elements wo SiTi"}

# read the center

if(m==1){
  methodname_"fcm"
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
  a1_a[(length(a)-606*k*3+1-k*(p+2)):(length(a)-606*k*3)]
  l_2:(p+1)

   for (i in 1:(k-1)){
     l_c(l,seq(from=l[length(l)]+3,to=l[length(l)]+(p+2),by=1))}

  a2_matrix(as.numeric(a1[l]),ncol=p,byrow=T)
  a3_t(a2)}

if(m==2){
  methodname_"gk"
  if(x==1){
    a_scan("~/public_html/fc-0.3.7/data/1",what="")
    a1_a[(length(a)-606*k*3-217*k):(length(a)-606*k*3)]

    if(k==2){l1_c(4:13,221:230)
             l_l1[1:20]}
```

```
  if(k==3){l_c(4:13,220:229,437:446)}
  if(k==4){l_c(4:13,220:229,437:446,654:663)}
  if(k==5){l_c(5:14,221:230,438:447,655:664,872:881)}
  if(k==6){l_c(5:14,221:230,438:447,655:664,872:881,1089:1098)}
  if(k==7){l_c(5:14,221:230,438:447,655:664,872:881,1089:1098,1306:1315)}
  if(k==8){l_c(5:14,221:230,437:446,654:663,871:880,1088:1097,1305:1314,
               1522:1531)}
  if(k==9){l_c(5:14,221:230,437:446,654:663,871:880,1088:1097,1305:1314,
               1522:1531,1739:1748)}

a2_matrix(as.numeric(a1[l]),ncol=10,byrow=T)
a3_t(a2)
   }

if(x==2){
  a_scan("~/public_html/fc-0.3.7/data/1",what="")
    if(k==2){a1_a[(length(a)-606*k*3-(471)*k):(length(a)-606*k*3)]
             l_c(1:15,473:487)}
    if(k==5){a1_a[(length(a)-606*k*3-(472)*k):(length(a)-606*k*3)]
             l_c(3:17,475:489,947:961,1419:1433,1891:1905)}
  a2_matrix(as.numeric(a1[l]),ncol=15,byrow=T)
  a3_t(a2)
}

if(x==3){
  a_scan("~/public_html/fc-0.3.7/data/1",what="")

  if(k==2){
  l1_c(-0.109303, -0.0970839, -0.0192894, -0.0168877, -0.0302753,
       -0.103589, -0.0157168, -0.0561102, -0.111359, -0.0203656,
       -0.0866696, -0.111207, -0.0346597, 0.0206732, -0.0560672,
       -0.089048, 0.00710532, -0.163117, -0.0680395, -0.00520533,
       -0.105934, -0.0821927)
  l2_c(0.206533, 0.199279, 0.047594, -0.010843, 0.0205994, 0.250655,
       -0.00957383, 0.109933, 0.253066, -0.00306461, 0.173058, 0.270948,
        0.0300528, -0.047582, 0.114262, 0.155957, -0.0687036, 0.361571,
        0.153158, -0.0242494, 0.238595, 0.167816)
  a3_cbind(l1,l2)}

  if(k==3){
  l1_c(-0.100287, -0.0921652, -0.0191033, -0.0113154, -0.0243687,
       -0.101436, -0.0108485, -0.0532414, -0.107392, -0.0147357,
       -0.0804162, -0.108742, -0.0279773, 0.0180938, -0.0505254,
       -0.07795, 0.0113418, -0.155435, -0.0649541, -0.00263608,
```

```
          -0.101462, -0.0774677)
    l2_c(-0.099746, -0.0918182, -0.0194562, -0.0108893, -0.0243246,
          -0.10128, -0.0104379, -0.0528545, -0.10717, -0.0143428,
          -0.0798274, -0.108905, -0.0276235, 0.0179013, -0.0499106,
          -0.077121, 0.0115964, -0.155183, -0.0646909, -0.00267948,
          -0.101105, -0.0768515)
    l3_c(0.469283, 0.467675, 0.116906, -0.066049, 0.0128097, 0.624553,
          -0.0574737, 0.258088, 0.616151, -0.0477033, 0.402842, 0.676726,
           0.032928, -0.118216, 0.264318, 0.339834, -0.213534, 0.865199,
           0.371532, -0.0850168, 0.581043, 0.395978)
    a3_cbind(l1,l2,l3)}

  }

}

# names of the variables

for(j in 1:p){
  rnam[j]_substring(rnam[j],1,2)
  }
rownames(a3)_rnam

ma_max(abs(a3))

# create the plot

par(mfrow=c(1,1),cex=1,cex.axis=1,cex.lab=1.5,xaxt="s",yaxt="s")

plot(a3[,1],type="n",xlim=range(0,1),ylim=range(-ma,ma),xlab="",ylab="")

        segments(0, 0, 1, 0)
        segments(0, 0.5, 1, 0.5, lty = 2)
        segments(0, -0.5, 1, -0.5, lty = 2)
  title(paste(xname,";",methodname,";",k,"cluster"))

 bb <- c(0,1)
 bb1 <- bb/k
 ba <- seq(from = bb1[1], by = bb1[2])
 ba1 <- ba[2]/20
 ba2 <- c(0,ba1)

 segments(0,-ma,0,ma)
```

```
for(i in 1:(k+1)){
        segments(ba[i],-ma,ba[i],ma)}


# create scaled weights
  xx_0
  ll_seq(from=1,to=k*(p+2),by=p+2)

  for(f in 1:k){
      xx[f]_a1[ll[f]]
    }

  sumweight_sum(as.numeric(xx))

#  for(j in 1:k){
#  w[j]_a1[ll[j]]/sumweight}

# text for fcm

if(m==1){
  for(j in 1:k){
  text(seq(from=ba[j]+ba[2]/p,to=ba[j+1]-ba[2]/p,
       by=(ba[j+1]-ba[j]-ba[2]/5)/(p-x)), a3[,j], rownames(a3),
       col="blue",cex=1.5)
  text(ba[j]+ba[2]/2,ma,paste("Cluster ",j,sep=""),col="red",cex=1.5)

  text(ba[j]+ba[2]/2,ma-0.1,paste("size= ",
       substring(as.numeric(a1[ll[j]])/sumweight,1,4),sep=""),
       col=5,cex=1.4)}
  }

# text for method gk

if(m==2){
  if(x==1 | x==2){

    for(j in 1:k){
      text(seq(from=ba[j]+ba[2]/p,to=ba[j+1]-ba[2]/p,
      by=(ba[j+1]-ba[j]-ba[2]/5)/(p-1)), a3[,j], rownames(a3),col="blue")
      text(ba[j]+ba[2]/2,ma,paste("Cluster ",j,sep=""),col="red")}
#     text(ba[j]+ba[2]/2,ma-0.05,paste("weight= ",
#     substring(as.numeric(a1[ll[j]])/sumweight,1,4),sep=""),col=5)
  }
```

```
  if(x==3){

    for(j in 1:k){
      text(seq(from=ba[j]+ba[2]/p,to=ba[j+1]-ba[2]/p,
           by=(ba[j+1]-ba[j]-ba[2]/5)/(p-3)), a3[,j], rownames(a3),col="blue")
      text(ba[j]+ba[2]/2,ma,paste("Cluster ",j,sep=""),col="red")}
#     text(ba[j]+ba[2]/2,ma-0.05,paste("weight= ",
#     substring(as.numeric(a1[ll[j]])/sumweight,1,4),sep=""),col=5)
       }
     }

}
```

Bibliography

C.C. Aggarwal, C. Procopiuc, J.L. Wolf, P.S. Yu, and J.S. Park. Fast algorithms for projected clustering. In *ACM SIGMOD Conference*, pages 61–72, Philadelphia, 1999. *http://cs.sungshin.ac.kr/∼ jpark/HOME/References/charu_sigmod99.ps.*

C.C. Aggarwal and P.S. Yu. Finding generalized projected clusters in high dimensional space. *Sigmond Record*, 29(2):70–92, 2000. *http://web.mit.edu/charu/www/gen.ps.*

R. Aggarwal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD Conference*, Seattle, 1998. *http://www.cs.cornell.edu/johannes/papers/1998/sigmod1998-clique.pdf.*

K. Al-Sultan. A tabu search approach to the clustering problem. *Pattern Recognition*, 28 (9):1443–1451, 1995. *http://www.dcc.unicamp.br/ic-tr-ftp/1997/97-13.ps.gz.*

M.R. Andenberg. *Cluster Analysis for Applications.* Academic Press, New York, 1973.

M. Ankerst, M. Breunig, H. Kriegel, and J. Sander. Optics: Ordering points to identify clustering structure. In *ACM SIGMOD Conference*, Philadelphia, 1999.

T.C Bailey and A.C. Gatrell. *Interactive spatial data analysis.* Longman Group Limited, Burnt Hill, England, 1995.

H. Bandemer. *Ratschläge zum mathematischen Umgang mit Ungewissheit.* B.G. Teuber, Leipzig, Germany, 1997.

H. Bandemer and W. Näther. *Fuzzy Data Analysis.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.

A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. II. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 29:786–801, 1999. *http://pc91066.cse.cuhk.edu.hk/TechDocs/Clustering/baraldi99b.pdf.*

D. Barbara and P. Chen. Using the fractal dimension to cluster datasets. In *6. ACM SIGKDD*, Boston, 2000. *http://www.ise.gmu.edu/∼dbarbara/kdd00.pdf.*

K.E. Basford and J.W. Tukey. *Graphical Analysis of Multiresponse Data.* Chapman & Hall/CRC, New York, 1999.

R.A. Becker, J.M. Chambers, and A.R. Wilks. *The New S Language*. Chapman & Hall, New York, 1988.

P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. *http://www.accrue.com/products/rp_cluster_review.pdf*.

J.C. Bezdek. Pattern recognition with fuzzy objective function. *IEEE Press, New York*, pages 483–495, 1981.

J.C. Bezdek and R.J. Hathaway. Some notes on alternating optimization. In R. Pal, N. and M. Sugeno, editors, *Advances in Soft Computing*, pages 288–300, Heidelberg, 2002. Springer.

J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 28(3), 1998.

H.H. Bock. Probability models and hypotheses testing in partitioning cluster analysis. In P. Arabie, L.J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 377–454. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

G. Box and D. Cox. An analysis of transformations. *Journal of the Royal Statistical Society*, Series B 26:211–252, 1964.

P.S. Bradley and U.M. Fayyad. Refining initial points for k-means clustering. In *Proc. 15th International Conf. on Machine Learning*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998. *ftp://ftp.research.microsoft.com/pub/dtg/fayyad/ml98/icml98.ps*.

P.S. Bradley, U.M. Fayyad, and O.L. Mangasarian. Mathematical programming for data mining: formulations and challenges. *INFORMS Journal on Computing*, 11(3):217–238, 1999. *ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-01.ps*.

D. Brown and C. Huntley. A practical application of simulated annealing to clustering. Technical report, IPC-TR-91-003, University of Virginia, 1991. *ftp://ftp.cs.virginia.edu/pub/techreports/IPC-91-03.ps.Z*.

C. Cheng, A. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *ACM SIGKDD*, San Diego, 1999. *http://www.cse.cuhk.edu.hk/~kdd/clustering/enclus.html*.

S. Chu, J.F. Roddick, and J.S. Pan. Efficient k-medoids algorithms using multi-centroids with multi-runs sampling scheme. Technical Report KDM-02-003, KDM Laboratory, Flinders University, Adelaide, South Australia, 2002. *http://www.cs.flinders.edu.au/People/John_Roddick/Papers/mdcrm02.ps*.

S.C. Chu, J.F. Roddick, and J.S. Pan. A comparative study and extensions to k-medoids algorithm. In *Fifth International Conference on Optimization: Techniques and Applications*, Hong Kong, China, 2001.

I. Clark. *Practical Geostatistics*. Appl. Science Publisher, London, 1979.

M.C. Cowgill, R.J. Harvey, and L.T. Watson. A genetic algorithm approach to cluster analysis. Technical report, Virginia Polytechnic Institute and State University, Virginia, 1998. *http://eprints.cs.vt.edu:8000/archive/00000495/01/TR-98-16.ps*.

J. Davis. *Statistics and Data Analysis in Geology*. John Wiley and Sons, Toronto, 1986.

W.H. Day. Complexity theory: An introduction for practitioners of classification. In P. Arabie, L.J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 199–234. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

G. De Soete. Optimal variable weighting for ultrametric and additive tree clustering. *Quality & Quality*, 20:169–180, 1986.

G. De Soete and J.D. Carrol. Tree and other network models for representing proximity data. In P. Arabie, L.J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 157–198. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

E. Dimitriadou, A. Weingessel, and K. Hornik. A combination scheme for fuzzy clustering. In R. Pal, N. and M. Sugeno, editors, *Advances in Soft Computing*, pages 288–300, Heidelberg, 2002. Springer.

B. Dom. An information-theoretic external cluster-validity measure, 2001. *http://www.almaden.ibm.com/cs/people/dom/rj10219.ps*.

S. Dudoit and Y. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002. *http://genomebiology.com/content/pdf/gb-2002-3-7-research0036.pdf*.

J.C. Dunn. A fuzzy relative of the isodata progress and its use in detecting compact well-separated clusters. *Cybernetics*, 3:32–57, 1974.

B.S. Duran and P.L. Odell. *Cluter Analysis*. Springer, Berlin, 1974.

T. Eckes and H. Rossbach. *Clusteranalysen*. Verlag W. Kohlhammer GmbH, Stuttgart, 1980.

M. Ester, H. Kriegel, J. Sander, and X. Xu. A density based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2. ACM SIGKDD*, Portland, 1996. *http://ifsc.ualr.edu/xwxu/publications/kdd-96.pdf*.

B. Everitt. *Cluster Analysis*. Heinemann Educational, London, 1974.

D. Fasulo. An analysis of recent work on clustering algorithms. Technical report, Univeristy of Washington: Department of Computer Science and Engineering, Seattle, 1999. *http://www.cs.washington.edu/homes/dfasulo/clustering.ps*.

U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996. *http://www.kdnuggets.com//gpspubs/aimag-kdd-overview-1996-Fayyad.pdf*.

P. Filzmoser. Statistische Analyse linguistischer Daten. Master's thesis, Inst. f. Statistik u. Wahrscheinlichkeitstheorie, Techn. Univ., Wien, 1993.

P. Filzmoser. *Principal Planes*. PhD thesis, Dept. of Statist. and Prob. Th., Univ. of Techn., Vienna, 1996.

C. Fraley and A.E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998. *http://www3.oup.co.uk/computer_journal/subs/ Volume_41/Issue_08/Fraley.pdf*.

Y.J.M. Fridlyand. *Resampling Methods for Variable Selection and Classification: Applications to Genomics*. PhD thesis, University of California, Berkeley, 2001. *http://stat-www.berkeley.edu/users/janef/Thesis/clustering.ps*.

J. Friedman and J.J. Meulman. Clustering objects on subsets of attributes, September 2001. *http://www-stat.stanford.edu/∼jhf/ftp/cosa.pdf*.

Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for the fuzzy c-means method. In *Proc. 5 th Fuzzy Syst. Symposium*, pages 247–250, Japanese, 1989.

J.C. Fullagar. *OpenGl*, 1994. *http://www.dojogame.com/Book/ogl3/ORM.pdf*.

V. Ganti, J. Gehrke, and R. Ramakrishnan. Cactus-clustering categorical data using summaries. In *5. ACM SIGKDD*, San Diego, 1999. *http://www.cs.cornell.edu/johannes/papers/1999/kdd1999-cactus.pdf*.

I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):773–781, 1989.

D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorial data: An approach based on dynamical systems. In *24. International Conference on Very Large Databases*, New York, 1998.

G. Glaeser and H. Stachel. *Open Geometry:* `OpenGL` *+ Advanced Geometry*. Springer Verlag, New York, 1999.

S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable supspace clustering for very large data sets. Technical report, Northwestern University, 1999. *http://www.ece.nwu.edu/∼harsha/mafia.ps*.

A.D. Gordon. Hierarchical classification. In P. Arabie, L.J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 65–122. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

A.D. Gordon. *Classification*. Chapman & Hall/CRC, Boca Raton, 2nd edition, 1999.

S. Guha, R. Rastogi, and K. Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 73–84, 1998. *http://www.cs.sfu.ca/CourseCentral/459/han/papers/guha98.pdf*.

S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000. *http://theory.stanford.edu/~sudipto/mypapers/categorical.pdf*.

D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. of the IEEE Conference on Decision and Control*, pages 761–766, 1979.

M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2,3):107–145, 2001. *http://www.db-net.aueb.gr/mhalk/papers/validity_survey.pdf*.

M. Halkidi and M. Vazirgiannis. Quality scheme assessment in the clustering process. In *Proceedings of PKDD*, Lyon, France, 2000. *http://www.db-net.aueb.gr/courses/postgrdb/cl_eval.pdf*.

M. Halkidi and M. Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *ICDM*, pages 187–194, 2001a. *http://www.db-net.aueb.gr/mhalk/papers/ halkvaz_ClusterValidity.pdf*.

M. Halkidi and M. Vazirgiannis. A data set oriented approach for clustering algorithm selection. *Lecture Notes in Computer Science*, 2168:165–183, 2001b. *http://www.db-net.aueb.gr/mhalk/papers/HV_PKDD01.pdf*.

J. Hartung und B. Elpelt. *Multivariate Statistik. Lehr- und Handbuch der angewandten Statistik*. Oldenbourg Verlag, München, 1986.

T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

A. Hinneburg and D. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Conference on VLBD*, Edinburgh, Scotland, 1999. *http://www.acm.org/sigmod/disc/protected/papers/vldb/optimalgridclusalda.pdf*.

A. Hinneburg, D.A. Keim, and W. Brandt. Clustering techniques for large data sets - from the past to the future, 2000. *http://www.informatik.uni-halle.de/~keim/PS/ClustTutPKDD2000.pdf*.

F. Höppner. *Fuzzy Clustering Algorithms - A Tool Library*, 2000. *http://www.fuzzy-clustering.de*.

F. Höppner and F. Klawonn. A new approach to fuzzy partitioning. In *Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 1419–1424, Vancover, 2001. *http://public.rz.fh-wolfenbuettel.de/∼hoeppnef/ paper/Hoeppner-NAFIPS-2001.pdf.*

F. Höppner, F Klawonn, and R. Kruse. *Fuzzy- Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse.* Vieweg Verlagsgesellschaft, Braunschweig, 1996.

R.J. Howarth. *Statistics and Data Analysis in Geochemical Prospecting.* Elsevier Scientific Publishing Company, Amsterdam, 1983.

A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999. *http://l2r.cs.uiuc.edu/∼cogcomp/AIML/papers/SPRING2003/jain99data.pdf.*

D. Johnson, L. Aragorn, L. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation. *Journal of Operations Research*, 37(6):865–893, 1989. *http://www.cs.uic.edu/∼jlillis/courses/ cs594_f02/JohnsonSA.pdf.*

R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis.* Prentice-Hall, London, 4th edition, 1998.

R.H.G. Jongman, C.J.F. Ter Braak, and O.F.R van Tongeren. *Data Analysis in Community and Landscape Ecology.* Pudoc, Wageningen, 1987.

G. Karypis, R. Aggorwal, V. Kumar, and S. Shekar. Multilevel hypergraph partitioning: application in vlsi domain. In *ACM/IEEE Design Automation Conference*, 1997. *http://www.iro.umontreal.ca/∼poirierg/cs741/karypis.pdf.*

L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data.* Wiley & Sons, New York, 1990.

G.N. Lance and W.T. Williams. A general theory of classificatory sorting strategies. hierarchical systems. *Computer Journal*, 9:373–380, 1966.

C. Lee and E. Antonsson. Dynamic partitional clustering using evolution strategies. In *3. Asia-Pacific Conference of Simulated Evolution and Learning*, Nagoya, 2000. *http://www.design.caltech.edu/Research/Publications/.*

P. Legendre and L. Legendre. *Numerical Ecology.* Book News, Portland, 1998.

C.B. Lucasius, A.D. Dane, and G. Kateman. On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparsion. *Analytica Chimica Acta*, pages 647–669, 1993.

J.H. Maindonald. *Using R for Data Analysis and Graphics*, 2002. *http://wwwmaths.anu.edu.au/∼johnm/.*

V. Makarenkov and P. Legendre. Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. *Journal of Classification*, 18:245–271, 2001. *http://www.fas.umontreal.ca/BIOL/legendre/reprints/ Optimal_Variable_Weighting.pdf*.

K. McGarigal, S. Cushman, and S. Stafford. *Multivariate Statistics for Wildlife and Ecology Research*. Springer Verlag, New York, 2000.

G.W. Milligan. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45:325–342, 1980.

G.W. Milligan. Clustering validation: Results and implications for applied analysis. In P. Arabie, L.J. Hubert, and G. Soete, editors, *Clustering and Classification*, pages 341–375. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

R. Ng and J. Han. Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016, 2002. *http://www.computer.org/tkde/tk2002/k1003abs.htm*.

A. Okada. A review of cluster analysis research in japan. In P. Arabie, L.J. Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 271–294. World Scientific Publishing Co. Pte. Ltd., Singapore, 1996.

C. O'Muircheartaigh and C. Payne. *Exploring Data Structures*. John Wiley & Sons, New York, 1977.

E.C. Pielou. *The Interpretation of Ecological Data: A Primer on Classification and Ordination*. John Wiley and Sons, New York, 1984.

L. Pirktl. *Probleme und Algorithmen der Clusteranalyse*. PhD thesis, Eidgenössische Technische Hochschule, Zürich, 1983.

S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD*, pages 761–766, 2000. *http://www.bell-labs.com/projects/serendip/Papers/outliers.ps.gz*.

C. Reimann. Externe laborkontrolle für umweltbezogene geochemieprojekte - notwendigkeit oder übertriebener aufwand? *Mitteilung der Österreichischen Geologischen Gesellschaft, Umweltgeologieband*, 79:175–191, 1986.

C. Reimann. *Chemical Elements in the Environment*. Springer, Berlin, 1998.

C. Reimann, M. Äyräs, V. Chekushin, I. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T.E. Finne, J.H. Halleraker, Ø. Jæger, G. Kashulina, O. Lehto, H. Niskavaara, V. Pavlov, M.L. Räisänen, T. Strand, and T. Volden. *Environmental Geochemical Atlas of the Central Barents Region*. Geological Survey of Norway (NGU), Geological Survey of Finland (GTK), and Central Kola Expedition (CKE), Special Publication, Trondheim, Espoo, Monchegorsk, 1998.

C. Reimann and P. Filzmoser. Normal and lognormal data distribution in geochemistry: death of a myth. consequences for the statistical treatment of geochemical and environmental data. *Environmental Geology*, 39(9):1001–1014, 1999.

C. Reimann, P. Filzmoser, and R.G. Garret. Factor analysis applied to regional geochemical data: problems and possibilities. *Applied Geochemistry*, 17:185–206, 2000.

B. Ripley. *R Data Import/Export*, 2002. *http://cran.r-project.org/doc/manuals/R-data.pdf*.

N.M. Rock. *Numerical Geology*. Springer-Verlag, Berlin, 1988.

A.J. Rossini, M. Mächler, K. Hornik, R.M. Heiberger, and R. Sparapani. *Emacs Speaks Statistics: A Universal Interface for Statistical Analysis*, 2001. *http://software.biostat.washington.edu/statsoft/ess/ess-techrep.pdf*.

P.J. Rousseeuw and B.C. van Zomeren. Unmasking multivariate outliers and leverage points. *J. Amer. Statist. Assoc.*, 85(411):633–651, 1990.

E.H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22–32, 1969.

E. Salazar, A.C. Velez, C.M. Parra, and O. Ortega. A cluster validity index for comparing non-hierarchical clustering methods. Universidad de Antioquia, June 2002. *http://bochica.udea.edu.co/%7Eoortega/papers/eiti2002-1.pdf.gz*.

J. Sander, M. Ester, H. Kriegel, and X. Xu. Density-based clustering in spatial databases: the algorithm gdbscan and its application. *Data Mining and Knowledge Discovery*, 2(2): 169–194, 1998. *http://www.cs.ualberta.ca/∼joerg/papers/GDBSCAN.pdf*.

M. Schader. *Scharfe und unscharfe Klassifikation qualitativer Daten*. Verlag Anton Hein Meisenheim GmbH, Meisenheim, Germany, 1981.

E. Schikuta. Grid-clustering: a fast hierarchical clustering method for very large data sets. In *13. International Conference on Pattern Recognition*, 1996. *http://softlib.rice.edu/pub/CRPC-TRs/reports/CRPC-TR93358.ps.gz*.

E. Schikuta and M. Erhart. The bang-clustering system: grid-based data analysis. In *Advances in Intelligent Data Analysis, Reasoning about Data*, London, UK, 1997. *http://www.pri.univie.ac.at/∼schiki/research/paper/ida97/ida97.ps*.

J. Schöll. *Clusteranalyse mit Zufallswegen*. PhD thesis, TU-Wien, 2002.

G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multiresolution clustering approach for very large spatial databases. In *24. Conference on VLBD*, New York, 1998. *http://www.ece.nwu.edu/∼harsha/Clustering/vldb98.ps* .

A.R.H. Swan and M. Sandilands. *Introduction to Geological Data Analysis*. Blackwell Sience, Oxford, 1995.

Y. Tai-Ning and W. Sheng-De, editors. *Fuzzy algorithms for Robust Clustering and C-Spherical Shells Algorithms*, Dong, Repuplic of Korea, 2002. ICS, Workshop on AI, National Dong Hwa University.

S. Thomopoulos, D. Bougoulias, and C. Wann. Dignet: An unsupervised-learning clustering algorithm for clustering and data fusion. *IEEE Trans. on Aerospace and Electr. Systems*, 31(1):1–38, 1995.

R Tibshirani, G Walther, D Botstein, and P Brown. Cluster validation by prediction strength, 2001. *http://www-stat.stanford.edu/∼tibs/ftp/predstr.ps*.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. Technical report, Dept. of Statistics, Stanford University, 2000. *http://www-stat.stanford.edu/∼tibs/ftp/gap*.

H. Timm. *Fuzzy-Clusteranalyse: Methoden zur Exploration von Daten mit fehlenden Werten sowie klassifizierten Daten*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Germany, 2002. *http://fuzzy.cs.uni-magdeburg.de/∼htimm/data/dissertation.pdf*.

A.K.H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. In *17. ICDE*, Heidelberg, Germany, 2001. *http://www.cs.ualberta.ca/simzaiane/postscript/ideas02_zaiane.pdf*.

A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

B. Venables and D.M. Smith. *An Introduction to R*, 2002. *http://cran.r-project.org/doc/manuals/R-intro.pdf*.

W.N. Venables and B.D. Ripley. *Modern applied statistics with* `S-PLUS`. Springer Verlag, New York, 1994.

J. Verzani. *simpleR*, 2002. *http://www.math.csi.cuny.edu/Statistics/R/simpleR/index.html*.

H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets. In *SIGMOD Conference*, 2002. *http://www.cis.ohio-state.edu/∼hakan/CIS888/SIGMOD02_2.pdf*.

W. Wang, J. Yang, and R.R. Muntz. STING: A statistical information grid approach to spatial data mining. In M. Jarke, M.J. Carey, K.R. Dittrich, F.H. Lochovsky, P. Loucopoulos, and M.A. Jeusfeld, editors, *Twenty-Third International Conference on Very Large Data Bases*, pages 186–195, Athens, Greece, 1997. Morgan Kaufmann. *http://www.math.tau.ac.il/∼matias/courses/papers/STING.ps*.

D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

T. Williams and C. Kelley. `Gnuplot`*: an interactive plotting program*, 1998. *http://www.comnets.rwth-aachen.de/doc/gnu/gnuplot37/gnuplot.html*.

X.L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):841–846, 1991.

X. Xu, M. Ester, H. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining large spatial data sets. In *14. ICDE*, Orlando, 1998. *http://ifsc.ualr.edu/xwxu/publications/icde-98.pdf.*

J. Yang, W. Wang, H. Wang, and P.S. Yu. δ-cluster: Capturing subspace correlation in a large data set. In *ICDE*, 2002. *http://www.cs.ucla.edu/∼jyang/paper/ICDE02_1.ps.*

K. Yeung and W. Ruzzo. An empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001. *http://www.cs.washington.edu/homes/ ruzzo/papers/pca-bioinf.pdf.*

T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, June 1996. *http://www-courses.cs.uiuc.edu/∼cs497jh/papers/zhang96.pdf.*

H.J. Zimmermann. *Fuzzy Set Theorie and its Application.* Kluwer Academic Publishers, Norwell, Massachusetts, 1991.