PIK Report

No. 126

CLUSTER ANALYSIS TO UNDERSTAND SOCIO-ECOLOGICAL SYSTEMS: A GUIDELINE

Peter Janssen, Carsten Walther, Matthias Lüdeke

P I K

POTSDAM INSTITUTE
FOR
CLIMATE IMPACT RESEARCH (PIK)

This report is the result of a joint study between the PBL Netherlands Environmental Assessment Agency and PIK.

Authors:
Dipl.-Phys. Carsten Walther
Dr. Matthias Lüdeke
Potsdam Institute for Climate Impact Research
P.O. Box 60 12 03, D-14412 Potsdam, Germany
Dr. Peter Janssen *
Netherlands Environment Assessment Agency (PBL)
P.O. Box 1, 3720 BA Bilthoven, The Netherlands
E-Mail: peter.janssen@pbl.nl
* corresponding author

Herausgeber:
Prof. Dr. F.-W. Gerstengarbe

Technische Ausführung:
U. Werner

POTSDAM-INSTITUT
FÜR KLIMAFOLGENFORSCHUNG
Telegrafenberg
Postfach 60 12 03, 14412 Potsdam
GERMANY
Tel.: +49 (331) 288-2500
Fax: +49 (331) 288-2600
E-mail-Adresse:pik@pik-potsdam.de

ISSN 1436-0179 POTSDAM, SEPTEMBER 2012

Abstract

In coupled human-environment systems where well established and proven general theories are often lacking cluster analysis provides the possibility to discover regularities – a first step in empirically based theory building. The aim of this report is to share the experiences and knowledge on cluster analysis we gained in several applications in this realm helping to avoid typical problems and pitfalls. In our description of issues and methods we will highlight well-known main-stream methods as well as promising new developments, referring to pertinent literature for further information, thus offering also some potential new insights for the more experienced. The following aspects are discussed in detail: data-selection and pre-treatment, selection of a distance measure in the data space, selection of clustering method, performing clustering (parameterizing the algorithm(s), determining the number of clusters etc.) and the interpretation and evaluation of results. We link our description – as far as tools for performing the analysis are concerned - to the R software environment and its associated cluster analysis packages. We have used this public domain software, together with own tailor-made extensions, documented in the appendix.

Contents

Acknowledgements

This report is the result of a joint study between the PBL Netherlands Environmental Assessment Agency and PIK, as part of a wider research effort to quantitatively analyse patterns of vulnerability. The authors like to acknowledge the feedback and inputs of Henk Hilderink, Marcel Kok, Paul Lucas, Diana Sietz, Indra de Soysa and Till Sterzel provided during the course of the study.

1 Introduction

Cluster analysis is a general methodology for exploration of datasets when no or little prior information is available on the data's inherent structure. It is used to group data into classes (groups or clusters) that share similar characteristics, and is widely used in behavioural and natural scientific research for classifying phenomena or objects under study without predefined class-definitions. In particular in coupled human-environment systems where well established and proven general theories are still lacking cluster analysis provides the possibility to discover regularities – a first step in empirically based theory building. A recent example is the application for assessing the vulnerability of human wellbeing against global change (Sietz et al., 2011 and Kok et al., 2010). The aim of this report is to share the experiences and knowledge on cluster analysis we gained in these applications helping to avoid typical problems and pitfalls.

A broad collection of clustering methods has been proposed in areas as statistics, data mining, machine learning, bioinformatics, and many textbooks and overview papers illustrate the variety of methods as well as the vigorous interest in this field over the last decade with the growing availability of computer power for analysing extensive datasets or data objects involving many attributes (i.e. finding clusters in high-dimensional space, where the data points can be sparse and highly skewed). Books on cluster analysis, there are many: e.g. Aldenderfer and Blashfield (1976), Jain and Dubes (1988), Kaufman and Rousseeuw (1990), Gordon (1999), Hastie et al. (2001), Everitt, Landau and Leese, 2001, Mirkin (2005); Xu and Wunsch (2009). The same holds for overview papers, see e.g. Jain, Murty and Flynn (1999), Omran, Engelbrecht, Salman (2005), Xu and Wunsch (2005), Wunsch and Xu (2008).

In this report we will highlight the major steps in the cluster analysis process, and link it – as far as tools for performing the analysis are concerned - to the R software environment and its associated cluster analysis packages (see appendix A and B). We have used this public domain software, together with own tailor-made extensions, to perform cluster analysis for identifying patterns of vulnerability to global environmental change (Kok et al. 2010), as part of a joint study of the PBL Netherlands Environmental Assessment Agency, PIK and the Norwegian University of Science and Technology. Examples from this study will be used as illustrative material in the present report.

Beyond this specific background, the report is set up in more general terms, and can be used by novices in the field of cluster analysis, as well as by people who have already some working experience with the method but want to extend their ability to perform cluster analyses.

In our description of issues and methods we will highlight well-known main-stream methods as well as promising new developments, referring to pertinent literature for further information, thus offering also some potential new insights for the more experienced. We do not extensively consider cluster analysis methods which explicitly account for spatial and/or temporal aspects of the data, but only briefly touch upon them.

1.1 Outline of the report

Our exposition is for an important part based on the excellent book of Everitt, Landau and Leese, 2001 on clustering and on Han and Kamber's book on data mining, which contains a concise chapter on cluster analysis (Han and Kamber, 2006, chapter 7). In discussing cluster analysis we will divide the clustering-process into a number of logical steps:

- ***Data-selection and pre-treatment:*** In its generality this concerns the selection of data of interest for the problem at hand and the treatment of missing values and outliers. Optionally it also involves dimension-reduction by selecting variables or extracting relevant features from the data, the use of data transformations to bring the data values to a more even scale and the standardization of data to make them mutually more comparable. These forms of data-processing can influence the outcomes of the clustering to a large extent, and should therefore be chosen with due consideration.
- ***Selection of a distance measure in the data space****:* In order to express the similarity or dissimilarity between data points a suitable distance measure (metric) should be chosen. It forms the basis for performing the clustering to identify groups which are tightly knit, but distinct (preferably) from each other (Kettenring, 2006). Often Euclidean distance is used as a metric, but various other distance measures can be envisioned as well.
- ***Selection of clustering method:*** The extensive – and ever-growing - literature on clustering illustrates that there is no such thing like an optimal clustering method. We will group the multitude of methods into a restricted number of classes, and will especially focus on two commonly used classes, one which is based on *hierarchically* performing the clustering, while the other consists of constructively *partitioning* the dataset into a number of clusters, using the *k-means* method. The other classes will be briefly discussed with due reference to literature for further information.
- ***Performing clustering:*** This involves parameterising the selected clustering algorithm(s) (e.g. choosing starting points for the partitioning method), determining the number of clusters, and computing the resulting clustering partition for these settings. Especially the issue of determining the *number of clusters* is an important one, and we will highlight a general approach which we applied for our vulnerability assessment study.
- ***Interpretation and evaluation of results:*** This concerns in the first place a *description* of the clustering in terms of cluster characteristics. Moreover - in order to use the clustering results - the characteristics and meaning of the various clusters have to be *interpreted* in terms of content matters, which often involve a process of knowledge building, hypothesis setting and testing, going back and forth from the clustering results to the underlying knowledge base.
  Finally, evaluation includes also a study of the *sensitivity* of the clustering results for the various choices during the various steps of the cluster analysis, e.g. concerning the data selection and pre-treatment, selection of clustering method etc. Also the effects of uncertainties and errors in the data should be addressed in this step.

The various steps are described in more detail in the following chapters. In the appendices more detailed information is given on the R software and on some specific clustering issues.

Clustering in various contexts (according to Han and Kamber, 2006):
As a branch of *statistics*, cluster analysis has been extensively studied, with a focus on distance-based cluster analysis. Cluster analysis tools based on k-means, k-medoids, hierarchical clustering and several other methods have been build into many software packages for statistical analysis such as S-Plus, SPSS and SAS. Also dedicated software (e.g. Wishart's CLUSTAN (http://www.clustan.com/index.html), Matlab Statistics toolbox) and public-domain packages abound (see the various R-packages on clustering).
In the *machine learning* context, clustering is an example of *unsupervised learning*, which does not rely on predefined classes and class-labeled training data. It is a form of learning by observation, rather than learning by examples as in supervised learning (as e.g. in data-classification).
In the *data mining* field efforts have focused on finding methods for efficient and effective analysis of large databases. Issues as the scalability of clustering methods, the ability to deal with mixed numerical and categorical data, complex shapes and types of data, high-dimensionality, the ability to deal with noisy data, to incorporate domain knowledge, to easily deal with updates of the databases, insensitivity to the order of input records, are important requirements for the clustering methods.

2 Data selection and pre-treatment

The main theme in cluster analysis is to identify groups of individuals or objects (i.e. ‘cases’ or ‘entities’) that are similar to each other but different from individuals or objects in other groups. For this purpose *data* on the individuals or objects have to be *collected,* and it is obvious that the data should be characteristic, relevant and of good quality to enable a useful analysis.

2.1 Data-collection: Some important issues

This means in the first place that an *adequate number* of *objects/cases/individuals* should be available in the dataset to study the phenomena of interest (e.g. identifying situations that show a similar reaction pattern under certain environmental stresses; identifying subgroups of patients with a diagnosis of a certain disease, on basis of a symptom checklist and results from medical tests; identifying people with similar buying patterns in order to successfully tailor marketing strategies etc.).
Moreover the researcher should choose the relevant *variables/features* which characterize the objects/cases/individuals on basis of which the groups should be subdivided in homogeneous subgroups. Milligan, 1996 strongly advices to be on the parsimonious side and *‘select only those variables that are believed to help discriminate the clustering in the data’*. Adding *‘ only one or two irrelevant variables can dramatically interfere with cluster recovery’* (Milligan, 1996).
For further analysis one must also decide - amongst others - whether to *transform* or *standardize* the variables in some way so that they all contribute equally to the distance or similarity between cases.
Furthermore *data quality* will be another important issue which involves various aspects as e.g. accuracy, completeness, representativeness, consistency, timeliness, believability, value added, interpretability, traceability and accessibility of the data, presence of noise and outliers, missing values, duplicate data etc. (cf. Pipino, Funk, Wang (2006)).

2.2 Data-collection: Type of data

An important distinction when considering the data that has been collected on the ‘objects’ and their ‘attributes’[1] (i.e. properties or characteristics of an object; e.g. eye colour of a person, length, weight) is the *(measurement) scale* which has been used in expressing these attributes:

- *Nominal scale*: In fact this is not really a scale because numbers are simply used as identifiers, or names, e.g. in coding a (no, yes) response as (0,1). The numbers as such are mostly meaningless in any quantitative sense (e.g. ID numbers, eye colour, zip codes).

[1] Concerning terminology: ‘attributes’ are also referred to as variables, features, fields, characteristics. A collection of attributes describes an ‘object’. An object is also known as record, point, case, sample, entity or instance. These terms are often used interchangeably.

- *Ordinal scale*: The numbers have meaning only in relation to one another, e.g. the scales (1, 2, 3), (10, 20, 30) and (1, 20, 300) are in a sense equivalent from an ordinal viewpoint. Examples of ordinal scale attributes are rankings, grades, or expressing height in {tall, medium, short}-categories.
- *Interval scale*: This scale is used to express data in a (continuous) measurement scale where the *separation* between numbers has meaning. A unit of measurement exists and the interpretation of the numbers depends on this unit (compare temperature in Celsius or in Fahrenheit).
- *Ratio scale*: This is a measurement scale where an absolute zero exist and a unit of measurement, such that the ratio between two numbers has meaning (e.g. distance in meters, kilometres, miles or inches).

The first two scales refer more to qualitative variables, and the latter to quantitative variables[2]. In practice, the attributes characterizing an object can be of mixed type.

Another distinction can be made between *'discrete'* and *'continuous'* attributes, where the first category refers to variables having a finite or countably infinite set of values (e.g. zip-code), and can often be represented as integer variables (1, 2, 3, …). Binary attributes, taking on the values 0, 1, or "No", "Yes" are a special case of discrete attributes. Continuous attributes can take values over a continuous range, and have real numbers as attribute values. Notice that in practice real values can only be measured and represented using a finite number of digits.

2.3 Data pre-processing

Since real data can be incomplete (missing attribute values), noisy (errors or outliers) and inconsistent (e.g. duplicates with different values), ***data pre-processing*** is an indispensable part of the cluster analysis. The major tasks involved in data pre-processing are:

- **[A] Data cleaning:** Filling in missing values, smoothing noisy data, identifying or removing outliers, correcting inconsistencies and resolving redundancies caused by integration or merging of data from various sources/databases.
- **[B] Data integration:** Integration of multiple databases, files or data cubes (data structures commonly used to describe time series of image data).
- **[C] Data transformation:** Putting data in form(at)s which are appropriate for further analysis. This includes normalization and performing summary or aggregation operations on the data, for instance.
- **[D] Data reduction:** Obtaining reduced representation in volume of the data that produce the same or similar analytical results**.**
- **[E] Data discretization:** Especially for numerical data this denotes a specific form of data reduction.
- **[F] Cluster tendency:** Determining whether there are clusters in the data.
- **[G] Cluster visualisation:** Using graphical techniques can greatly enhance the analysis of the underlying cluster/group-structure in the data.

[2] We restrict our attention to data which have numerical values, and don't consider symbolic objects. See e.g. Ravi and Gowda (1999) for cluster analysis of this category of objects.

In the sequel we will outline these activities in more detail:

2.3.1 Data cleaning

Various techniques for performing data-cleaning can be used, of which we only briefly discuss the way *missing data* and *outliers* can be handled. Additional dedicated methods for data cleaning originating from the data warehouse literature can e.g. be found in Rahm and Do (2000).

(i) Handling missing data
Values can be missing since information is not collected or attributes are not applicable in all cases (e.g. annual income for children). One obvious way of handling missing data is simply eliminating the corresponding data objects, and analysing only that part of the dataset which is complete (called *marginalization* by Wagstaff and Laidler, 2005). This strategy does not lead to the most efficient use of the data and is recommended only in situations where the number of missing values is very small. Another option (called *imputation*) to deal with missing data is to replace the missing values by a global constant (e.g. 'unknown', a new class) or by an estimate, e.g. the mean, median, a most probable value; cf. various forms of data-imputation (e.g. *mean, probabilistic* or *nearest neighbourhood* imputation[3], as presented in Wagstaff and Laidler, 2005).

Jain and Dubes (1988, page 19-20)) recommend - on basis of experimental results of Dixon (1979) - to use an imputation approach which redefines the distance between data points x_i and x_k which contain missing values as follows: First define the distance d_j between the two points along the *j*-the feature as $d_j=0$, if x_{ij} or x_{kj} is missing, and x_{ij}-x_{kj} otherwise, then the distance between x_i and x_k is defined as: $d_{ik} = \frac{m}{m - m_o} \sum d_j^2$ where m_o is the number of features missing in x_i or x_k or both, and m is the total number of features. d_{ik} as defined above is the squared Euclidean distance in case there are no missing values.

Wagstaff and Laidler (2005) notice that in some applications imputation and marginalization is not suitable since the missing values are physically meaningful and should not be supplemented or discarded. They implemented an algorithm, called *KSC* (*K-means* with soft constraints) that is dealing with the whole data set including the partially measured objects.
Additional information on dealing with missing values can be found in Little & Rubin (1987).

(ii) Smoothing noisy data
Noisy data are caused by (random) error or variance in a measured variable, as well as incorrect attribute values due to faulty data collection instruments, data entry and transmission problems, inconsistencies in naming convention etc. In case of noisy

[3] 'Mean imputation' involves filling the missing values with the mean of the remaining ones, while 'probabilistic imputation' consists of filling it with a random value drawn from the distribution of the feature. 'Nearest neighborhood imputation' replaces it with value(s) from the nearest neighbor.

data one can decide to filter/smooth them first in order to partially remove some of the effects of the noise. E.g. *binning*, which consists of first sorting the data and then partitioning them into (equal frequency) bins and subsequently smoothing them by replacing them by their bin means, medians or bin boundaries, is a simple way of filtering the data. More advanced approaches, like using e.g. regression analysis, trend-detection or noise-filtering (applying e.g. moving averages) can also be invoked to partially remove noise from the data.

(iii) Handling outliers

Outliers are data values that are extremely large or small relative to the rest of the data. Therefore they are suspected to misrepresent the population from which they were collected. Outliers may be the result of errors in measurements, model-results, data-coding and transcription, but may also point to (often unexpected) true extreme values, indicating more variability in the population than was expected. Therefore, in treating outliers one has to be cautious not to falsely remove outliers when they characterize important features (e.g. hotspots) of the phenomenon at hand; it is obvious that the decision to discard an outlier should not be based solely on a statistical test but should also be taken on basis of scientific and quality assurance considerations.

The *first step* in handling outliers consists of the *detection of outliers* (see also Rousseeuw et al. 2006). Though detecting outliers can partly be based on process-information and combined computer and human inspection of graphical representations of the data, one often relies on statistical techniques. Hubert and Van der Veeken (2008) recently proposed a statistical technique which is especially suited for detecting outliers in *skew distributed* multivariate data and is also related to the adjusted boxplot for skew distributed data (Hubert and Vandervieren (2008)). Though several more refined robust estimators and outlier detection methods exist which are typically geared to specific classes of skewed distributions, their approach is very useful when no prior information about the data distribution is available, or when an automatic and fast outlier detection method is required. In the CRAN-package <<*robustbase*>>[4] functionality is available for this form of outlier detection (function <<*adjOutlyingness*>>) as well as for the adjusted box-plot determination (function <<*adjbox*>>).

The *second step* involves the *pre-treatment of outlier-values* before performing cluster analysis. In general three general strategies can be applied: (a) *using the outlying* data points in the subsequent analysis, accounting for their effects on the outcomes; *(b) trimming:* removing the outlier data from the data set, and not incorporating them in the dataset for the subsequent cluster analysis; *(c) winsorising:* replacing the outlying values by a truncated variant, e.g. a specific percentile (e.g. the 1th or 99th percentile) of the dataset, or an associated cut off-value of the skewed boxplot (Hubert and Van der Veeken, 2008). These truncated data points are included in the cluster analysis.

The above procedure is in fact centred around detecting outlying values with respect to an (supposedly) underlying distribution of the attribute-dataset, *before* the cluster analysis takes place. There is however also the issue of detecting outliers with respect to the obtained partition of the objects into clusters, i.e. *after* the cluster analysis has been performed:

[4] Cf. http://cran.r-project.org/web/packages/robustbase/

- Irigoien and Arenas (2008) recently proposed a geometrically inspired method for detecting potential atypical outlying data-points.
- Also the Silhouette statistic proposed by Rousseeuw (1987) can be used as an indication of the outlyingness of a point in a cluster. It measures how well a certain data point/object, say *i*, is matched to the other points/objects in its *own* cluster, versus how well matched it would be, if it were assigned to the *next closest* cluster. The Silhouette of *i* is expressed as *s(i)=[b(i)-a(i)]/max[a(i),b(i)],* where *a(i)* denotes the average distance between the *i*-th point and all other points in its cluster, and *b(i)* is the average distance to points in the "nearest" clusters with nearest being defined as the cluster minimizing *b(i). s(i)* is a value between -1 and +1, and large (positive) values indicate strong clustering, while negative values indicate that clustering is bad. See e.g. Figure 1 which gives an example of a Silhouette plot, as well as the associated 2-dimensional projection of the cluster points. The Silhouette statistic can e.g. be calculated with the function <silhouette> in the CRAN-package <<cluster>>[5].

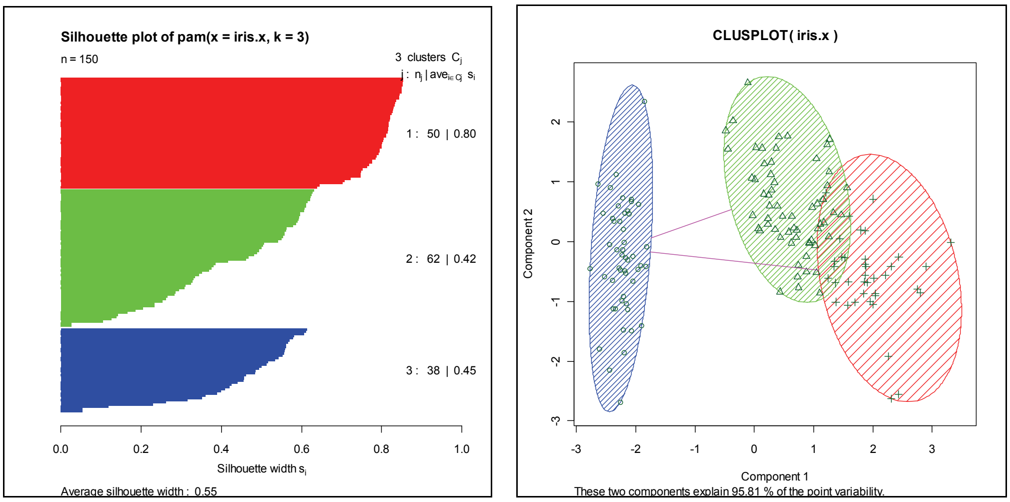


Figure 1: An example of a Silhouette plot for a cluster analysis with three clusters. The plot expresses the (ordered) silhouette values for the points in the three clusters. It shows that most points in the first cluster have a large silhouette value, greater than 0.6, indicating that the cluster is somewhat separated from neighbouring clusters. The second and third cluster contain also several points with low silhouette values indicating that those two clusters are not well separated, as exemplified in the 2-dimensional cluster plot in the right frame.

The R-commands for constructing these results are:

```
## Partitioning iris-data (data frame) into 3 clusters,
## and displaying the silhouette plot.
## Moreover a 2-dimensional projection of the partitioning is given.

library(cluster)        # Load the package cluster
data(iris)              # Load the famous (Fisher's or Anderson's) iris-dataset
iris.x <- iris[, 1:4]   # Select the specific datacolumns: i.e. Sepal.Length,
Sepal.Width, Petal.Length, Petal.Width

pr3 <- pam(iris.x, 3)   # Perform the clustering by the PAM-method with 3 clusters
si<-silhouette(pr3)     # Compute the Silhouette information for the given
clustering
```

[5] Cf. http://cran.r-project.org/web/packages/cluster/

```
plot(si, col = c("red", "green", "blue")) # draw a silhouette plot with
clusterwise coloring

clusplot(iris.x, pr3$clustering, shade=TRUE,color = TRUE, col.clus= c("red",
"green", "blue")) # draw a 2-dimensional clustering plot for the given clustering
```

For more information on outlier-detection and analysis we refer to section 7.11 in Han and Kamer, 2006, who distinguish 4 different approaches to outlier analysis: statistical distribution-based, distance-based, density-based local outlier detection and the deviation-based approach.

2.3.2 Data integration

When integrating multiple data-sources (databases, files or data-cubes) redundant data can occur, since e.g. the same attribute or object may have different names in different sources, or one attribute may be a 'derived' attribute in another source (e.g. annual values, instead of monthly values). Correlation analysis can e.g. be used to point at potential redundancies in the data, while additional post-processing (e.g. *data-reduction*; see later) can be used to alleviate their effects.
In data integration one should also be aware of potential data value conflicts which can occur when attribute values from different sources are different e.g. due to different representations or scales. These problems can be avoided by carefully performing and checking the data integration.

2.3.3 Data transformation

Data transformation first of all includes *normalization* of the data to bring them into a form which is more amenable for the subsequent analysis. It is well-known that measurement scale can have a large effect in performing cluster analyses, as illustrated in Figure 5 of Kaufman and Rousseeuw, 1990 or in Silver, 1995. Therefore it is considered important to bring the data into a form which is less dependent on the choice of measurement/representation scale. A typical standardization (the "*(min,max)*-range standardization") which is used for this purpose consists of determining the range of values[6] and redefinining the value of *X(i)* by: *(X(i)-min)/(max-min),* thus obtaining values between 0 and 1, where 0 and 1 refer to the extreme values (i.e. min and max[7]). Other statistical transformations, like the Z-transform - which replaces *X(i)* by *(X(i)-mean)/stdev*, with *mean* being the average value, and *stdev* the standard deviation of all data-values *X(i)* - are also conceivable, but are considered less apt when performing cluster-analysis (cf. Milligan and Cooper, 1988, Kettenring, 2006).

Remark: Though the (min,max) standardization has the function of transforming the variables into a comparable format, some caution is due in using it. E.g. in situations where certain variables are already measured in a commensurable scale, applying this additional standardization can result in an artificial rescaling of the variables which obscures their actual differences. E.g. when the actual min-max ranges differ (e.g. the actual values for variable A

[6] This can e.g. be the actual range, consisting of the actual maximum-minimal value of the current data, or the maximal feasible range one can think of (i.e. beyond the actual data-sample).

[7] The min and max can here refer to the actual minimum and maximum of the dataset at hand, but can also refer to the feasible minimum and maximum which can realistically be expected, and which can be smaller (for the minimum) or larger (for the maximum) than the actual ones.

range from .2 to .25, while those of variable B range from .015 to .8), rescaling on basis of the actual min-max range will result for both variables in values running from 0 to 1 which renders a very different (and erroneous) view on their difference. In this situation one could argue for not automatic rescaling these variables, but proceed with the unscaled version. However, one can as easily argue against this, by stating that the use of an unscaled version for these variables will result in an unfair bias towards other variables which have been re-scaled into the complete (0, 1) range by applying the (min,max) standardization. What choices will be made in the end will depend on what is considered important. This situation in fact asks for a sensitivity analysis to study what effects the applied alternative standardization options can possibly have on the clustering results.

Another issue concerns the use of *non-linear transformations* on the variables to bring them into a form which e.g. fits more to the underlying assumptions: e.g. a right-skewed distribution could possibly be transformed into approximately Gaussian form by using logarithmic or square-root transformation, to make the data more amenable to statistical techniques which are based on normality assumptions. In analyzing these transformed data one should however realize that re-interpretation of the obtained results in terms of the original untransformed data requires due care, since means and variances of the transformed data render biased estimates when transformed back to the original scale. Therefore, if the nonlinear transformations of the data are expected to have no noticeable benefits for the analysis, it is usually better to use the original data with a more appropriate statistical analysis-technique (e.g. robust regression in case one wants to relate variables to each other).

2.3.4 Data reduction

In situations where the dataset is very large, data reduction is in order to reduce run time and storage problems in performing cluster analysis. The challenge is to obtain a reduced representation of the dataset that is much smaller in volume but produces the same (or almost the same) analytical results. Various reduction strategies are in order to achieve this:

(i) Aggregation: consists of combining two or more attributes (or objects) into a single attribute (or object), thus resulting in a reduced number of attributes or objects. One should strive to find aggregations which make sense, and highlight important aspects of the problem at hand. This can also involve a change of scale (e.g. cities aggregated into regions, states, countries; daily, weekly, monthly averages), and can render more ‘stable’ data (less variability), however at the price of losing information on the more detailed scale.

(ii) Sampling: Instead of processing the complete dataset one can decide to process part of the dataset which is obtained by selecting a restricted (random) sample. In this process one has to be sure that the selected sample accurately represents the underlying cluster- or populations structure in which one is interested.

(iii) Feature selection: Feature Selection consists of identifying and removing features (or equivalently attributes, variables) which are redundant (e.g. duplicating much of the information in other features) or irrelevant (e.g. containing no information that is useful for the data mining task at hand, e.g. identifiers of objects). Apart from brute force approaches which try all possible feature subsets, more advanced techniques can be invoked as e.g. *filter* and *wrapper* approaches to find the best subset of attributes

(see the extensive literature on these topics in machine learning and data-mining, e.g. Blum and Langley, 1997, Kohavi and John, 1997; see also Xing, 2003, Guyon and Elisseeff, 2003, Guyon et al., 2006, Handl and Knowles, 2006, Liu, Yun, 2005, Saeys et al. 2007). This last class of techniques can be implemented in a forward (stepwise forward selection) or a backward (stepwise backward elimination) fashion, similar to stepwise regression. See also table 5 in Jain et al. (2000) where a number of feature selection methods are briefly discussed in the context of statistical pattern recognition.

A number of (recent) publications more specifically address feature (or variable, attribute) selection for cluster analysis:

- Friedman and Meulman (2004) proposed, in the context of hierarchical clustering methods, a method to cluster objects on subsets of attributes. It is based on the idea that subsets of variables which contribute most to each cluster structure may differ between the clusters. Software is available in R to perform this analysis (*COSA*; see http://www-stat.stanford.edu/~jhf/COSA.html). Damian et al. (2007) describe applications of this algorithm in medical systems biology.
- Raftery and Dean (2006), in the context of model-based clustering, propose a variable selection method, which consistently yields more accurate estimates of the number of groups and lower classification error rates, as well as more parsimonious clustering models and easier visualization of results. See the CRAN-package <<*clustvarsel*>>[8] for related software.
  For interesting further developments see the recent paper of Maugis et al. (2008, 2009). Methods which especially focus on situations with very many variables (high-dimensional data), are furthermore presented in McLachlan et al. 2002, Tadesse et al. (2005), Kim et al. (2006). See also Donoho and Jin (2008, 2009) for the related case of discriminant analysis (i.e. supervised classification).
- Steinley and Brusco (2008b) compared various procedures for variable selection proposed in literature, and concluded that a novel variable weighting and selection procedure proposed by Steinley and Brusco (2008a) was most effective.
- Mahoney and Drineas (2009) recently proposed so called *CUR matrix decompositions*, i.e., low-rank matrix decompositions that are explicitly expressed in terms of a small number of actual columns and/or actual rows of the original data matrix as a means for improved data-analysis, which can be usefully applied in clustering.
- Donoho and Jin (2008, 2009) address optimal feature selection in the context of classification and discriminant analysis in case that useful features are rare and weak. Their idea of using a thresholding strategy for feature Z-scores can be extended to cluster analysis applications.
- Fraiman et al. (2008) recently introduced two procedures for variable selection in cluster analysis and classification, where one focuses on detecting ‘noisy’ non-informative variables, while the other also deals with multi-colinearity and general dependence. The methods are designed to be used after a ´satisfactory´ grouping procedure has already been carried out, and moreover presuppose that the number of clusters is known and that the resulting clusters are disjoint. The main underlying idea is to study which effect the blinding of subsets of variables (by freezing their values to their marginal or conditional mean) has on the clustering results as compared to the clustering the full variable set. To enable analysis for high-dimensional data a heuristic forward-backward algorithm is proposed to

[8] Cf. http://cran.r-project.org/web/packages/clustvarsel/

consecutively search (in a non-exhaustive way) for an appropriate variable selection. The performance of Fraiman's methods in simulated and real data examples is quite encouraging, and at points it also outperformed Steinley and Brusco (2008a) method.

- Krzanowski and Hand (2009) recently proposed a simple F-test like criterion to evaluate whether the ratio of the between-group and the within-group sum of squares for each specific variable is significantly greater than what would be expected in a single homogeneous population (i.e. if no clustering would be involved). On basis of this easily computable test they expect to make an appropriate pre-selection/reduction of the variables for clustering applications with very many variables involved. This is especially the case for applications like the genetic characterization of diseases by microarray techniques, where typically very many gene expression levels p are involved as compared to subjects n (e.g. values of n are in the hundreds, while values of p are in the thousands). More specialized approaches for these high dimensional situations are more computationally demanding and more specifically bound to specific cluster analysis techniques like mixture model-based approaches (cf. McLachlan et al. 2002, Tadesse et al. (2005), Kim et al. (2006)).

In appendix D, we highlight some simple alternatives related to the latter two methods that can be straightforwardly used for performing this feature selection, and give some examples of their use.

Complementary to *variable selection* one can also consider the use of *variable weighting* to express the relative (ir)relevance of features or variables (Gnanadesikan, Kettenring and Tsao, 1995). De Soete, (1986, 1988) initially has developed optimal schemes for ultrametric and additive tree clustering (see also Milligan, 1989), and Makarenkov and Legendre (2001) have extended these[9] also for K-means partitioning methods. For k-means type clustering Huang et al., 2005 propose a procedure that automatically updates variable weights based on the importance of the variables in clustering. Small weights reduce the effects of insignificant or noisy variables. As a further improvement on Huang's procedure, Tsai and Chiu (2008) recently proposed a weight self-adjustment (FWSA) mechanism for K-means to simultaneously minimize the separations within clusters and maximize the separations between clusters. They discuss the benefits of their method on basis of synthetic and experimental results. Gnandesikan et al. (2007) recently proposed simple methods for weighting (and also for scaling) of variables.

(iv) Dimension Reduction/Feature Extraction: For reducing the dimensionality of the dataset, various methods can be applied which use (non-linear) transformations to discover useful and novel features/attributes from the original ones (cf. Jain et al. 1999, 2000, Law and Jain, 2006, Camastra, 2003, Fodor, 2002). E.g. principal component analysis (PCA) (Jolliffe, 2002) is a classical technique to reduce the dimensionality of the data set by transforming to a new set of variables which summarizes the main features of the data set. Though primarily defined as a linear feature extraction technique, suitable non-linear variants (kernel PCA) have been developed in the last decades (see Schölkopf et al. 1999). PCA is often used as a preliminary step to clustering analysis in constraining attention to a few variables. But

[9] For downloading this software see http://www.bio.umontreal.ca/casgrain/en/labo/ovw.html

its use can be problematic as illustrated by Sneath, 1980, Chang, 1983. These references show that clusters embedded in a high-dimensional data-space will not automatically be properly represented by a smaller number of orthogonal components in a lower dimensional subspace. Yeung and Russo, 2001 also demonstrate that clustering with the PC's (Principal Components) instead of the original variables does not necessarily improve cluster quality, since the first few PC's (which contain most of the variation in the data) do not necessarily capture most of the cluster structure.
In addition to PCA, alternative techniques can be envisioned for the task of dimension reduction, like factor analysis, projection pursuit, independent component analysis, multi-dimensional scaling (MDS[10]), Sammon's projection[11], IsoMap, Support Vector Machines, Self-Organizing Maps etc. (cf. De Backer et al. 1998, Jain et al. 2000, Fodor, 2000, Tenenbaum et al. (2000)). However, the same caveats as mentioned before for the PCA remain active. Moreover one should realize that feature extraction - unlike feature selection - typically results in transformed variables, consisting of (non)linear combinations of the original features, for which the original meaning has been lost. This can be an impediment in interpreting the results of the subsequent clustering in terms of the original variables.
In R the packages[12] <<*kernlab*>> and <<*MASS*>> deal with several of these computational techniques.

(v) Mapping data to a new space
In order to highlight specific dynamics in the data, techniques like using Fourier transforms or wavelet transforms can be used to map the data into a new space, where further analysis can take place (cf. § 2.5.3. in Han and Kamber, 2006). Underlying rationale is that in the novel space less dimensions are needed to characterize the dataset to a sufficient extend, thus achieving data reduction.

[10] MDS (multidimensional scaling) represents the similarity (or dissimilarity) among pairs of objects in terms of distances between points in a low-dimensional (Euclidean) space, and offers a graphical view of the dissimilarities of the objects in terms of these distances: the more dissimilar two objects are, the larger the distance between these objects in Euclidean space should be (Norg and Groenen, 1997).

[11] Sammon's nonlinear mapping is a projection method for analysing multivariate data. The method attempts to preserve the inherent structure of the data when the patterns are projected from a higher-dimensional space to a lower-dimensional space by maintaining the distances between patterns under projection. Sammon's mapping has been designed to project high-dimensional data onto one to three dimensions. See Lerner et al. (2000) for information on initialising Sammon's mapping.

[12] Cf. http://cran.r-project.org/web/packages/kernlab and http://cran.r-project.org/web/packages/MASS/

2.3.5 Data discretisation

By dividing the range of continuous attributes into intervals one can reduce the number of values. Reduction of data can also be established by replacing low level concepts by higher level concepts (e.g. replacing numeric values for the attribute ‘age’ by categories as young, middle-aged or senior). Techniques like binning, histogram analysis, clustering analysis, entropy-based discretisation and segmentation by natural partitioning can be applied for this purpose (cf. § 2.6 in Han and Kamber, 2006)

2.3.6 Cluster tendency

One difficulty of cluster algorithms is that they will group the data into clusters even when there are none. Later we will discuss the possibilities of validating the results of a clustering but here we present a number of ways by which the user can estimate *a priori* whether data contains structure.

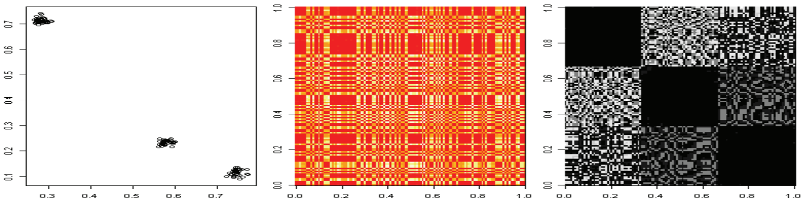


Figure 2: Artificial data set (left), image-plot (R-function) of the distance matrix of this data set (centre), image-plot of the data set after applying VAT-algorithm (right).

In the VAT-algorithm Bezdek, Hathaway and Huband (2002) represent each pair of objects by their distance. The emerging dissimilarity matrix is subsequently ordered and visualized by grey levels (0 if distance is zero and 1 for the maximum distance) (Figure 2, right). See also Bezdek, Hathaway and Huband (2007) where a technique is presented for the visual assessment of clustering tendency on basis of dissimilarity matrices.

Hu and Hathaway (2008) further developed this idea beyond the pure graphical interpretation of the result. They implemented several tendency curves that average the distances in the dissimilarity matrix. The peak-values in the tendency curves can then be used as a signal for cluster structures and for automatic detection of the number of clusters.

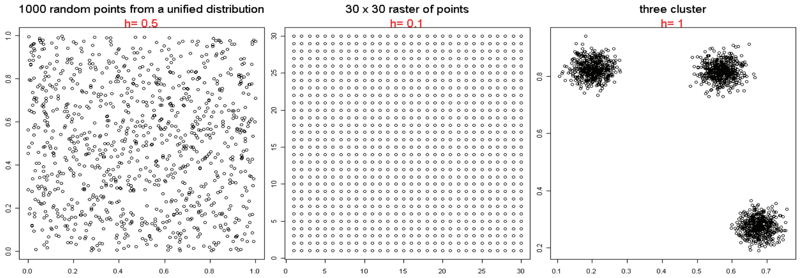


Figure 3: Artificial data set with uniformly distributed values (left) – h=0.5, Artificial raster data set (centre) – h=0.1, data with three artificial normally distributed clusters (right) – h=1.

Another possibility to check whether there are clusters in the data or not is the Hopkins-Index, which is described in Runkler (2000) or Jain&Dubes (1988). The latter reference proposes to use hypothesis tests of randomness for getting insight into the data structure. Also tests like quadrate analysis, inter-point distance and structural graphs can be employed.

Which datasets are 'clusterable'?
Ackerman and Ben-David (2009) theoretically assess several notions of *clusterability* discussed in literature and propose a new notion which captures the robustness of the resulting clustering partition to perturbations of the cluster centres. They discover that the more clusterable a data set is, the easier it is (computationally) to find a close-to-optimal clustering of that data, even showing that near-optimal clustering can be efficiently computed for well clusterable data. In practice it is however usually a computer-intensive problem (NP-hard) to determine the clusterability of a given dataset.

2.3.7 Visualizing clusters

A number of graphical techniques for visualizing and identifying clusters in one or two dimensions can be employed, such as histograms, scatter plots and kernel density estimators. For multivariate data with more than two dimensions one can e.g. use scatterplot matrices, but these only project two-dimensional marginal views and do not necessarily reflect the true nature of the structure in the p-dimensional dataspace. An alternative approach is to project the multivariate data into one or two dimensions in a way that the structure is preserved in some sense as fully as possible. A common way (although not necessarily the most appropriate) is principal component analysis. Other methods like exploratory projection pursuit, multidimensional scaling, support vector machines are also potential candidates for visualization of clusters. See e.g. chapter 2 and section 8.6 in Everitt et al. 2001, and chapter 9 in Xu and Wunsch, 2009 for more information. Also graphical techniques for exploring the structure in multivariate datasets, like co-plots or trellis graphics (see e.g. chapter 2 in Everitt and Dunn, 2001) can offer useful insights for cluster analysis. R offers various possibilities to generate such plots. In chapter 6 some of these will be discussed.

Summary

In this chapter we extensively highlighted what issues and decisions are involved in selecting and pre-processing data of interest for the problem at hand. This not only involves the treatment of missing values and outliers, but also a judicious selection of variables or features of interest (e.g. removing redundancies, avoiding overly strong dependencies) for the subsequent cluster analysis, as well as adequate data transformations to bring the data values to a more even and comparable scale. Preliminary checks on whether the data indeed contain clusters, and whether some group structure is visible will also render important information for the next steps in the actual clustering. Finally, since data-processing can influence the outcomes of the clustering, it will be important at the end to study the sensitivity of the identified clusters for feasible alternative choices in data selection and pre-treatment.

3 Selection of a distance measure in the data space

A central issue in clustering objects is knowledge on how ‘close’ these objects are to each other, or how far away. This reflects itself in the choice of the distance measure or the (dis)similarity measure on the objects.

In case that the distances between the objects are ‘directly available’, as e.g. in surveys where people are asked to judge the similarity or dissimilarity of a set of objects, the starting point of the clustering is a n-by-n *proximity matrix*, which stores the (dis)similarities between the pairs of objects (i.e. d(i,j) is the dissimilarity between objects i and j, with i, j = 1,…, n).

If distances are not directly available, information on the objects is typically available on their features/attributes. The typical starting point for a cluster analysis is then a *data-matrix* in the form of a table or n-by-p matrix that represents the n objects (rows) with their associated p attributes (columns). In discussing how this data-matrix can be transformed into a dissimilarity matrix, we assume that after the previous step highlighted in section 2 (i.e. “data pre-treatment”) the data space is in a definite form, and does not need additional normalization or weighing. This means e.g. that the application of weights to individual features to express differences in relevance has already been established. Moreover it presupposes that care has been exerted not to include non-informative features in our data, since they can trash the clustering by disturbing or masking the useful information in the other features/ variables.

3.1 The binary data case

In case that all the attributes are ***binary*** (say *0* or *1,* or no/yes), the similarity between objects is typically expressed in terms of the counts in the matches and mismatches the *p* features for two objects are compared.

		Object j		
	Outcome	1	0	T*otal*
Object i	1	a	b	a+b
	0	c	d	c+d
	Total	a+c	b+d	p

Table 1: Counts of binary outcomes for two objects

A number of similarity measures have been proposed, and a more extensive list can be found in Gower and Legendre (1986).

	Measure	Similarity-measure
S1	Matching coefficient	S(i,j)=(a+d)/(a+b+c+d)
S2	Jaccard coefficient (Jaccard, 1908)	S(i,j)=a/(a+b+c)
S3	Rogers and Tanimoto (1960)	S(i,j)=(a+d)/[(a+2(b+c)+d)]
S4	Sokal and Sneath (1963)	S(i,j)=a/[a+2(b+c)]
S5	Gower and Legendre (1986)	S(i,j)=(a+d)/[a+.5*(b+c)+d]
S6	Gower and Legendre (1986)	S(i,j)= a/[a+.5*(b+c)]

Table 2: Similarity measures for binary data, cf. table 3.3 in Everitt et al. (2001)

Notice that some of these similarity measures do not count zero-zero matches (i.e. *d*). In cases where both outcomes of binary variables are equally important (e.g. as in gender: male/female) it is logical to include zero-zero-matches when expressing the similarity between objects. However, in more asymmetric situations where the presence of a feature (e.g. an illness) is considered more important than the absence, it is advisable to exclude the zero-zero matches (i.e. the *d*) when assessing the similarity of objects, since these could dominate the similarity between objects, especially if there are many attributes absent in both objects (i.e. *d* is large, corresponding to *a,b,c*). When co-absences are considered informative, the simple matching coefficient S1 is usually employed, while Jaccard's coefficient S2 is typically used if co-absences are non-informative. S3 and S5 are examples of symmetric coefficients treating positive and negative matches in the same way, but assigning different weights to matches and non-matches. Sokal and Sneath (1963) argue that there are no fixed rules regarding the inclusion or exclusion of negative or positive matches, and that each dataset should be considered on its merits. The choice of the specific similarity measure can influence the cluster analysis, since the use of different similarity coefficients can result in widely different distance values, as is e.g. the case for S1 and S2. Gower and Legendre show that S2, S4 and S6 are monotonically related, as are S1, S3 and S5.

3.2 The categorical data case

Categorical data where the attributes have ***more than two levels*** (e.g. eye colour) could be dealt with similarly as binary data, when regarding each level of an attribute as a single binary variable. This is however not an attractive approach since many 'negative 'matches (i.e. *d*) will inevitably be involved. A far better approach is to assign a score s_{ijk} of zero or one to each attribute *k*, depending on whether the two objects *i* and *j* are the same on that attribute. These scores are then averaged over all *p* attributes to give the required similarity coefficient as:

$$s_{ij} = \frac{1}{p}\sum_{k=1}^{p} s_{ijk}$$

Notice that this similarity coefficient is a generalisation of the matching coefficient S1 for binary data.

3.3 The continuous data case

When all the attribute values are ***continuous***, the proximities between objects is expressed in terms of a distance-measure in the dataspace. Often Euclidean distance is used:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

but various other distance measures can be applied as well, as the Manhattan or city-block distance:

$$d(i,j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \cdots + | x_{ip} - x_{jp} |$$

or the general Minkowski distance (q ≥1)

$$d(i,j) = \left((x_{i1} - x_{j1})^q + (x_{i2} - x_{j2})^q + \cdots + (x_{ip} - x_{jp})^q\right)^{1/q}$$

We assume that missing values have been treated, e.g. by replacing them by the mean-value over the non-missing part, or by redefining the distance measure accordingly.

Also the correlation between the *p*-dimensional observations of the *i*th and *j*th objects can be used to quantify dissimilarities between them, as in:

$$d(i,j) = \frac{(1 - \rho_{ij})}{2}; \quad where \quad \rho_{ij} = \frac{\sum_{k=1}^{p} (x_{ik} - m_i)(x_{jk} - m_j)}{\sqrt{\sum_k (x_{ik} - m_i)^2} \sqrt{\sum_k (x_{jk} - m_j)^2}}$$

with m_i and m_j the corresponding averages over the *p* attribute-values. This measure is however considered contentious as a measure for dissimilarity since it does not account for relative differences in size between observations (e.g. x_1=(1,2,3) and x_2=(3,6,9) have correlation 1, although x_1 is three times x_2). Moreover the averages are taken over different attribute values, which is problematic if their scales are different. But in situations where attributes have been measured on the same scale, and refer to relative profile (e.g. for classifying animals or plants absolute sizes of organism or parts are often considered less important than their shapes), correlation measures can be also used to express dissimilarities. Further information can be found in section 3.3 in Everitt et al. (2001), Gower and Legendre (1986) and Calliez and Kuntz (1996).

3.4 The mixed data case

When the attribute values are ***mixed***, i.e. containing both continuous and categorical data values, a similarity measure can be constructed from weighing and averaging the similarities for the separate attribute values, as proposed by Gower (1971):

$$s_{ij} = \frac{\sum_{k=1}^{p} w_{ijk} s_{ijk}}{\sum_{k=1}^{p} w_{ijk}}$$

where s_{ijk} is the similarity between the *i*th and the *j*th object as measured by the *k*th feature, and w_{ijk} is typically one or zero depending on whether or not the comparison is considered valid. E.g. w_{ijk} can be set to zero if the outcome of the *k*th feature is missing for either or both of the objects *i* and *j*, or if the *k*th feature is binary and it is thought appropriate to exclude negative matches. For binary variables and for categorical variables with more than two categories the component similarities, s_{ijk}, take value one when the two objects have the same value and zero otherwise. For continuous variables the similarity measure is defined as:

$$s_{ij} = 1 - \frac{|x_{ik} - x_{jk}|}{R_k}$$

where R_k is the range of observations for the *k*th attribute (i.e. the city-block distance is used after scaling the *k*th variable to unit range).

3.5 The proximity between groups of objects

The proximity between the individual objects can be used as a basis to construct expressions for the proximity between *group of objects*. Various options exist for this: e.g. taking the smallest dissimilarity between any two objects, one from each group, leads to a nearest-neighbour distance and is also the basis for the hierarchical clustering technique applying ‘single linkage’.
The opposite is to define the inter-group distance as the largest distance between two objects, one from each group and renders the furthest-neighbour distance which is the basis for the ‘complete linkage’ hierarchical clustering technique. An in-between approach is taking the average dissimilarity, which leads to a form of ‘group average’ clustering when applied to hierarchical clustering methods. Cf. Everitt et al. 2001, section 3.5, where also alternative ways to express inter-group distances are proposed which are based on group summaries for continuous as well as for categorical data.

Summary

In order to express the similarity or dissimilarity between data points a suitable distance measure (metric) should be chosen. It forms the basis for performing the clustering to identify groups which are tightly knit, but distinct (preferably) from each other. Often Euclidean distance is used as a metric, but various other distance measures can be envisioned as well.

4 Selection of clustering method

The extensive (and ever-growing) literature on clustering illustrates that there is no such thing like an optimal clustering method, an observation which is further underpinned by theoretical insights from Kleinberg (2002); see also Zadeh and Ben-David (2009). From the multitude of methods we will consider a number of classes of methods, giving most attention to traditional methods based on performing the clustering hierarchically and methods that constructively partition the dataset into a number of clusters (section 4.1 and 4.2), while describing the other methods only briefly (section 4.3-4.6). We will finish this chapter with a brief discussion on which method to choose (section 4.7).

4.1 Hierarchical methods

A hierarchical clustering method groups data objects into a tree of clusters. It does so in an iterative way by constructing clusters from joining (agglomerative) or dividing (divisive) the clusters obtained in a previous iteration. *Agglomerative* methods start this iterative process from the initial situation where *each data point* is considered as a separate cluster, and form the hierarchical composition in a bottom up fashion by *merging* the clusters. *Divisive* methods start with the mega-cluster consisting *all data points*, and work in a top-down fashion by *splitting* the clusters subsequently. Merging or splitting is done on basis of the mutual distances between the clusters. A number of linkage-rules can be applied to express the distance between clusters. For example the “simple”-rule *(‘single-linkage’*) always takes the smallest of all possible distances between the data points within two different clusters; the “complete”-rule *(‘complete-linkage’*) chooses the largest of all distances, while the “average”-rule is based on the average distance (*‘average-linkage’*). A popular linkage-rule is the *“Ward’s” method* which merges clusters that produce the least within-cluster variance. All the information on the process of merging can be represented in a tree (*dendrogram*) which can be cut at a selected point (number of clusters), revealing a suitable cluster structure for the data. A more formal method for determining the number of clusters, based on detecting the ‘knee’ in an associated clustering evaluation graph, is proposed in Salvador and Chan (2004) and favourably compared with two alternative methods.

Hierarchical clustering methods have a large computational complexity *$(O(n^2))$*, where n is the number of data points or objects, which constrains their application usually to small and medium data size. In building the dendrogram, non-uniqueness and inversions can occur due to ties in data and due to the order of the dataset, cf. Morgan and Ray (1995), MacCuish et al. (2001) and Spaans and Heiser (2005).

The linkage-rule in hierarchical clustering can be tuned to the data, and thus also non-spherical clusters can be identified. One should however be aware that applying hierarchical clustering can lead to very different results on the same dataset, dependent on the linkage rule used: the single linkage strategy tends to produce unbalanced and elongated clusters, especially in large data sets, since separated clusters with ‘noise’ points between them tend to be joined together (‘chaining’);

complete linkage leads to compact clusters with equal diameters; average linkage tends to join clusters with small variances and is an intermediate between single and complete linkage; Ward's method assumes that the objects can be represented in Euclidean space and tends to find spherical clusters of similar size. It is sensitive to outliers. See e.g. table 4.1 in Everitt et al. (2001) and Kaufman and Rousseeuw for more information on the effects of linkage rules.

In their pure form hierarchical methods suffer from the fact that is not possible to adjust a merge or a split decision which was taken in a previous iteration. This rigidity is useful since it restricts computational costs in preventing a combinatorial number of different choices, but it may lead to low-quality clusters if the merge or split decisions turn out to be not well-chosen. To improve this one can try to integrate hierarchical clustering with other clustering techniques, leading to multi-phase clustering. Three such methods are discussed in more details in Han and Kamber (2006). The first, called *BIRCH*, applies tree structures to partition the objects into 'microclusters' and then performs 'macroclustering' on them using another clustering method such as iterative relocation. The second method, called *ROCK*, merges clusters based on their interconnectedness, and is a hierarchical clustering algorithm for categorical data. The third method, called Chameleon, explores dynamic modelling in hierarchical clustering.

In R hierarchical clustering can be invoked by the general function *hclust()*; various more specific hierarchical clustering techniques have also been implemented, e.g. the methods proposed in Kaufman and Rousseeuw (1990) (see the R-package <<*cluster*>>):

- *DIANA()* for divisive clustering
- *MONA()* for clustering binary data., using the monothetic divisive algorithm.
- *AGNES()* for agglomerative clustering, providing six methods for the agglomeration process:

Other R-packages with hierarchical clustering methods are <<*ctc*>> (function "*xcluster()*"); <<*amap*>> (function "*hcluster()*" and "*hclusterpar()*").

Example: Hierarchical Agglomerative Clustering

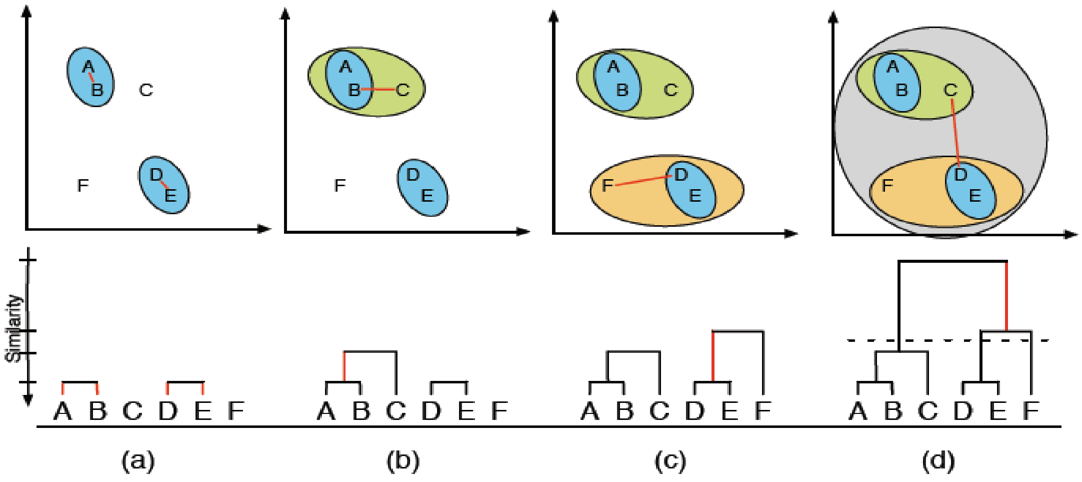


Figure 4: Example of hierarchical clustering: clusters are consecutively merged with the most nearby clusters. The length of the vertical dendogram-lines reflect the nearness.

4.2 Partitioning methods

Partitioning algorithms divide a data set into a number of clusters, typically by iteratively minimizing some criterion expressing the distances between the data points and prototypical elements of a cluster (e.g. *cluster-centroids*).
Usually the square error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - m_i|^2$$

where E is the sum of the square error for all objects in the data set; x is the point in the space representing a given object, and m_i is the mean of cluster $C_{i.}$ I.e. for each object in each cluster the distance from the object to its cluster centre is squared and the distances are summed. This criterion tries to make the resulting k clusters as compact and as separate as possible. The number of clusters k is usually predetermined, but it can also be part of a search procedure using an explicit error-function.
When using the popular *k-means* partitioning algorithm one starts with k initial cluster centroids. The data points are then assigned to the nearest centroid. Subsequently the new center is determined as the average of all points within the cluster thus obtained and again all points are re-assigned to their nearest centroid. This procedure is repeated until a convergence is reached (e.g. points no longer change position), see Figure 5.

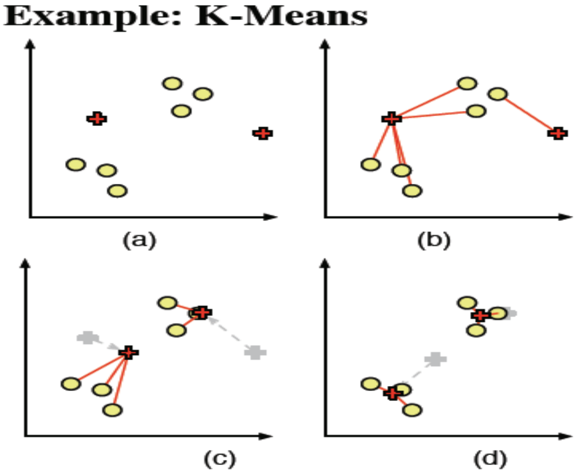


Figure 5: Example of the iterative cluster-partitioning by K-means. Starting with an initial guess of the centroids (a), consecutively the data points are grouped to the nearest centroids (b), and the new centroids are determined as the centres of these groups. In the next step (c) the points are regrouped to the nearest (new) centroid. This process is repeated until the groups don't change anymore.

k-means has a computational complexity of order *O(kn),* where n is the number of data points, and is therefore also suitable for large datasets (n large). Its outcomes are sensitive for the initialization of the iterative search process and an appropriate initialization is therefore of concern. E.g. Milligan (1980) proposes an initialisation on basis of Hierarchical clustering with Ward's method on a small random subset of the large dataset; Arthur and Vassilvitskii (2001) recently proposed a smart seeding technique for initializing k-means. See Steinley & Brusco (2007) on various strategies for initializing k-means.

Another shortcoming of *k-means* is that it does not perform well for non-spherical and non-well separated clusters, or for clusters of very different sizes. Moreover it is sensitive to noise and outlier data points since a small number of such data can drastically influence the mean value/center points.

There are quite some variants of the *k-means* method (see e.g. Steinley (2006)), which have been developed to improve the weak points. E.g. when clustering categorical data, the means of the clusters are not suitable representatives, and *k-means* has been replaced by the *k-modes method* (Chaturvedi, Green and Carroll (2001)) which uses new dissimilarity measures to deal with categorical objects and a frequency-based method to update modes of clusters. For data with mixed numeric and categorical values *k-means* and *k-modes* can be integrated.

To deal with the sensitivity to outliers Kaufman and Rousseeuw (1990) proposed *k-medoids* clustering by the *PAM*-approach (Partitioning Around Medoids; see the function *pam()* in the R-package <<*cluster>>)*). The main difference to *k-means* is the choice of *representative objects* as cluster centres instead of the arithmetic mean. In the same way as above after choosing *k* representative medoids the objects of the data set will be assigned to the nearest representative medoids. In fact the partitioning method is performed by minimizing the sum of the dissimilarities between each object and its corresponding representative point, i.e. using the absolute error criterion which is less sensitive to outliers

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \left| x - o_i \right|$$

where o_i is the representative medoid, being the most centrally located object of its cluster. In an iterative way the set of representative medoids will be calculated followed by a new assignment of the objects and so on. A nice feature in connection with *PAM* is the Silhouette plot (in R: *silhouette ()* or by plotting the *PAM*-Result). This plot illustrates how well an object lies within a cluster or merely at the edge of the cluster (Rousseeuw (1987).

The computational complexity of *PAM* is in the order $O(k(n-k)^2$, which makes computation very costly for large values of *n* and *k*. For these situations Kaufman and Rousseeuw constructed a method called *CLARA* (Clustering LARge Applications). In the first step a small portion of the dataset is chosen as a representative of the complete dataset. Using *PAM* on this small sample, medoids are determined, which are subsequently used to assign each object of the complete dataset to a specific cluster or medoid. *CLARA* draws multiple small samples from the complete dataset, applies *PAM* on each sample and returns its best clustering as the output. The computational complexity is of the order $O(ks^2+k(n-k))$, where *s* is the size of the subsample. The effectiveness of *CLARA* is dependent on the sample sizes and - in case that the best medoids of the selected subsample do not cover the best overall medoids - *CLARA* will never find the best clustering. The quality and scalability of *CLARA* can be enhanced by allowing for an extra randomization in the iterative search for new medoids, leading to the so-called *k-medoids* algorithm *CLARANS* (Clustering Large Applications based upon RANdomized Search) proposed by Ng and Han

(1994), and improved by Ester, Kriegel and Xu (1995). *CLARANS* also enables the detection of outliers and has a computational complexity of about *$O(n^2)$.* Its clustering quality is dependent on the sampling method used. See also section 7.4.2 in Han and Kamber (2006).

Another way to generalize k-means is to explicitly consider other clustering criteria for an optimal partitioning of the clusters. In chapter 5 of Everitt, 2001 some alternatives are presented to minimizing the total within-cluster sums of squares, which underlies k-means (i.e. trace W), and which are less sensitive to scale changes in the observed data and which can also tackle clusters of different shapes (than spherical) and sizes.

Also k-means can be generalized by considering it as a special case of model-based clustering, which applies a mixture of normal distributions to describe the underlying probability density of the dataset (see section 4.5).

Other extensions of k-means - as e.g. *X-means* (Pelleg and Moore, 1999, Ishioka,2005), *G-means* and *PG-means* (Hamerly and Elklan, 2003; Feng and Hamerly,2006), *PW-K-means* (Tseng,2007) - focus especially on the automatic estimation of the number of clusters, where the X-means variant implements Bayesian Information criterion to tackle the choice of dimension. See also Tseng (2007) who proposes the use of penalty terms and weighting (*PW-K-means*) to extend K-means for clustering with scattered objects and prior information. See Bies et al. (2009) for a recent comparison study of X-means, G-means and some other methods for estimating the number of clusters.

To identify *non-convex* clusters, extensions as *kernel k-means* and *spectral clustering* have been put forward, which enable identifying clusters that are non-linearly separable in input space (see e.g. Schölkopf et al.,1999, Girolami, 2002, Camastra and Verri,2005, Filipone et al. 2007, Chang et al., 2008). See also section 4.6.

Finally, the sensitivity to initial conditions in *K-means* is a well-known problem for which many initialization strategies have been proposed (see e.g. Arthur and Vassilvitskii, 2001, Steinley and Brusco, 2007). Barbakh and Fyfe (2008) propose a new family of algorithms to solve the problem of sensitivity to initial conditions in *K-means*, by applying alternative performance functions which incorporate global information.

4.3 Density-based methods

Density-based clustering methods have been developed to discover clusters with *arbitrary* shape. These methods typically regard clusters as dense regions of objects/points in the dataspace that are separated by regions of low density (representing noise). *DBSCAN* grows clusters according to a density-based connectivity analysis. *OPTICS* is an extension of *DBSCAN*, producing a cluster ordering obtained from a wide range of parameter settings. *DENCLUE* clusters objects based on a set of density distribution functions. It has a solid mathematical foundation, allowing compact mathematical description of arbitrarily shaped clusters in high dimensional datasets. It generalizes various clustering methods, including partitioning and hierarchical methods, and applies a computationally efficient calculation by applying a tree-based access structure. However the method requires careful selection of the density parameters and noise threshold that may significantly influence the quality of the clustering results. For a concise description of these methods we refer to Han and Kamber, 2006. See Tan et al., 2010 for a recent proposal for improvements of density-based clustering algorithms.

4.4 Grid-based methods

This approach uses a multi-resolution grid data structure. For this purpose it quantizes the data space into a finite number of cells, forming the grid structure.
The main advantage of the approach is its fast processing time, which depends only on the number of cells in each dimension of the quantized space, and not on the number of data objects. Approaches as *STING*, *WaveCluster* and *CLIQUE* are various examples of this approach and can be found in section 7.7 and 7.9 of Han and Kamber, 2006.

4.5 Model-based clustering methods

Model-based clustering methods hypothesize a model for each of the clusters and find the best fit of the data to the given model. The clusters are determined by constructing a density function reflecting the spatial distribution of the data points. Often also the number of clusters can be automatically determined on basis of statistical criteria taking account of noise and outlier effects (see the textbox below).
In fact the k-means method can be viewed as a special case of model-based clustering for a Gaussian mixture model with equal mixture weights and equal isotropic variances (see Celeux and Govaert, 1992). As noticed before, this directly offers a fruitful alley for generalization of k-means and finding more suitable forms of clustering non-spherical clusters and large datasets. Celeux and Govaert (1995), propose a generalization of k-means which enables the clustering of non-spherical models (Biernacki et al.,2006). The *MIXMOD*- software that they developed to analyse multivariate datasets as mixtures of Gaussian populations, for clustering and classification purposes, can be downloaded from http://www-math.univ-fcomte.fr/mixmod/index.php. Another popular package is the EMMIX-software which was developed by McLachlan et al. (2000). Related is also the R-package <<*mclust*>> developed by (McLachlan, Fraley and Raftery (2002), Fraley and Raftery (2007). See also Samé et al. (2007), Maugis et al. (2009) which discuss the application in variable selection; see also Li (2005), Yeung (2001).
Establishing such a probabilistic framework for clustering also suggests the use of several information criteria to automatically determine the number of clusters, like Akaike's first information criterion, Schwartz Bayesian information criterion, and the integrated classification-likelihood (see textbox below). See also Fraley and Raftery (1998) and Tibshirani et al. (2001) paper on the use of the gap statistic for estimating the number of clusters (the R-package <<*clusterSim*>> provides functionality to calculate this statistic).

Information criteria for k-means

To view k-means in a statistical context it is assumed that the underlying density for the points in the data space can be expressed as a mixture of K equally weighted Gaussian distribution having mean μ_k and common variances σ^2:

$$P(x_j \mid M, \sigma^2) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\frac{\left\|x_j - \mu_k\right\|^2}{\sigma^2}\right]$$

In fact the μ_k refers to the centres of the resulting clusters k=1, …, K, while the variance σ^2 refers to the within-cluster variances,

$$\sigma^2 = \frac{1}{N}\sum_{k=1}^{K}\sum_{j\in C_k}\left\|x_j - \mu_k\right\|^2$$

where N is the number of data points. The associated likelihood of the complete dataset D={x_j} is equal to, under the assumption of independence:

$$P(D \mid M, \sigma^2) = \prod_j P(x_j \mid M, \sigma^2)$$

By assigning each data point x_j to the mixture component k_j having highest probability, the *classification likelihood* of the data point x_j is equal to:

$$P_c(x_j \mid M, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\frac{\left\|x_j - \mu_{k_j}\right\|^2}{\sigma^2}\right]$$

K-means can be viewed as an attempt to maximize the joint negative classification log-likelihood of the data:

$$\ln(P_c(D \mid M, \sigma^2)) = \ln(\prod_j P_c(x_j \mid M, \sigma^2)) = \sum_j \ln P_c(x_j \mid M, \sigma^2)$$

$$= -\frac{1}{2}\left(K\cdot\ln(2\pi\sigma^2) + \sum_j \frac{\left\|x_j - \mu_{k_j}\right\|^2}{\sigma^2}\right)$$

In the light of this interpretation a number of information criteria can be proposed to estimate the optimal number of clusters (see the appendix in Goutte et al. 2001):

– Akaike's first information criterion:

$$AIC = 2\cdot\ln(P_c(D \mid M, \sigma^2) - 2\cdot(K\cdot p + 1)$$

where (Kp+1) is the number of free parameters in the underlying mixture model with K components (i.e. K times the number of parameters in the mean μ_k and the variance σ^2)

– Schwartz Bayesian information criterion:

$$BIC = 2\cdot\ln(P_c(D \mid M, \sigma^2) - (K\cdot p + 1)\cdot\ln(N)$$

– The *integrated completed likelihood* (Goutte et al., 2001):

$$ICL = 2\cdot\ln(P_c(D \mid M, \sigma^2) - (K\cdot p + 1)\cdot\ln(N) - 2\cdot\sum_{i=1}^{N}\ln(i + \frac{K+2}{2}) + 2\cdot\sum_{k=1}^{K}\sum_{j=1}^{N_k}\ln(j + \frac{3}{2})$$

with p being the number of attributes, N the number of data points where $N = \sum_k N_k$ with N_k being the number of data points in cluster C_k. The number of clusters K_{opt} rendering the highest value of the information criterion is chosen in the end as the number of clusters K.

The AIC is known to overestimate the number of clusters, especially if the clusters are non-sperical, while the BIC is known to asymptotically estimate the 'true' model structure in case that the underlying Gaussian mixture model is an adequate model. The ICL takes into account that the underlying mixture model might not be an adequate model for classifying the data points accordingly. See (Goutte et al. 2001) for further details and references.

For a good recent overview paper on finite mixture models and model-based clustering methods see Melnikov and Raita (2010). We notice that other approaches also can be listed in the category of model-based approaches, like *COBWEB* which is a conceptual learning algorithm taking concepts as a model for clusters and performing an associated probabilistic analysis. *SOM* (or self-organized feature map; see next section) is a neural-network-based algorithm that maps high-dimensional data into a 2-D or 3-D feature map, which renders useful data visualization and can be used subsequently as a basis for clustering.

4.6 Clustering methods: Miscellanea

Below we briefly discuss various alternative methods which have been developed for specific application situations.

SOM

The self-organizing map (*SOM*) due to Kohonen (1982) is a well-known neural network method for unsupervised learning and thus can be suitably applied for cluster analysis. The network classifies the data points according to internally generated allocation rules, which it learns from the data. SOM's goal is to represent all points in the original (often high-dimensional) data space by points in a low-dimensional one (usually 2-D or 3-D), such that the topology (distance and proximity relations) is preserved as much as possible. The method is particularly useful when a nonlinear mapping is inherent in the data, and it is an appropriate tool for clustering and data-visualisation of high dimensional data spaces.
See Murtagh and Hernandez-Pejaras (1995), Flexer (2000), Vesanto (1999), Vesanto and Alhoniemi (2000) and Bacao et al. (2005) for further information. Waller et al. (1998) compared SOM with two partitioning and three hierarchical methods for more than 2500 datasets and showed that SOM was similar to or better in performance than the other methods. Moya-Anegón et al. (2005) compared SOM to Multi Dimensional Scaling (*MDS*) and Ward's method for analysing co-citations in the context of scientometrics and illustrated the complementarity of the various methods. See also Yiang and Kumar (2005) for further results on comparison of SOM with k-means.

Fuzzy clustering

All the methods described so far have in common that an object is always fully assigned to one and only one cluster. In the so called fuzzy clustering the objects/points have a degree of belonging ('membership' expressed in a value between 0 and 1) to the various clusters. Points on the edge of a cluster may thus be in the cluster to a lesser degree than points in the centre of a cluster. For each point x we have a coefficient $u_k(x)$ giving the degree of which it is in the k-th cluster. Typically these coefficients are normalized such that they sum up to 1 for each x. k-means can now be generalized into *'fuzzy c-means'*, where the centroid of the cluster is a kind of 'mean' of *all* points, weighted by their degree of belonging to the specific cluster:

$$center_j = \frac{\sum_x \left(u_j(x)\right)^\gamma x}{\sum_x \left(u_j(x)\right)^\gamma}$$

with $v \geq 1$ being a coefficient which is called the fuzzifier. Typically v is taken as 2. See Hathaway and Bezdek (1988) for further details. See also Kaufman and Rousseeuw, 1990 with their fuzzy cluster analysis program FANNY, which is available as function (*fanny()*) in the R-package <<*cluster*>>[13]. Mingoti and Lima (2005) present a comparative study between SOM, fuzzy c-means, k-means and traditional hierarchical clustering, showing that especially fuzzy c-means has a very good performance and renders robust results in the presence of outliers and overlapping clusters.

Clustering high-dimensional data

The curse of dimensionality is plaguing the clustering in applications where objects that contain a large number of features or dimensions have to be classified (e.g. text-documents containing thousands of keywords as features; DNA microarray data providing information on the expression levels of thousands of genes under hundreds of conditions). Many dimensions may not be relevant, moreover the data become increasingly sparse when the number of dimensions increases, causing the distance measure between pairs of points to become meaningless, while the average density of points in the data-space is likely to be low. This requires specific clustering methodologies for high-dimensional data. *CLIQUE* and *PROCLUS* are two influential subspace clustering methods, searching for clusters in subspaces or subsets of dimensions, rather than in the entire data-space. Another methodology, so called *frequent pattern-based clustering*, extracts patterns to group objects into meaningful clusters. An example of this is *pCluster*. See section 7.9 of Han and Kamber (2006), and chapter 8 in Xu and Wunsch (2009).

Constraint-based clustering

Most clustering approaches discussed by now are implemented in an automatic, algorithmic fashion, with little user guidance or interaction involved. However in situations where there are clear application requirements (e.g. preferences and constraints), one ideally wants to use these requirements to guide the search for clusters.

This can include e.g. information on the expected number of clusters, the minimal or maximal cluster size, weights for different objects, and other desirable characteristics of the resulting clusters. For clustering tasks in high-dimensional spaces, user input on important dimensions or desired results can render crucial hints or meaningful constraints for effective clustering. Some examples how constraints and *semi-supervised clustering* tasks can be established are presented in section 7.10 of Han and Kamber (2006).

Multi-objective clustering

When clustering a dataset having different properties or when analyzing it from various user-perspectives, the reliance on one sole clustering criterion is often not appropriate. In these cases it is more of interest to consider various clustering criteria simultaneously, although they can be partially complementary and even conflicting to a certain extent. The framework of multi-objective clustering allows this perspective, by framing clustering as a multi-objective optimization problem, see e.g. Handl and Knowles (2006a). They propose *MOCK* (Multi Objective Clustering with automatic K-determination) as an multi-objective extension of *k-means*, which uses an

[13] See http://cran.r-project.org/web/packages/cluster/

evolutionary search algorithm to obtain a set of trade-off solutions between the various (often conflicting) goals as a good approximation of the Pareto front. These solutions correspond to different compromises of the considered objectives, and provide a range of alternative hypotheses to the researcher. Moreover they may lead to additional insight into the properties of the data, and thus increase confidence in the results obtained. The algorithm is shown to give robust performance for data with different properties and outperforms traditional single-objective methods. Moreover it allows for automatic determination of the number of clusters. Runtime of the method is however high, and for data where clustering criteria are more specifically known, specialized methods will generally be more efficient. In Handl and Knowles (2007) and Handl, Kell and Knowles (2007) alternative applications of multi-objective optimization are presented in the context of semi-supervised learning and feature selection.

Mining sequential data (data streams, time-series)

Sequential data consist of a sequence of sets of objects with possibly variable length and other changing characteristics like dynamic behaviour and time constraints. Recognizing patterns or groups in these dynamic datasets requires specific approaches, which we will not discuss. We refer to chapter 8 of Han and Kamber (2006) and chapter 7 in Xu and Wunsch (2009) for more information on these topics.

Spatial clustering

When spatial dimensions are involved in the data, e.g. for objects having a location or having features which differ as function of location, then it can be beneficial to explicitly account for spatial structure when looking for clusters in the data. Methods for exploratory spatial data analysis can serve as means to identify groups in the data. E.g. methods for identifying (local) spatial associations and correlations from the field of spatial statistics and GIS (see e.g. Jacquez, 2008), like Moran's *I* or Geary's *c* (cf. Bao and Henry, 1996) of Anselin's LISA (Local Indicators of Spatial Association, cf. Anselin, 1995, 2005), or Getis and Ord's statistics (Getis and Ord, 1996, Ord and Getis, 2001, Aldstadt and Getis, 2006) for identifying statistical significant hot spots can be a good basis for these analyses, leading to the identification of characteristic spatial patterns (see e.g. Premo, 2004, Nelson and Boots, 2008). For software see the R-package <<*spdep*>>[14] which supports part of these analyses. See also the information page on spatial statistical software in R[15] for further software for further software, as e.g. packages as <<*DCluster*>> and <<*clustTool*>>[16].

Discovering clusters in networks

The analysis of networks and their structure and behaviour is presently an important topic in studying complex systems in nature and society (e.g. Palla et al. 2005). Especially the property of the *'community structure'*, in which network nodes are joined together in tightly knit groups, between which there are only loose connections, is an important research topic, as exemplified by Girvan and Newman (2002), Newman (2003,2004), Newman and Leicht (2007), Mishra et al. (2007), Handcock

[14] http://cran.r-project.org/web/packages/spdep/index.html

[15] http://www.spatialanalysisonline.com/output/html/R-Projectspatialstatisticssoftwarepackages.html

[16] http://cran.r-project.org/web/packages/DCluster/index.html and http://cran.r-project.org/web/packages/clustTool/index.html

et al. (2009, 2007). See also the R-package <<*latentnet*>>[17] which has been developed for the analysis reported in the latter reference.
Remark: According to (Newman, 2003) network clustering is not to be confused with data clustering which detects groupings of data points in high-dimensional data spaces. The two problems have common features, and algorithms for the one can be adapted for the other, and vice versa, but, on balance, one typically finds that this transposition of algorithms between fields works less than the algorithms which have been directly developed.

Bootstrapping cluster analysis
By experimentally replicating the cluster analysis, using e.g. random restarts/initializations or random noise simulations, one can get clues about the stability (robustness) of the clustering results. Kerr and Churchill, 2001 elaborate on this technique in an ANOVA setting, allowing for a distinction between systematic sources of variations and noise. They illustrate the *bootstrapping* technique with a publicly available data set and draw conclusions about the reliability of clustering results in light of variation in the data; implications of replication and good design in microarray experiments are discussed. See also the R-package <<*maanova*>> [18]which builds *consensus groups* (for *k-means* methods) or *consensus trees* (for hierarchical methods) on basis of bootstrap.

Random Forest clustering:
'Random Forests' (RF) is a popular 'ensemble-based learning' technique, based on constructing many classification trees from bootstrap sampling of the data, and subsequently generating a classification on basis of the thus generated 'forest' of trees. The procedure provides a classification with an associated estimate of the error rate, and moreover generates a measure of the importance of the involved (predictor) variables, as well as a measure of the internal structure of the data (e.g. the proximity of different data points to each other). The RF-technique is user-friendly and performs very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against over fitting (Breiman, 2001).

Though initially meant for supervised learning activities like classification and regression, it can also be applied for unsupervised learning, like clustering. To this end one invokes a 'trick', calling the original data "class 1", and constructing a synthetic dataset, "class 2". The synthetic dataset "class 2" can be constructed in two ways: (1) the "class 2" data are sampled from the product of the marginal distributions of the variables (by independent bootstrap of each variable separately);
(2) the "class 2" data are sampled uniformly from the hypercube containing the data (by sampling uniformly within the range of each variable).

Subsequently one tries to classify the combined data with the RF-procedure. The idea is that real data points that are *similar to each other* will often end up in the same terminal node of a tree, as measured by the proximity matrix returned by the RF-technique. This proximity matrix can thus be taken as a similarity measure, and clustering or multi-dimensional scaling on basis of this similarity can be used to

[17] http://www.stat.washington.edu/raftery/latentnet.html
[18] http://cran.r-project.org/web/packages/maanova/index.html

divide the original data points into groups for visual exploration. See the example in Liaw and Wiener (2002) as a work-out how to perform such an analysis with the <<*randomForest*>> package in R[19].

Kernel-Based Clustering, Support Vector Clustering and Spectral clustering

All these approaches allow to identify non-spherical clusters, which is typically not provided for by direct *k-means* oriented methods. The *kernel-based* method approaches the problem by non-linearly transforming the data into a high dimensional 'feature space'. In this space it is more likely to obtain a linear separation of these clusters/patterns, applying e.g. a SVM (*Support Vector Machines*) which constructs an optimal hyper-plane on basis of a small number of support points (the "support vectors"). The difficulty of the curse of dimensionality in the mapping to a high-dimensional 'feature space' can be overcome by the 'kernel trick', i.e. applying an inner-product kernel which avoids the time-consuming process of explicitly nonlinear mapping the data-points to the transformed space. Commonly used kernels include polynomial kernels, Gaussian radial basis function kernels and sigmoid kernels (cf. Muller et al. 2001). Different kernel functions usually lead to different non-linear separating hyper-surfaces (and thus clusters) in the original data-space. The selection of an appropriate kernel is still an open problem and is currently determined empirically. In the above way kernel versions of classical clustering algorithms can be constructed. See e.g. papers on kernel k-means and support vector clustering (Ben-Hur et al. (2001), Moguerza, Munoiz, Martin-Merino (2002) and Winters-Hilt and Merat (2007).

Spectral clustering is based on regarding the data as a graph with a set of vertices and edges (with corresponding weights). The clustering is configured as a graph cut problem where an appropriate objective function has to be optimized. The problem is solved by an eigenvector algorithm involving the matrix of weights, which performs the spectral decomposition. It results in an optimal sub graph-partitioning (see e.g. Shi and Malik, 200, Ng et al. 2002, von Luxburg, 2008). Dhillon et al. (2004), Filippone et al. (2007) show that spectral clustering and kernel-based clustering are in fact closely linked; see also Kulis et al. (2009a).
To enable analysis of large datasets - for which a full spectral decomposition is computationally prohibitive – Fowkles et al. (2004) propose the use of the Nyström method for solving eigenfunction problems; see also Drineas and Mahoney (2005) for more information on the use of this approximation in kernel-based learning. Recently Belabbas and Wolfe (2009a) provide two methods, one based on sampling and sorting, to enable the use of spectral models for very large datasets.

R-software for performing spectral clustering is available in the R-package <<*kernlab*>>[20]. The high-computational costs of the above methods (polynomial, order (O(n^3)) can be prohibitive, but recently proposals for alternative faster variants have been put forward, see e.g. Yan et al. 2009, Kulis et al. 2009b, Belabbas and Wolfe (2009a, 2009b).

Bi-clustering

Bi-clustering (*co-clustering* or *two-mode clustering*) is a clustering method which attempts to simultaneously cluster both the samples and the features (i.e. rows and

[19] http://cran.r-project.org/web/packages/randomForest/

[20] http://cran.r-project.org/web/packages/kernlab/

columns of the data-matrix), with the goal of finding “bi-clusters”, subsets of features that seem to be closely related for a given subset of samples. It is for example used in gene expression analysis by clustering microarray-data (see e.g. Cheng and Church, 2000, Madeira and Oliveira, 2004, and Tanay et al., 2002). The field shows a rapid expansion of approaches and software tools, compare e.g. Wu and Kasif (2005), Kerr et al. (2007,2008), Li et al. (2009). See also the <<*BicARE*>> R-package[21] for Biclustering Analysis and Results Exploration in the BioConductor-suite

Consensus clustering
Consensus clustering, also called ‘*ensemble clustering*’ or ‘*clustering aggregation’*, involves reconciling of diverse clusterings performed on the same dataset. The various clusterings come e.g. from different sources (e.g. using different clustering algorithms; different selections of attributes) or from different runs of the same algorithm (using other parameters; different subsamples, selections of attributes). When viewed as an optimization problem (“given a number of clusterings of some set of elements, find a clustering of those elements that is as close as possible to all the given clusterings”), it is known as *“median partition”*, and has been shown to be a computationally hard problem (NP-complete), see Goder and Gilkov (2008). For further information on alternative approaches to consensus clustering we refer to literature, e.g. Strehl and Ghosh (2002), Monti et al. (2003), Gionis et al. (2005). See also the R-software package <<*clue*>>[22] which provides an extensible computational environment for creating and analysing cluster ensembles.

4.7 Which method to choose?

Against the background of the multitude of methods (different, as well related) for cluster analysis, one is confronted inevitably with the question ‘which one to choose’? In a certain sense clustering can be considered both as an art and as a science, as reflected by discussions on a recent conference on this issue (http://stanford.edu/~rezab/nips2009workshop).
The choice of the clustering algorithm is not an application-independent issue, but should always be addressed in the context of its end-use, taking also account of the character and type of data which is available. Typically it is considered a good idea to try several algorithms on the same data to study what they will disclose. This however leaves one with the task to decide what methods to apply, and how to use and interpret them. An important issue in using and interpreting the results from the cluster analysis will be the flexibility in going back-and-forth from statistical technique to subject-content. This involves combining expertise on cluster analysis with expertise on the specific subject area where the cluster analysis is applied, and typically requires a close cooperation between content-expert and cluster-analysts, if the analysis is not done by the content-expert.

Obviously it will depend on the available expertise (on clustering and on the specific subject), software, time, money and mancraft to what extent the choice of the clustering algorithm is covered. Requirements with which one should account can be diverse, as exemplified e.g. by the list of issues like ‘scalability’, ‘ability to deal with

[21] See http://www.bioconductor.org/packages/2.6/bioc/vignettes/BicARE/inst/doc/BicARE.pdf
[22] http://cran.r-project.org/web/packages/clue/index.html

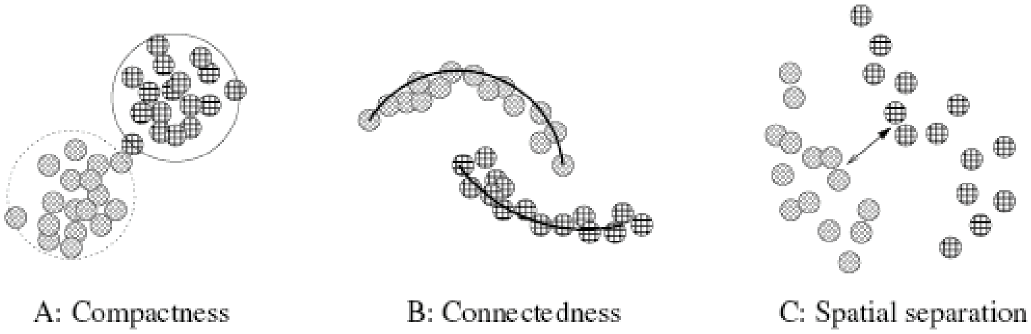
Three fundamental properties for clustering
(according to Handl and Knowles (2005))

Handl and Knowles (2005) distinguish three fundamental properties for clustering, which can give rise to conflicting objectives, and which would argue for a multi-objective approach towards clustering as exemplified e.g. in Handl and Knowles (2005).

Compactness: Generally this is implemented by keeping intra-cluster variation small. Algorithms like k-means, average link-agglomerative clustering, self-organizing maps or model-based clustering fit into this category. These methods are very effective for spherical or well-separated clusters, but may fail for more complicated cluster structures.

Connectedness: This more local concept of clustering is based on the idea that neighboring data items should share the same cluster, and methods as density based clustering and single-linkage agglomerative clustering are related to this property. Detection of arbitrarily shaped clusters is possible, but these methods can lack robustness in case clusters are not clearly separated spatially.

Spatial separation: This property on its own does not give much guidance for clustering, and can easily lead to trivial solutions. Typically it is combined with other objectives, as compactness of clusters or balance of cluster sizes.



Examples of data sets exhibiting compactness, connectedness and spatial separation, respectively. Connectedness and spatial separation are related (albeit opposite), and in principle, the cluster structure in data sets B and C can be identified by a clustering algorithm based on either connectedness or on spatial separation, but not by one based principally on compactness.

different types of attributes', 'discovery of clusters with arbitrary shape', 'ease of using the cluster analysis procedure', 'ability to deal with noisy data', 'treatment of newly inserted data', 'insensitivity to the order of the input records', 'high dimensionality' presented in chapter 7.1 Han and Kamber (2006). Moreover also issues related to cluster validity (see next chapter) will be of importance.

Handl and Knowles (see textbox) state that in clustering various objectives are involved, which can be conflicting. Therefore they argue that multi-objective approach to clustering is appropriate.

Summary

The extensive – and ever-growing - literature on clustering illustrates that there is no such thing like an optimal clustering method. We have grouped the multitude of methods into a restricted number of classes, and have especially focused on two commonly used classes, one which is based on hierarchically performing the clustering, while the other consists of constructively partitioning the dataset into a number of clusters, using the k-means method. The other classes are briefly discussed with due reference to literature for further information.

5 How to measure the validity of a cluster?

5.1 Comparing cluster solutions

The comparison of cluster solutions (e.g. partitions or trees) either with each other or with benchmark information is an important aspect of cluster validation. For example, testing whether different subsamples of the same dataset or different methods applied to the data generate similar results is considered as a relevant activity in evaluating the cluster-quality (‘*robustness issue*’). Moreover, in situations where an external classification is available, one would like to check the similarity of this classification and the clustering results as an indication of *external* clustering validity.

Below we briefly highlight a number of well-established techniques for comparing two partitions. See Everitt et al. 2001, section 8.4, for additional material on comparing two dendrograms/trees or two proximity matrices; see also Campbell, Legendre and Lapointe (2009) for further information on these issues.

E.g., when two classifications of a group of n objects are available, one can represent them as a c_1-by-c_2 matrix $N=[n_{ij}]$ where n_{ij} is the number of objects in group i of partition *1* (i=1, …, c_1) and group j of partition *2* (j=1,…,c_2). The labelling of the two partitions are arbitrary. When the partitions have the *same number of clusters* and their agreement is good, it is usually obvious from inspection how the labels correspond, and one partition can straightforwardly be relabelled to match the other. Using simple percentage agreement or the *kappa coefficient* (see Cohen, 1960) the partitions can then be compared, after relabeling.

Remark: One can think of various procedures to match the labels of two cluster partions, say 1 and 2.
A straightforward strategy consists of:

(a) first determining the *Euclidean distances* between the cluster-centres for clustering 1 and clustering 2. These distances are stored in a ‘distance matrix’ with entry $d_{i,j}$ expressing the distance between the i-th cluster-centre for clustering 1 and the j-th cluster-centre for clustering 2;
(b) next linking the labels for clustering1 and 2 by consecutively searching for the smallest entry in this matrix (smallest distance), matching the corresponding row and column and eliminating them from the matrix consecutively.

In this way a match between the cluster-classes in clustering 1 and those in clustering 2 is obtained iteratively. This is however not the only procedure to perform this matching. One can easily come up with alternatives when considering these steps:

- Concerning step (a): Matching can also be done by comparing the cluster-class *counts* in the cross table-matrix N. The idea behind this matching is to find a match which renders the *largest number of counts* (data points) in the corresponding matched cluster-classes. Notice that the match proposed sub (a) above, is based on the underlying (average) features of the data points, and aims to establish a match on basis of these averages.
- Concerning the search step (b): Instead of performing the search *heuristically* like sketched above one can envisage to perform this search *exhaustively (i.e. exact)* by considering *all cluster-combinations* involved, and finding the one which renders the *sum*

of the distances minimal[23]. Although the number of all cluster-combinations involved is equal to *k!* (*k* is the number of clusters in clustering 1), this task of finding the exact optimum cluster combinations can be performed far more efficiently (in O(k^3) steps) by using the *'Hungarian algorithm'* proposed by Kuhn (1955) and Munkres (1957). This algorithm is available in the R-package "<<*clue*>>[24]*"*, i.e. use the LSAP function for optimal cluster matching/assignment

A simple example illustrates that the outcomes of both search methods (in step (b) can be different. E.g. let the cross-table for two cluster partions (5 cluster-classes) be:

17	24	1	8	15
23	5	7	14	16
25	6	13	20	22
10	12	19	21	3
11	18	25	2	9

The heuristic search method and the optimal search method match the rows 1,2,3,4, and 5 with the columns 2, 5, 1, 4, 3 (*heuristic*) and 2, 1, 5, 4, 3 (*exact*) respectively, giving a total number count of 111 and 115 respectively, which shows the (slightly) suboptimal performance of the heuristic method.

Remark: Cohen's Kappa-statistic which corrects for chance effects in comparing two cluster partitions is given by (N* stands for the relabelled cross-table):

$$P_{agree,real} = Trace(N^*)/Sum(N^*)$$

$$P_{agree,chance} = \frac{1}{[Sum(N^*)]^2}\sum_i\left[\left(\sum_k N^*_{ik}\right)\cdot\left(\sum_j N^*_{ji}\right)\right]$$

$$I_{Kappa} = \frac{P_{agree,real} - P_{agree,chance}}{1 - P_{agree,chance}}$$

where $P_{agree,real}$ refers to the relative observed agreement between clustering 1 and 2, and $P_{agree,chance}$ refers to the hypothetical probability of the agreement by chance, in case random classes would have been assigned to the objects for both clustering 1 and 2. If the clusterings are the same I_{Kappa} is 1, if there is no agreement, other than the one happening by chance, I_{Kappa} <=1.

When the *number of clusters differs* between the two partitions/clusterings, one can take another alley towards comparing the partition rather than by analysing the cross-tabulation of frequencies. Starting point is to investigate the co-occurrence of the groupings of *every pair* of *n* objects in the partitions. This can be presented in a 2 x 2 *contingency* table:

[23] For the case of matching on basis of cluster-counts, one would strive to find a match which renders a *maximal sum of the number of counts*.

[24] http://cran.r-project.org/web/packages/clue/index.html

		Partition 2		
		Pair in same group	Pair in different groups	T*otal*
Partition 1	Pair in same group	a	b	a+b
	Pair in different groups	c	d	c+d
	Total	a+c	b+d	$\binom{n}{2}$

This contingency table can be directly derived from the cross-table *N* with cluster-class counts, using the relationships presented in table 1 and 2 of Hubert and Arabie, 1985.
The *Rand* and *Jaccard* index for expressing the correspondence of these partitions are defined by *(a+d)/(a+b+c+d)* and *a/(a+b+c)* respectively. Correcting for the effects of chance in grouping points in clusters, adjustments of the Rand index have been put forward in literature of which the *adjusted Rand index* of Hubert and Arabie (1985) is especially judged a suitable one (see also Steinley, 2004). It is defined as:

$$adjRand = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c) + (c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c) + (c+d)(b+d)]}$$

where $\binom{n}{2}$ denotes the total number of object-pairs (i.e. (a+b+c+d)).

Meila (2007) recently proposed a novel criterion for comparing partitions, the "*Variation of Information*"-criterion, which accounts for the amount of information-loss and gain when changing from clustering 1 to clustering 2. It is calculated on basis of information theoretic measures which can be directly evaluated in terms of the entries in the cross-table-matrix *N* with the cluster-class counts. See Meila (2007) for details. Vinh et al. (2009) recently argue that also for information theoretic measures a correction for chance is needed, similar to the adjustment of the Rand index.

The above mentioned indices that can be calculated on basis of the cross-table *N* of the cluster-class counts appear to be *insensitive to permutations* of the columns and rows of the cross-table. This implies that they do not depend on the cluster-label-matching strategy involved in linking clustering 1 to 2.

The presented indices have been implemented in the CRAN-package <<*mcclust*>>[25] where the adjusted rand index is evoked by the function *arandi()* and Meila's criterion by the function *vi.dist()*.

[25] http://cran.r-project.org/web/packages/mcclust/

Remark: The above indices can be used to *measure the influence of individual data points on a cluster analysis*: by comparing the partitioning which results from deleting specific data points from the dataset, with the partitioning of the complete reference dataset one can detect highly influential data points that directly impact the resulting partition. Cheng and Milligan (1995, 1996a,b) e.g. advocate the use of the adjusted Rand index for this purpose. See also section 8.5.3 in Everitt et al. 2001.

An axiomatic approach to measure cluster quality

Ackerman and Ben-David (2008) have recently initiated a systematic study of measures for the quality of a given data clustering. These measures, given a data set and its partition into clusters, return a non-negative real number representing how ‘strong’ or ‘conclusive’ the clustering is. They propose to use the notion of *‘cluster quality measure’* as a basis for developing a formal theory of clustering, which unlike Kleinberg’s axiomatic approach (Kleinberg, 2002) does not lead to contradictions.

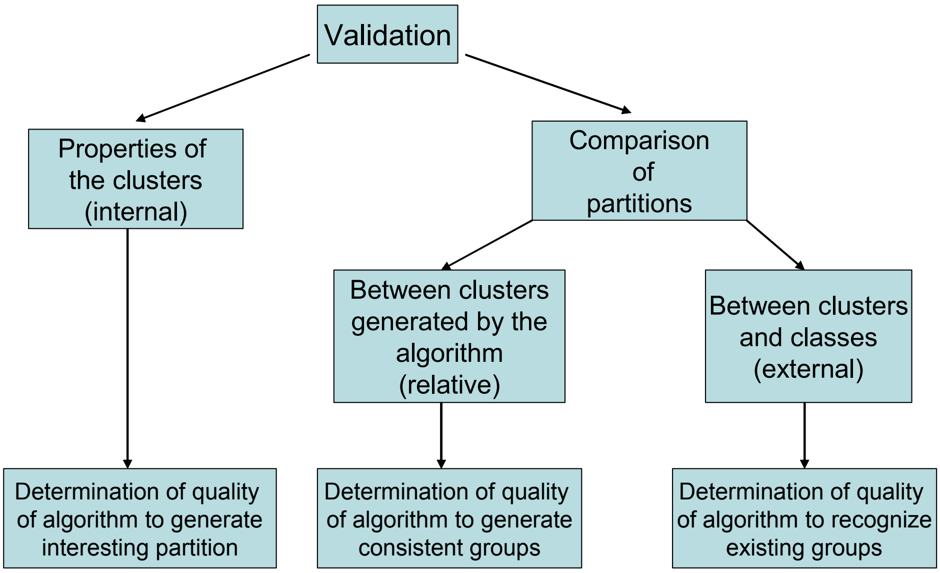
Ackerman and Ben-David have proposed quality measures for wide families of common clustering approaches, like center-based clustering (e.g. k-means, k-median), loss-based clustering (e.g. k-means) and linkage-based clustering (e.g. hierarchical clustering), and analyze their computational complexity. In addition, they show that using these quality measures, the clustering quality of a clustering can be computed in low polynomial time.

5.2 Validation measures

Validation measures are intended to measure how well the clustering captures the underlying structure in the data. An excellent account of different types of validation measures and their potential biases is given in Handl et al. (2005). This reference underlines that there does not exist a golden standard in clustering methods nor in validation measures. It will often not be sufficient to use a single clustering algorithm and/or a single validation measure when the real underlying structure of the data is unknown. Rather one should apply a number of different clustering algorithms and validation measures that optimize different aspects of a partitioning for an appropriate range of cluster s*i*zes. Also Brun et al. (2007) address similar points, and advise to be cautious with automatically applying and interpreting results from calculated validity indices.

Typically three groups of validation measures are distinguished (see Figure 6): the first type is based on calculating properties of the resulting clusters, such as compactness, separation, roundness, and is called internal validation, since it does not require additional information on the data.

The second approach is called relative validation and is based on comparisons of partitions generated by the same algorithm with different parameters (e.g. initializations), or different subsets of the data. This approach in fact measures robustness of the clustering results and - similar to internal validation - also doesn’t require additional information.



Validation
Properties of the clusters (internal)
Comparison of partitions
Between clusters generated by the algorithm (relative)
Between clusters and classes (external)
Determination of quality of algorithm to generate interesting partition
Determination of quality of algorithm to generate consistent groups
Determination of quality of algorithm to recognize existing groups

Figure 6: Different approaches for cluster validation

The third approach, called ***external validation*** is based on *comparison* of the clustering partition of the data with a *known* class partition of the data, thus presupposing that the class labels are known and uncontested. It is clear that this kind of validation will only be possible for a limited number of situations, e.g. for benchmark data, or for situations where cluster labels are known beforehand. It will evidently depend on the application field whether (and which) explicit validation criteria are feasible and useful: e.g. Datta and Datta (2006) propose two specific evaluation indices in the context of gene expression data-analysis with a content related meaning, namely the biological *homogeneity* index and the biological *stability* index.

In appendix E a large number of ***internal validation indices*** are listed that use the inter-cluster and the intra-cluster distances to identify the best partition. These indices use the inter-cluster and the intra-cluster distances to identify the best partition. They are appropriate when clusters are compact and well-separated, but fail when sub-clusters exist or when the clusters are arbitrarily shaped (and thus have no representative centre points to assess the inter-cluster variance). Therefore frequently alternative approaches are put forward in literature, which are compared to the established ones on basis of synthetic and/or real data. These comparative studies are necessarily always limited to a certain extent: their scope is given by the datasets which are analysed, and one can often find other data on which the one method performs better than another candidate. Jonnalagadda and Srinivasan (2009) propose an approach that overcomes this limitation by not using inter-cluster distances, but instead focusing on information which is lost or gained when a cluster intersects with another. The proposed NIFTI-index (Net InFormation Transfer Index) was compared with other ones - Dunn's, Silhouette, Davies-Bouldin and the Gap-statistic – and it was shown - on synthetic datasets as well as on real-life data - that NIFTI outperforms these methods in determining the appropriate number of clusters. However, the proposed method has as limitation that it models clusters as hyper-spheres, which make it less appropriate for clusters that do not have a spherical shape. Also Saitta et al. (2008) propose a new bounded index for cluster validity, the score function (SF). It is found to be always as good or better than four common validity indices – Dunn's, Silhouette, Davies-Bouldin and the Maulik Bandyopadhyay-statistic – in the case of hyper-spherical clusters. It works well on multidimensional data sets and accommodates unique and sub-cluster cases.

Relative validation indices are based on measuring the consistency of algorithms, comparing the clusters obtained by the same algorithm under different conditions, or by different clustering algorithms, and two typical approaches are discussed subsequently:

- The use of a *Figure of Merit* (*FOM*, see Yeung, Haynor and Russo, 2001) assesses the 'predictive power' of a clustering technique and strikes a balance between the external and internal criteria: *FOM* requires no prior knowledge nor relies entirely on information from the clustering process. It can e.g. be obtained by leaving out a variable, *j*, clustering the data (into *k* clusters), then calculate the *RMSE* (Root Mean Squared Error) of *j* relative to the cluster means:

$$RMSE(j,k) = \sqrt{\frac{1}{N}\sum_{r=1}^{k}\sum_{x_i \in C_r}(x_{ij} - \mu_{C_r}(j))^2}$$

  with x_{ij} being the measurement of the *j*-th variable for the *i*-th observational unit; *N* the number of observational units, C_r the set of observational units in the *r*-th cluster; $\mu_{C_r}(j)$ the mean of variable *j* over the observational units in the r-th cluster. Summing these RMSE over all variables *j* renders an *aggregate FOM (AFOM)*:

$$AFOM(k) = \sum_{j=1}^{p} RMSE(j,k)$$

  Calculating the *AFOM* for each *k* and adjusting for cluster size, and dividing by the number of variables 'left out' renders the *adjusted AFOM*:

$$AFOM_{adj}(k) = \frac{1}{p\sqrt{\frac{N-k}{N}}} \cdot AFOM(k)$$

  Low values of the clustering algorithm's *AFOM* indicate a high predictive power. By comparing the *AFOM* values at each *k* for different clustering algorithms their performance can be compared. However, Yeung et al. (2001) comment that this should only be done if the similarity metrics of the compared clustering algorithms are identical. Olex et al. (2007) show limitations of the *FOM* when the underlying similarity measure is non-Euclidean. For similarity measures based on the Pearson correlation coefficients they propose a more suitable alternative *FOM*.

- The use of a *stability measure* expresses how the cluster-membership assignment is affected by small changes/alterations in the dataset (e.g. sampling different data(sub)sets; adding noise to data) or by applying different parameter-settings for the cluster algorithm. It provides information on the *stability/robustness* of the prevailing clustering partition for these alternative choices. The stability measure is typically based on the use of an explicit criterion for cluster comparison, like the adjusted Rand index, or Meila's variation of information criterion, cf. Meila (2007). The stability-based approach can also be used to determine the appropriate number of clusters *k*, by studying for which *k* the resulting cluster partition is relatively stable/robust towards (re)sampling of the data or noise in the data. This approach is presently very popular and was initially advocated by Dudoit and Fridlyand (2002), Tibshirani et al. (2002), Ben-Hur et al. (2002), Bel Mufti and Bertrand (2007). Notice that these resampling methods in fact assume that the employed subset-samples are representative enough to reflect the inherent structure in the *whole* dataset. In situations where some clusters are of small size, this may be a problematic assumption. See also Lange et al. (2005), Hennig (2006), the <<*fpc*>> package[26] and Volkovich et al. (2008) for related approaches. Kuncheva and Vetrov (2006) specifically analyse the stability of the *k*-means cluster results with respect to random initialization. See the next textbox for s critical remarks on the appropriateness of the stability approach for the determination of the number of clusters

In the cluster analysis that we have set up for identifying patterns of vulnerability for global change we have implemented the above mentioned *stability procedure* in the following way in order to determine an adequate number of clusters *k* on basis of repetitively performing clustering for *k=2* until a maximum value K_{max}:

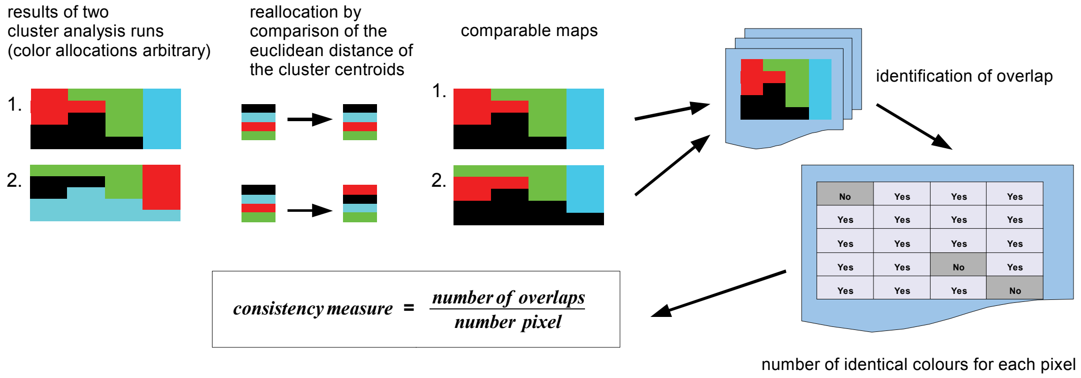
1. Initialize *k:=2*;
2. IF [$k \leq K_{max}$] THEN
   { Repeatedly (e.g. *n=150*) perform two clusterings by *k-means,* initializing each clustering with a random start-setting and compare these clusterings on basis of a criterion which gives a value between 0 and 1 to express their similarity (values around 1 hint at high similarity of the pair of clusterings). Next take the average of this criterion value $\overline{S}(k)$ over all these *n* repetitions as a measure for the stability of this resampling procedure for the specific *k*.}

   ELSE Go to step 4

[26] http://cran.r-project.org/web/packages/fpc/index.html

3. *k:=k+1*; Go to step 2;
4. Plot the average values $\overline{S}(k)$ as a function of *k* for *k=2, …, K_{max}*. This is a so-called *consistency graph*, which displays the average stability/robustness of the outcome of the clustering analysis for the resampling.

Figure 7 gives a graphical overview of the procedure (from Dietz et al., 2011). Since we used the counting of overlap method we had to reallocate the labelling of the cluster via the straightforward method of the Euclidean distance (See 5.1) to achieve comparable maps.



results of two cluster analysis runs (color allocations arbitrary)
1.
2.
reallocation by comparison of the euclidean distance of the cluster centroids
comparable maps
1.
2.
identification of overlap
number of identical colours for each pixel

No	Yes	Yes	Yes
Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes
Yes	Yes	No	Yes
Yes	Yes	Yes	No

$$\textit{consistency measure} = \frac{\textit{number of overlaps}}{\textit{number pixel}}$$

Figure 7: Operational sequence for calculating the consistency measure exemplary for k=4.

The value of *k* for which this *consistency measure* is optimal indicates a suitable choice for the number of clusters. Figure 8, shows an example from Kok et al. (2011). Besides the global optimum at k=3 there is an interesting relative maximum for eight clusters, suggesting that this number of clusters reflects also the structure of the data in case one is looking for a more differentiated partition.

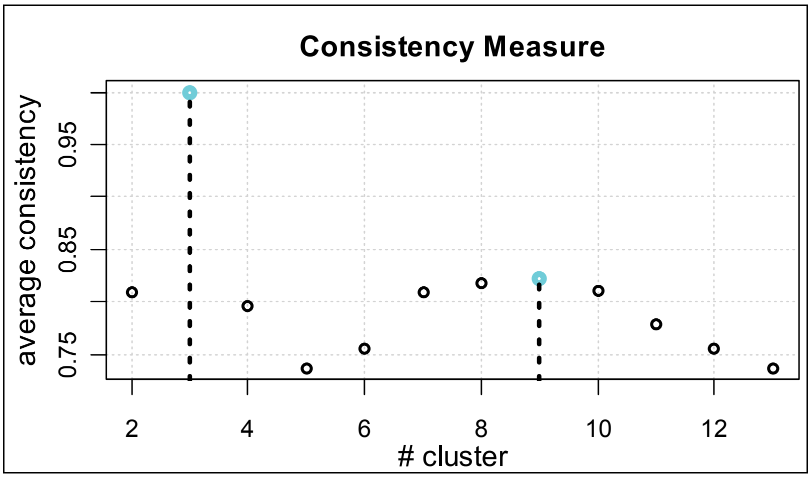


Figure 8: Consistency graph for determining the number of clusters. The local optimum at k=8 indicates that possibly an interesting suitable clustering can result if choosing e.g. 8 clusters. The number of repetitions n has been 150 in this case.

Although the above procedure is formulated primarily for the *k-means* method, it can also be applied to other clustering methods as well.
Moreover, our R-code offers various options the choice of the criterion to express the similarity between the clusterings: next to using the adjusted Rand index or Meila's variation of information criterion, it is possible to explicitly calculate the fraction of data points which have been clustered similarly when repeating the clustering with a random restart. In this case the average value $\overline{S}(k)$ can be viewed as the *average* fraction of data points which are clustered similarly when randomly restarting the clustering for this specific *k*. Typically the criterion choice does not lead to different choices in the 'optimal' number of clusters.

Criticism on the stability-based approach for choosing the number of clusters

Ben-David and von Luxburg (2006) have recently criticized the popular stability-based methods on basis of a theoretical analysis of stability issues in cluster-analysis methods that determine the clusters by globally minimizing an objective function. They discovered that for large datasets the common belief (and practice) that stability reflects the validity or meaningfulness of the chosen number of clusters is *not true*. For an elegant and useful exposition of the implications of these and other related findings see the recent publication von Luxburg (2009). Albeit the initial critical theoretical findings on the stability-based approach von Luxburg at the end draws a "carefully optimistic picture about model selection base on clustering stability for the *k-means* algorithm. Stability can discriminate between different values of *k*, and the values of *k* which lead to stable results have desirable properties. *If* the data set contains a *few well-separated* clusters which can be represented by *center-based* clustering then stability has the potential to discover the correct number of clusters." (von Luxburg, 2009; italics are added by us). In case of *very elongated clusters* or clusters with *complicated shapes* the *k*-means algorithm *cannot* find a good representation of the dataset, regardless of the number *k* used, and in these situations stability based model selection *breaks down*. Von Luxburg moreover states that these results only hold true for situations where the number of clusters is *relatively small* (in the order of 10, rather than in the order of 100). For other clustering algorithms that work very different from *k-means* it remains an *open question* whether the stability-based model selection is a suitable approach.

5.3 Software for cluster validation

The R-package <<*clValid*>> provides software for cluster validity (see Brock et al., 2008), where the generic function *cl_validity()* can be used to evaluate cluster validity indices for partitions and hierarchies obtained by clustering. See also cluster.stats in package <<*fpc*>> for a variety of cluster validation statistics; fclustIndex in package <<*e1071*>> for several fuzzy cluster indexes; clustIndex in package <<*cclust*>>*;* silhouette in package <<*cluster*>>. The R-package <<*clusterSim*>> provides various measures to express the performance of a clustering on a dataset, including the Tibshirani et al. (2001) gap statistic.

Alternative tools for validity assessment are proposed by Bolshakova et al. (2003, 2005a,b) and contain also visualization method for evaluating the clustering results.

Summary

Various ways to evaluate clustering performance and compare different clusterings have been presented. A general (stability-based) approach is put forward which assesses the robustness of clustering results for repeated analysis of the dataset under different settings (e,g, initialisations) of the cluster algorithm. It can be used for estimating the number of clusters.

6 Graphical representation of the results

Data visualisation can greatly support the interpretation of the cluster analysis. Various ways to visualize the results of the cluster analysis are possible (see also section 2.3.7). In the last chapter of this guideline we do not intend to give a comprehensive overview of all possibilities but to show some examples which occurred to be useful to us.

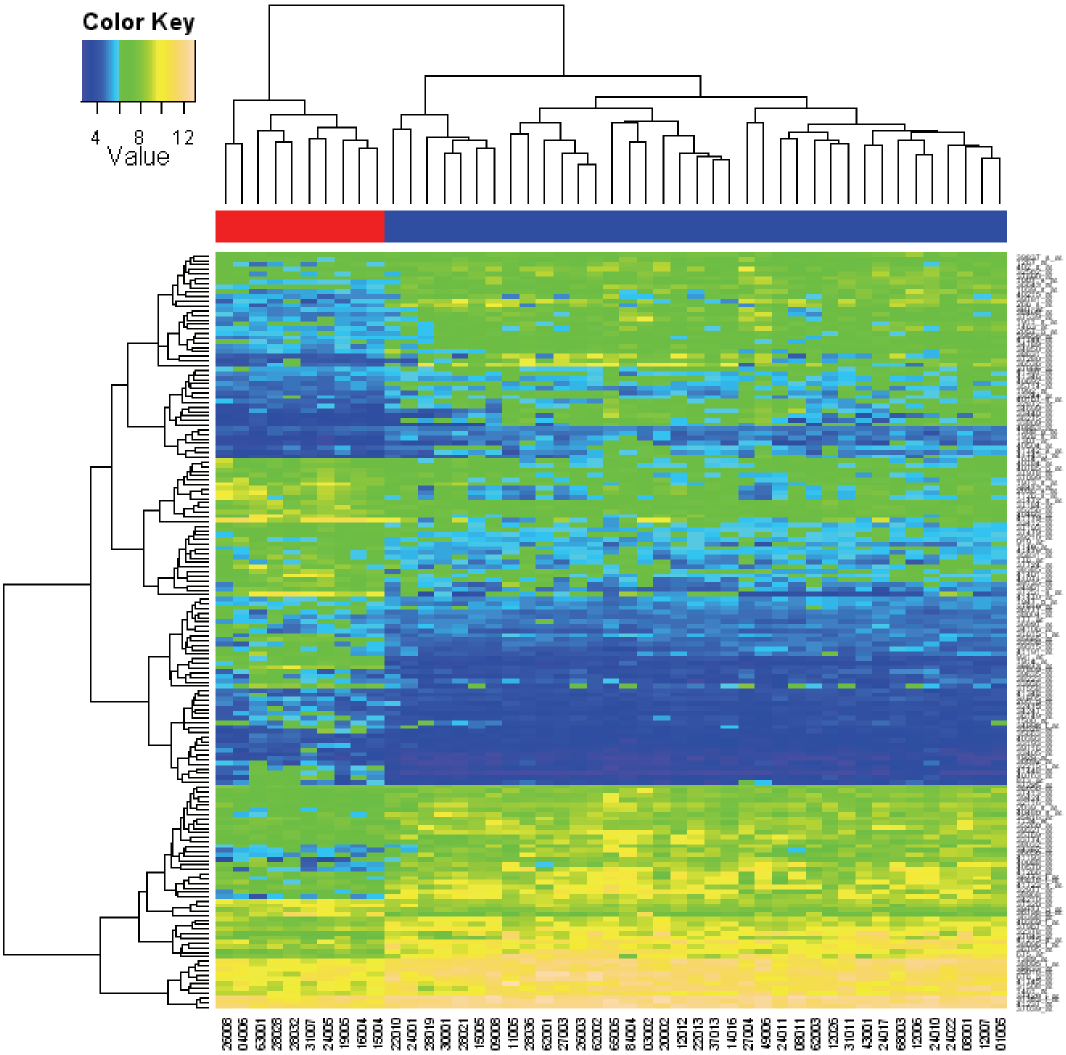


Figure 9: Heatmap of the dataset shown in Gentleman et al. (2004). See http://www2.warwick.ac.uk/fac/sci/moac/students/peter_cock/r/heatmap/ for further explanation.

6.1 *Hierarchical cluster analysis*

Hierarchical cluster analyses are typically illustrated by dendograms, showing clearly how the groupings are established. This information can further be enhanced by using *heat-maps* which provide a sorting/structuring of the data-matrix, permuting the columns and rows of this matrix to conform with the hierarchical clustering of variables and objects (see Figure 9).

The '*clustergram graph'* proposed by Schonlau (2002,2004) as alternative to dendrogram-graphs (e.g. by using the R-function dendrogram()) is in fact of similar nature as the branching diagram. It examines how objects are assigned to clusters as the number of clusters increases. Clustergrams are useful for non-hierarchical clustering algorithms such as *k*-means as well as hierarchical cluster algorithms when the number of objects is large enough to make dendrograms impractical.

Agrafiotis et al. (2007) propose *radial clustergrams* to visualize the aggregate properties of hierarchical clusters, which are specially apt for visualizing large trees which can not be displayed appropriately in straightforward dendrograms. One can also consider the use of the Dendroscope software from the University of Tübingen for this purpose (Huson et al. 2007, see Figure 10).

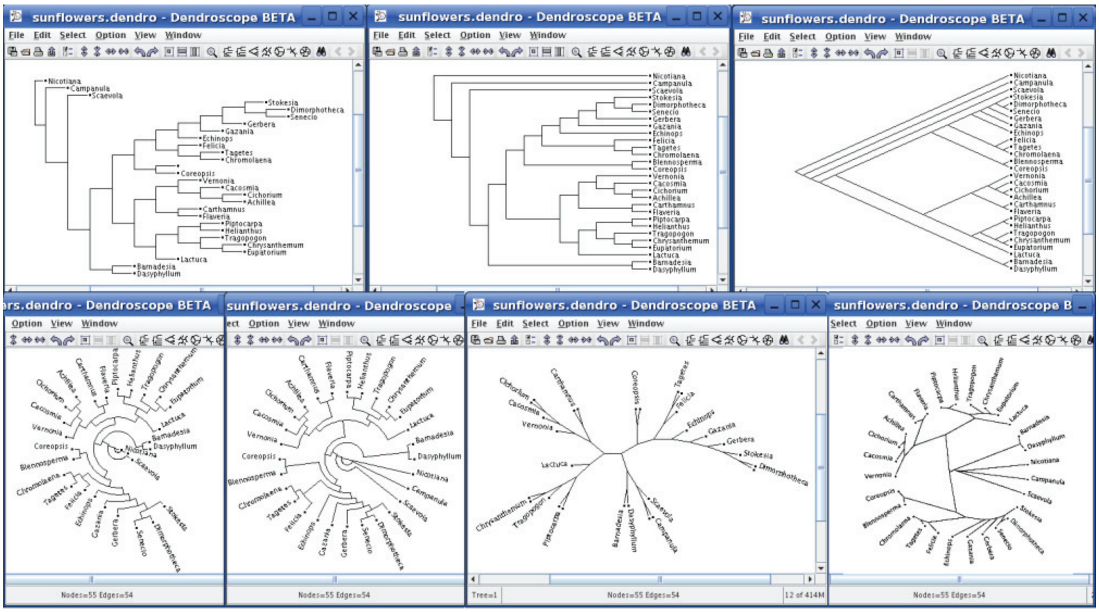


Figure 10: Seven alternative views for visualizing the same tree, implemented in the Dendroscope software (Huson et al. 2007): Rectangular Phylogram, Rectangular Cladogram, Slanted Cladogram, Circular Phylogram, Circular Cladogram, Radial Phylogram and Radial Cladogram.

6.2 *Partitioning cluster analysis*

Partitioning cluster analyses are often visualized by projecting the data in two-dimensional space, using e.g. *multidimensional* scaling (MDS) or *self-organized maps (SOM)* (see Figure 11, using Clusplot as in Pison et al. 1999; see also Vesanto, 1999, Ewing and Sherry, 2001).

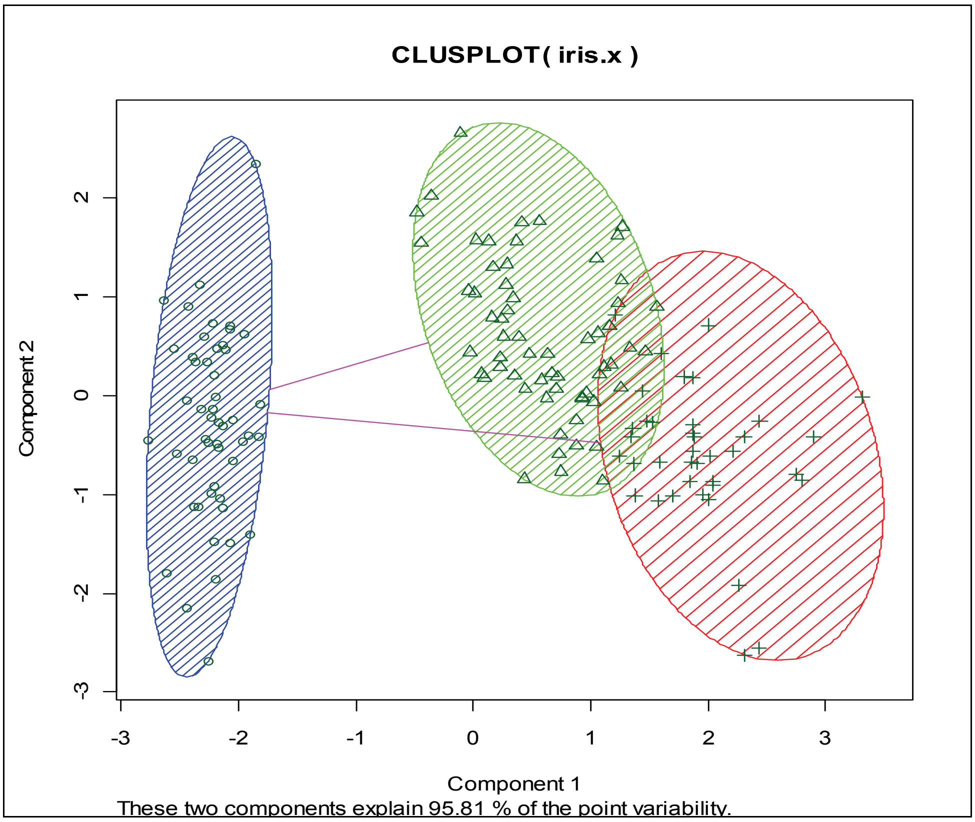


Figure 11: Two dimensional projection of the clusterpoints for the Iris dataset.

6.3 Cluster membership

Cluster membership is usually indicated by different colours and glyphs. The characteristics of the various clusters can e.g. be displayed by showing boxplots per variable/feature for the various clusters (see Figure 12), or by showing a graph of the cluster centres (see the spectral plot Fig. 13).

In the boxplot the cluster centre is indicated by the circle, while the spread around this centre is indicated by the box-boundaries denoting the lower and upper quartiles (25th and 75th percentile) of the data; thus the box-length indicates the interquartile distance, IQR. The band near the middle of the box denotes the median. Typically, boxplots are extended by whiskers denoting the minimum or maximum data values within 1.5 IQR of the lower and upper quartile. But, since we are specifically interested in high/low end percentiles, and in highlighting potential asymmetry of the distribution, we have chosen to work with alternative whiskers, and indicate them by the ends of the dotted lines which show the 5th and 95th-percentile. So between these two points 90% of the objects within a cluster are located. Notice that the boxplots for the clusters in fact only display one-dimensional information, as projected on the individual axes associated to the various variables/indicators. Information on the specific spatial structure of the cluster of points in the multi-dimensional data space (spanned by all variables/indicators considered) does not clearly show up in the boxplot.

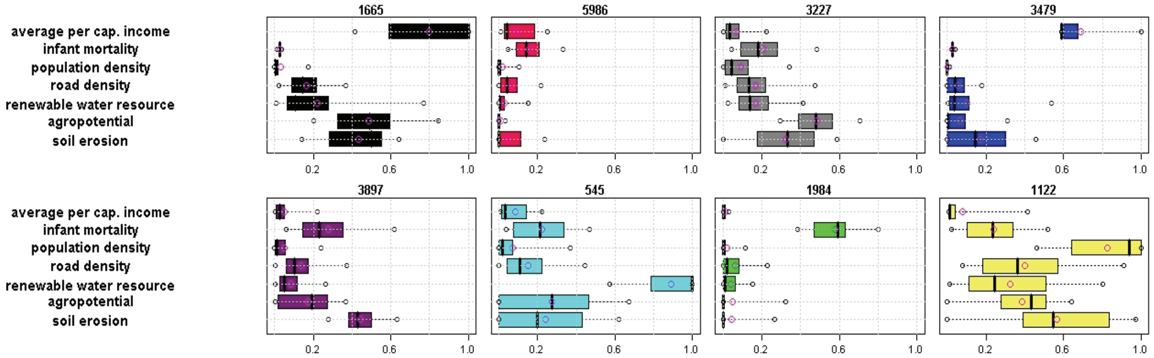


Figure 12: Boxplots, showing the variation in indicator values per cluster (colours indicate clusters; all indicator values are between 0 and 1); see Kok et al. (2010).
Note: the boxes present the 25-75 percentile range of the indicator values; the circles at the end of the dotted lines indicate the 5- and 95-percentile, while the red circle indicates the arithmetic mean; the band near the middle of the box indicates the median value. The number of points in the respective clusters is indicated in the top of the sub frames.

Graphs of the normalized cluster centres give information on how the average characteristics of the clusters differ (see Figure 13). They are helpful in suggesting the (dis)similar properties and characteristics of the various clusters.

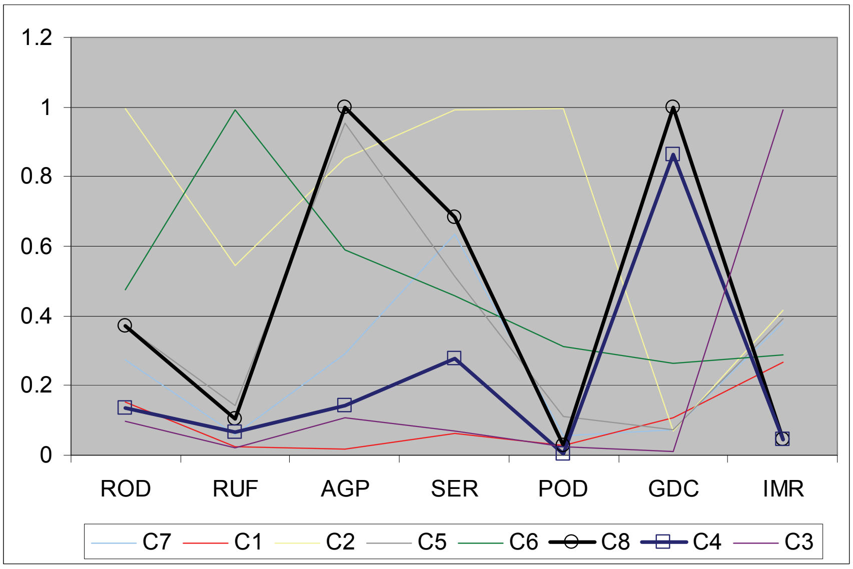


Figure 13: Cluster centres (= typical indicator values) for the 8 clusters C1 - C8; see Kok et al. (2010).

In case that the data have a *spatial* dimension, showing maps can give a clue on how the clusters are geographically distributed, serving to identify and connect features with similar characteristics at different geographical locations (see Figure 14).

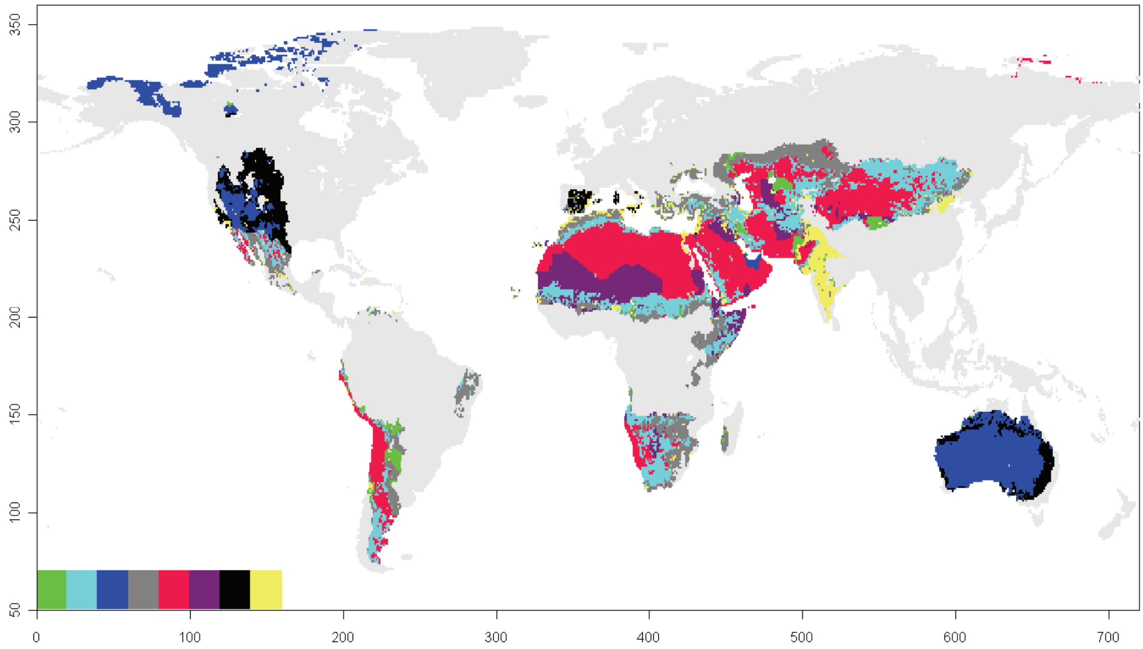


Figure 14: Distribution of clusters within the drylands (see Kok et al., 2010). Light grey: non-arid areas. Each of the 8 clusters denotes a typical constellation of the 7 indicators road density, renewable water resource, agro-potential, soil erosion, population density, GDP/cap and infant mortality rate, which are also displayed in the boxplots of Figure 12.

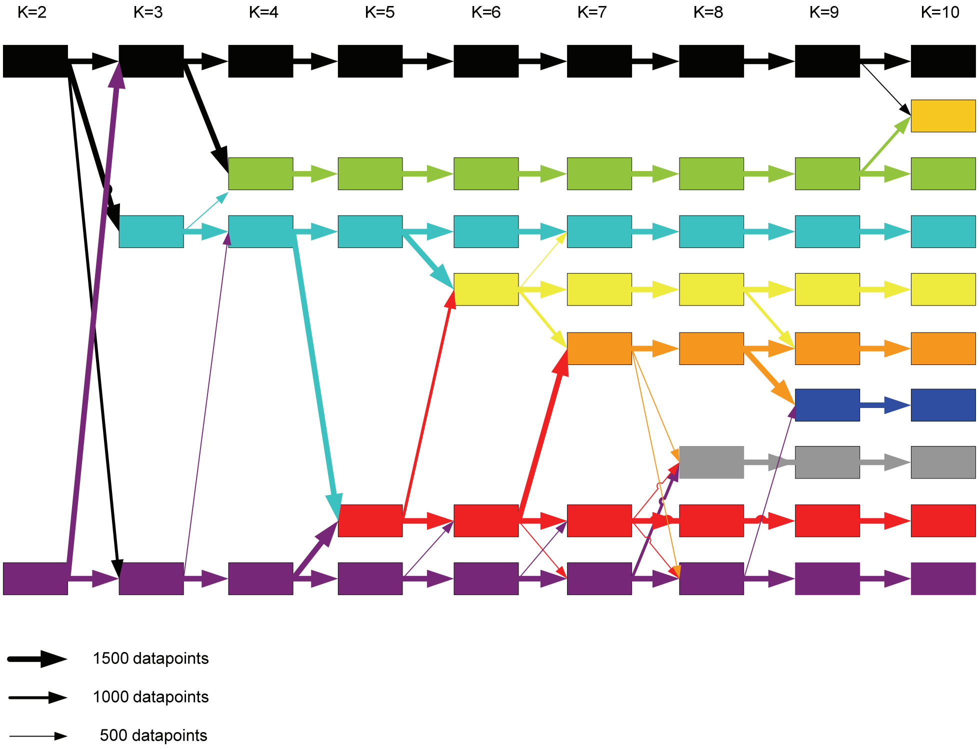


Figure 15: Branching diagram, showing cluster subdivision when increasing cluster-numbers in k-means cluster analysis of the dataset (N=45000) consisting of the indicators for the forest overexploitation archetype

6.4 Branching diagrams

When performing the cluster analysis repeatedly for a consecutive number of clusters it is insightful to construct a ‘branching diagram’ (see figure 15) which displays how the clustering structure changes when using another number of clusters. This diagram grossly indicates which clusters are split or merged, and thus renders useful information on the potential relatedness of the clusters.

Besides the above presented methods Leisch (2008, 2009) recently provide an overview of various visualization possibilities for centroid based clustering methods (neighbourhood graphs, convex cluster hulls, bar charts of cluster medoids etc.). The CRAN-package <<*flexclust*>> contains implementations of these visualization methods. See also the interactive visualization toolbox for cluster analysis in the context of gene expression data <<*gcExplorer*>> developed by Scharl and Leisch, 2009. Additional information can be found in literature on visualization methods for bioinformatics applications, like analysing gene expression microarray clusters, see e.g. Hibbs et al., (2005), Saraiya et al. (2005).

Summary

A number of possibilities is given for graphically displaying different properties of clusters. It turned out that adequate graphical representations play a vital role in the process of identifying promising further questions and next steps in a clustering oriented research process.

7 References

Ackerman, M., Ben-David, S. (2008). Measures of Clustering Quality: A Working Set of Axioms for Clustering. Proceedings of Neural Information Processing Systems (NIPS 2008). http://books.nips.cc/papers/files/nips21/NIPS2008_0383.pdf.

Ackerman, M., Ben-David, S. (2009). Which Data Sets are 'Clusterable'? – A Theoretical Study of Clusterability. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, 2009. http://www.cs.uwaterloo.ca/~shai/publications/ability_submit.pdf.

Agrafiotis, D.K., Bandyopadhyay, D., Farnam, M. (2007). Radial Clustergrams: Visualizing the aggregate properties of hierarchical clusters. Journal. Chem. Inf. Model., Vol. 47, 69–75.

Aldenderfer, M.S., Blashfield, R.K. (1976). Cluster Analysis. Sage, Beverly Hills, CA.

Anselin, L. (1995). "Local indicators of spatial association – LISA". Geographical Analysis, 27, 93–115.

Anselin, L. (2005). "Exploring Spatial Data with GeoDATM: A Workbook". Spatial Analysis Laboratory. p. 138. http://www.csiss.org/clearinghouse/GeoDa/geodaworkbook.pdf.

Anselin, L., Kim, Y.-W., Syabri, I. (2004b). Web-based analytical tools for the exploration of spatial data. Journal of Geographical Systems, 6, 197–218.

Bao, S., Henry, M.S. (1996). "Heterogeneity issues in local measurements of spatial association." Geographical Systems, 1996, Vol. 3, 1–13.

Bao, S., Martin, D. (1997). Integrating S-PLUS with ArcView in Spatial Data Analysis: An Introduction to the S+ArcView Link, ESRI's Users Conference, San Diego, CA.

Bao, S. (1999). Literature Review of Spatial Statistics and Models. China Data Center, http://141.211.136.209/cdc/docs/review.pdf.

Bao, S., Li, B. (2000). Spatial Statistics in Natural Resources, Environment and Social Sciences (eds.), A Special Issue of the Journal of Geographic Information Science.

Bação, F., Lobo1,V., Painho, M. (2005). Self-organizing Maps as Substitutes for K-Means Clustering. V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, 476–483, 2005.

Barbakh, W., Fyfe, C. (2008). Local vs global interactions in clustering algorithms: Advances over K-means. International Journal of Knowledge-based and Intelligent Engineering Systems, 12, 1–17.

Belabbas, M.-A., Wolfe, P.J. (2009a). Spectral methods in machine learning and new strategies for very large datasets. PNAS January 13, 2009, Vol. 106 (2), 369–374.

Belabbas, M.-A., Wolfe, P.J. (2009b).On landmark selection and sampling in high-dimensional data analysis, in Philosophical Transactions, Series A, of the Royal Society 367 (2009), 4295–4312.

Ben-Hur, A., Elisieeff, A., Guyon, I. (2002). A stability based method for discovering structure in clustered data. Pac Symp Biocomput. 2002, 6–17.

Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V. (2001). Support vector clustering. J. Mach. Learn. Res., 2 ,125–137.

Bezdek, J.C., Hathaway, R.J. (2002). VAT: A Tool for Visual Assessment of (Cluster) Tendency, Proc. IJCNN 2002, IEEE Press, Piscataway, N.J., 2225–2230.

Bezdek, J.C., Hathaway, R.J., Huband, J.M. (2007). Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices. IEEE Trans. On Fuzzy Systems, Vol. 15 (5), 890–903.

Bezdek, J.C., Pal, N.R. (1998). Some new indexes of cluster validity. IEEE Trans Syst Man Cybern B Cybern 1998, 28 (3), 301–315.

Bies, B., Dabbs, K., Zou, H. (2009). On Determining The Number Of Clusters – A Comparative Study. Paper during 2009 IMA Interdisciplinary Research Experience for Undergraduates, June 28 to July 31.
http://www.ima.umn.edu/~iwen/REU/paper4.pdf.

Blum, A.L., Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, Vol. 97, 245–271.

Bolshakova, N., Azuaje, F. (2003). Cluster validation techniques for genome expression data. Signal Processing 2003, 83, 825–833.

Bolshakova, N., Azuaje, F., Cunningham, P. (2005a). A knowledge-driven approach to cluster validity assessment. Bioinformatics 2005, 21, 2546–2547.

Bolshakova1, N., Azuaje, F., Cunningham, P. (2005b). An integrated tool for microarray data clustering and cluster validity assessment. Bioinformatics. Vol. 21 (4), 451–455.

Breiman, L. (2001). Random forests. Machine Learning, 45 (1), 5–32.

Brock, G., Pihur, V., Datta, S., Datta, S. (2008). clValid, an R package for cluster validation. http://louisville.edu/~g0broc01/.

Brys, G. (2006). Finding groups in a diagnostic plot. In: COMPSTAT 2006, Proceedings in Computational Statistics.

Cai, W., Chen, S., Zhang, D. (2009). A simultaneous learning framework for clustering and classification. Pattern Recognition, Vol. 42 (7), 1248–1259.

Camastra, F. (2003). Data Dimensionality Estimation Methods: A Survey. Pattern Recognition, Vol. 36 (12), 2945–2954, Elsevier Science, Amsterdam, (2003).

Camastra, F., Verri, A. (2005). A novel kernel method for clustering. IEEE Transaction on PAMI, Vol. 27, 801–805.

Campbell, V., Legendre, P., Lapointe, F.-J. (2009). Assessing Congruence Among Ultrametric Distance Matrices. Journal of Classification, Vol. 26, 103–117.

Celeux, G., Govaert, G. (1995). Gaussian parsimonious clustering models, Pattern Recognition, 28, 781–793.

Chang, W.-C. (1983). On Using Principal Components Before Separating a Mixture of two Multivariate Normal Distributions. Applied Statistics, 32, 267–275.

Cheng, R., Milligan, G.W. (1995). Mapping Influence Regions in Hierarchical Clustering Multivariate Behavioral Research, Vol. 30.

Cheng, R., Milligan, G.W. (1996a). Measuring the influence of individual data points in a cluster analysis. Journal of Classification. Vol. 13 (2), 1432–1343.

Cheng, R., Milligan, G.W. (1996b). K-means clustering with influence detection. Educational and Psychological Measurement, Vol. 56, 833–838.

Cheng, Y., Church, G.M. (2000). Biclustering of expression data. Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 93–103.

Cristianini, N., Shawe-Taylor, J., Kandola, J. (2002). Spectral kernel methods for clustering. In NIPS 14, 2002.

Damian, D., Orešič, M., Verheij, E., Meulman, J., Friedman, J., Adourian, A., Morel, N., Smilde, A., van der Greef, J. (2007). Applications of a new subspace clustering algorithm (COSA) in medical systems biology, Metabolomics 3.

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

De Soete, G. (1988). OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting. Journal of Classification, 5, 101–104.

Dixon, J.K. (1979). Pattern recognition with partly missing data. IEEE Transactions on Systems, Man and Cybernetics SMC 9, 617–621.

Donoho, D., Jin, J. (2008). Higher criticism thresholding: Optimal feature selection when useful features are rare and weak. Proceedings of the National Academy of Sciences, Vol. 105, 14790–14795.

Donoho, D., Jin, J. (2009). Feature selection by higher criticism thresholding achieves the optimal phase diagram. Phil Trans R Soc A, 367, 4449–4470.

Drineas, P., Mahoney, M.W. (2005). On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. Journal of Machine Learning Research, Vol. 6, 2153–2175.

Dudoit, S, Fridlyand, J. (2002a). A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biology, Vol. 3 (7).

Dudoit, S., Fridlyand, J., Speed, T.P. (2002b). Comparison of discriminant methods for the classification of tumors using gene expression data. J. Am. Stat. Assoc., 97, 77–87.

Everitt, B.S., Dunn, G. (2001). Applied Multivariate Data Analysis. (Second Edition). Hodder Education.

Everitt, B.S., Landau, S., Leese, M. (2001). Cluster Analysis. Fourth edition. Arnold Publishers.

Ewing, R.M., Sherry, J.M. (2001). Visualization of expression clusters using Sammonb's non-linear mapping. Bioinformatics, Vol. 17, 658–659.

Filippone, M., Camastra, F. Masulli, F., Rovetta, S. (2007). A survey of kernel and spectral methods for clustering. Pattern Recognition. Vol. 41 (1), 176–190.

Flexer, A. (2001). On the use of self-organizing maps for clustering and visualization. Intelligent Data Analysis, 5, 373–384.

Fodor, I.K. (2002). A survey of dimension reduction techniques. (pdf file) US-department of Energy. https://e-reports-ext.llnl.gov/pdf/240921.pdf.

Fowlkes, E.B., Gnanadesikan, R., Kettenring, J.R. (1988). Variable selection in clustering. Journal of Classification, 5, 205–228.

Fraiman, R., Justerl, A., Svarc, M. (2008). Selection of Variables for Cluster Analysis and Classification Rules. Journal of the American Statistical Association September 2008, Vol. 103 (483), 1294–1303.

Fraley, C., Raftery, A.E. (1998). How many clusters? Which clustering method? - Answers via Model-Based Cluster Analysis. Computer Journal, 41, 578–588.

Fraley, C., Raftery, A.E. (1999). MCLUST: Software for model-based clustering. Journal of Classification, 16, 297–306.

Fraley, C., Raftery, A.E. (2002). Model-Based Clustering, Discriminant Analysis, and Density Estimation. Journal of the American Statistical Association, 97, 611–612.

Fraley, C., Raftery, A.E. (2003). Enhanced model-based clustering, density estimation and discriminant analysis software: MCLUST. Journal of Classification, 20, 263–296.

Friedman, J.H., Meulman, J.J. (2004). Clustering objects on subsets of attributes (with discussion). Journal of the Royal Statistical Society Series B (Statistical Methodology), 66 (4), 815–849.

Gat-Viks, I., Sharan, R., Shamir, R. (2003). Scoring clustering solutions by their biological relevance. Bioinformatics, Vol. 19 (18), 2381–2389.

Geary, R. (1954). The contiguity ratio and statistical mapping. Incorporated Statistician, Vol. 5, 115–145.

Gentleman, R.C., et al. (2004). Bioconductor: open software development for computational biology and bioinformatics, Genome Biology, 2004, 5:R80.

Getis, A., Ord, J.K. (1992). The analysis of spatial association by use of distance statistics. Geographical Analysis, Vol. 24 (3), 189–206.

Getis, A., Ord, J.K. (1996). Local spatial statistics: an overview. In: P. Longley and M. Batty (eds.) “Spatial analysis: modeling in a GIS environment” (Cambridge: Geoinformation International), 261–277.

Gionis, A., Mannila, H., Tsaparas, P. (2005). Clustering Aggregation. 21st International Conference on Data Engineering (ICDE 2005).

Girolami, M. (2002). Mercer Kernel-Based Clustering in Feature Space. IEEE Transactions on Neural Networks, 13 (3), 780–784.

Girvan, M., Newman, M.E.J. (2002). Community structure in social and biological networks. PNAS June 11, 2002, Vol. 99 (12), 7821–7826.

Gnanadesikan, R., Kettenring, J.R., Tsao, S.L. (1995). Weighting and Selection of Variables for Cluster Analysis. Journal of Classification, 12, 113–136.

Gnanadesikan, R., Kettenring, J.R., Maloor, S. (2007). Better alternatives to current methods of scaling and weighting data for cluster analysis. Journal of Statistical planning and Inference, Vol. 137, 3483–3496.

Goder, A., Filkov, V. (2008). Consensus Clustering Algorithms: Comparison and Refinement. Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX) — San Francisco, January 19, 2008. Society for Industrial and Applied Mathematics.

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

Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, Vol. 3, 1157–1182.

Günter, S, Bunke, H. (2003). Validation indices for graph clustering. Pattern Recognition Letters, Vol. 24, 1107–1113.

Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.) (2006). Feature Extraction, Foundations and Applications. Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, 2006.

Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2001). On clustering validation techniques. Journal of Intelligent Information Systems, 17, 107–145.

Han, J., Kamber, M. (2006). Data mining: concepts and techniques. Second Edition. Morgan Kaufmann, 2006.

Handcock, M.S., Raftery, A.E., Tantrum, J. (2007). Model-based clustering for social networks (with Discussion). Journal of the Royal Statistical Society, Series A, 170, 301–354.

Handcock, M.S., Hunter, D.R., Butts, C.T., Goodreau, S.M., Morris, M. (2009). statnet: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data. J Stat Softw. 2008, 24 (1), 1548–7660.

Handl, J., Knowles, J. (2004). Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPSYSBIO-2004-02. UMIST, Manchester, UK.

Handl, J., Knowles, J. (2005a). Multiobjective clustering around medoids. Proceedings of the Congress on Evolutionary Computation (CEC 2005). Vol. 1, 632–639. Copyright IEEE Press.

Handl, J., Knowles, J. (2005b). Improving the scalability of multiobjective clustering. Proceedings of the Congress on Evolutionary Computation (CEC 2005). Vol. 3, 2372–2379. Copyright IEEE Press.

Handl, J., Knowles, J. (2005c). Exploiting the trade-off – the benefits of multiple objectives in data clustering. Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), 547–560, LNCS 3410.

Handl, J., Knowles, J. (2006a). Multiobjective clustering and cluster validation. In Multiobjective machine learning edited by Yaochu Jin. Springer Series on Computational Intelligence 16, 21–47.

Handl, J., Knowles, J. (2006b). Feature subset selection in unsupervised learning via multiobjective optimization. International Journal of Computational Intelligence Research, 2 (3), 217–238.

Handl, J., Knowles, J. (2007). An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation, 11 (1), 56–76.

Handl, J., Knowles, J., Kell, D.B. (2005). Computational cluster validation in post-genomic data analysis. Bioinformatics, Vol. 21 (15), 3201–3212.

Handl, J., Knowles, J., Kell, D.B. (2007). Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Transactions on Computational Biology, Vol. 4, 279–292.

Hastie, T., Tibshirani, R., Friedman, J. (2001). Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer-Verlag, New York.

Hinton, G.E., Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. Science, 313, 504–507.

Hothorn, T., Hornik, K., Zeileis, A. (1996). party: A Laboratory for Recursive Part(y)itioning. [http://CRAN.R-project.org/package=party]. R package version 0.9-96.

Hothorn, T., Hornik, K., Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. Journal of Computational and Graphical Statistics 2006, 15 (3), 651–674.

Hubert, L., Arabie, P. (1985). Comparing Partitions. Journal of Classification, Vol. 2, 193–218.

Hu, Y., Hathaway, R.J. (2008). An Algorithm for Clustering Tendency Assessment. WSEAS TRANSACTIONS on MATHEMATICS, Vol. 7 (7), 441–450, 2008.

Huang, J.Z., Ng, M.K., Rong, H., Li, Z. (2005). Automated Variable weighting in k-Means type clustering. IEEE T-on Pattern Analysis and Machine Intelligence, Vol. 27 (5), may 2005, 657–668.

Hubert, L., Schultz, J. (1976). Quadratic assignment as a general data-analysis strategy. British Journal of Mathematical and Statistical Psychologie, 29, 190–241. http://machaon.karanagai.com/validation_algorithms.html.

Hubert, M., Vandervieren, E. (2008). An adjusted boxplot for skewed distributions, Computational Statistics and Data Analysis, 52, 5186–5201.

Hubert, M., Van der Veeken, S. (2008). Outlier detection for skewed data, Journal of Chemometrics, 22, 235–246.

Hubert, L., Arabie, P. (1985). Comparing Partitions. Journal of Classification, Vol. 2, 193–218.

Hurley, C.B. (2004). Clustering Visualizations of Multidimensional Data, Journal of Computational & Graphical Statistics, Vol. 13 (4), 788–806.

Huson, D.H., Richter, D.C., Rausch, C., Dezulian, T., Franz, M., Rupp, R. (2007). Dendroscope: An interactive viewer for large phylogenetic trees. BMC Bioinformatics, 8, 460.

Irigoien, I., Arenas, C. (2008). INCA: New statistic for estimating the number. Statist. Med., 27, 2948–2973.

Jain, A.K., Dubes, R.C. (1988). Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, 1988.

Jacquez, G.M., Jacquez, J.A. (1999). Disease clustering for uncertain locations. In: Disease mapping and risk assessment for public health. A.B. Lawson, A. Biggeri, D. Bohning, E. Lesaffre, J.-F. Viel, and R. Bertollini, eds. New York: John Wiley & Sons.

Jacquez, G.M. (2008). Spatial Cluster Analysis. Chapter 22 In "The Handbook of Geographic Information Science", S. Fotheringham and J. Wilson (Eds.). Blackwell Publishing, 395–416.

John, G.H., Kohavi, R., Pfleger, K. (1994). Irrelevant features and the subset selection problem. Volume 129. New Brunswick, NJ, USA, Morgan Kaufmann; 1994.

Jolliffe, I.T. (2002). Principal Component Analysis. 2nd-edition. New York: Springer-Verlag.

Jonnalagadda, S., Srinivasan, R. (2009). NIFTI: An Evolutionary Approach for Finding Number of Clusters in Microarray Data. BMC Bioinformatics, Vol. 10, p 40.

Kannan, R., Vempala, S., Vetta, A. (2004). On clusterings: Good, bad and spectral. Journal of the ACM, 51 (3), 497–515.

Kaufman, L. Rousseeuw, P.J. (1990). Finding Groups in Data. John Wiley and Sons. New York.

Kemp, C., Tenenbaum, J.B. (2008). The discovery of structural form. Proc. Natl. Acad. Sci. USA 2008, 105, 10687–10692.

Kerr, M.K., Churchill, G.A. (2001). Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. PNAS, Vol. 98 (16), 8961–8965.

Kerr, G., Ruskin, H.J., Crane, M. (2007). Pattern Discovery in Gene Expression Data. Intelligent Data Analysis: Developing New Methodologies Through Pattern Discovery and Recovery.

Kerr, G., Ruskin, H.J., Crane, M., Doolan, P. (2008). Techniques for Clustering Gene Expression Data. Computers In Biology And Medicine, 38 (3), 283–293.

Kettenring, J.R. (2006). The practice of cluster analysis, J. Classif., 23, 3–30.

Kleinberg, J. (2002). An Impossibility Theorem for Clustering. Advances in Neural Information Processing Systems (NIPS) 15, 2002.

Kohavi, R, John, G.H. (1998). Wrappers for Feature Subset Selection. Artificial Intelligence, Vol. 97 (1–2), 273–324.

Kohavi, R, John, G.H. (1998). The Wrapper Approach. In "Feature Extraction, Construction and Selection: a data mining perspective", eds. Liu, H., Motoda, H.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43, 59–69.

Kok, M.T.J., Lüdeke, M.K.B., Sterzel, T., Lucas, P.L., Walter, C., Janssen, P., de Soysa, I. (2010). Quantitative analysis of patterns of vulnerability to global environmental change. Den Haag: Netherlands Environmental Assessment Agency (PBL) 90 p.

Krzanowski, W.J., Hand, D.J. (2009). A simple method for screening variables before clustering microarray data. Computational Statistics and Data Analysis, Vol. 43, 2747–2753.

Kulis, B., Basu, S., Dhillon, I.S., Mooney, R.J. (2009a). Semi-Supervised Graph Clustering: A Kernel Approach. Machine Learning, Vol. 74 (1), 1–22, January 2009.

Kulis, B., Sustik, M.A., Dhillon, I.S. (2009b). Low-Rank Kernel Learning with Bregman Matrix Divergences. Journal of Machine Learning Research, Vol. 10, 341–376.

Law, M.H.C., Jain, A.K. (2006). Incremental Nonlinear Dimensionality Reduction By Manifold Learning. IEEE Transactions of Pattern Analysis and Intelligence. Vol. 28, 377–391.

Leisch, F. (2006). A toolbox for k-centroids cluster analysis. Comput. Stat. Data Anal., 51 (2), 526–544.

Leisch, F. (2008). Visualizing cluster analysis and finite mixture models. In: Chen, C., Härdle, W., Unwin, A. (eds.) Handbook of Data Visualization. Springer Handbooks of Computational Statistics. Springer, Berlin (2008). ISBN 978-3-540-33036-3.

Leisch, F. (2009). Neighborhood graphs, stripes and shadow plots for cluster visualization. Statistics and Computing, 2009. to appear.

Lerner, B., Guterman, H., Aladjem, M., Dinstein, I. (2000). On the Initialisation of Sammon's Nonlinear Mapping. Pattern Analysis & Applications, Vol. 3, 61–68.

Li, G., Ma, Q., Tang, H., Paterson, A. H., Xu, Y. (2009). QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Res., August 1, 2009; 37(15): e101 - e101.

Liaw, A., Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18–22. URL http://CRAN.R-project.org/doc/Rnews/.

Little, R.J.A., Rubin, D.A. (1987). Statistical analysis with missing data. John Wiley and Sons.

Liu, H., Yu, L. (2005). Towards integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 17 (3), 1–12.

Luxburg, U. von. (2007). A tutorial on spectral clustering. Statistics and Computing, 17 (4), 395–416.

Luxburg, U. von. (2010). Clustering stability: an overview. Foundations and Trends in Machine Learning, Vol. 2 (3), 235–274,
URL (30-08-2010): http://arxiv.org/abs/1007.1075.

Luxburg, U. von., Belkin, M., Bousquet, O. (2008). Consistency of Spectral Clustering. Annals of Statistics 36 (2), 555–586.

MacCuish, J., Nicolaou, C., MacCuish, N.E. (2001). Ties in proximity and clustering compounds. J. Chem. Inf. Comput. Sci., 41, 134–146.

Madeira, S.C., Oliveira, A.L. (2004). Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE Transactions on Computational Biology and Bioinformatics, 1 (1), 24–45.

Mahoney, M.W., Drineas, P. (2009). CUR matrix decompositions for improved data analysis. PNAS January 20, 2009, Vol. 106 (3), 697–702.

Makarenkov, V., Legendre, P. (2001). Optimal Variable Weighting for Ultrametric and Additive Trees and K-means Partitioning: Methods and Software. Journal of Classification, 18, 245–271.

Maruca, S.L., Jacquez, G.M. (2002). Area-based tests for association between spatial patterns. Journal of Geographic Systems, 4 (1), 69–83.

McCullagh, M. J. (2006). Detecting Hotspots in Time and Space. ISG06.

McLachlan, G., Peel, D., Basford, K.E., Adams, P. (2000). The EMMIX software for the fitting of mixtures of normal and t-components. Journal of Statistical Software, 4, 1–14.

McQuitty, L.L. (1966). Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data. Educational and Psychological Measurement, 26, 825–831.

Meila, M. (2007). Comparing clusterings – an information based distance. Journal of Multivariate Analysis, 98, 873–895.

Melnykov, V. Maitra, R. (2010). Finite mixture models and model-based clustering. Statistics Surveys, 2010, Vol. 4, 80–116.

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

Milligan, G.W., Cooper, M.C. (1985). An examination of procedures for determining the number of clusters in a data set. Psychometrika, 50, 159–179.

Milligan, G.W., Cooper, M.C. (1987). Methodology Review: Clustering Methods. Applied Psychological Measurement, Vol. 11 (4), 329–354.

Milligan, G.W., Mahajan, V. (1980). A note on procedures for testing the quality of a clustering of a set of objects. Decision Sciences, 11, 669–677.

Milligan, G.W. (1989). A validation study of a variable weighting algorithm for cluster analysis. Journal of Classification, 6 (1), 53–71.

Milligan, G.W. (1996). Clustering validation: results and implications for applied analyses. In P. Arabie, L. J. Hubert, and G. D. Soete, editors, In Clustering and Classication., pages 341–375. World Scientic Publishing, River Edge, NJ, 1996.

Mingoti, S.A., Lima, J.O. (2006). Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms. European Journal of Operational Research, 174, 1742–1759.

Mirkin, B. (2005). Cluster Analysis for Data Mining: A Data Recovery Approach. CRC Press.

Mishra, N., Schreiber, R., Stanton, I., Tarjan, R.E. (2007). Clustering Social Networks A. Bonato and F.R.K. Chung (Eds.): WAW 2007, LNCS 4863, pp. 56–67, 2007.

Moguerza, J.M., Muñoz, A., Martin-Merino, M. (2002). Detecting the number of clusters using a support vector machine approach. Proc. International Conference on Artficial Neural Networks. Lecture Notes in Comput. Sci. 2415. 63.768. Springer, Berlin.

Monti, S., Tamayo, P., Mesirov, J., Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Mach. Learn., 52, 91–118.

Moran, P.A.P. (1948). The interpretation of statistical maps. Journal of the Royal Statistical Society, Series B., Vol. 10, 243–251.

Morgan, B.J.T., Ray, A.P.G. (1995). Non-uniqueness and inversions in cluster analysis. Applied Statistics, 44, 117–34.

Moya-Anegón, F., Herrero-Solana, V., Jiménez-Contreras, E. (2006). A connectionist and multivariate approach to science maps: the SOM, clustering and MDS applied to library and information science research Journal of Information Science, 32 (1) 2006, 63–77.

Murtagh, F., Hernández-Pajares, M. (1995). The Kohonen self-organizing map method: An assessment. Journal of Classification. Vol. 12 (2), 165–190.

Nardo, M., Saisana, M., Saltelli, A., Tarantola, S., Hoffman A., Giovannini, E., (2005). Handbook on Constructing Composite Indicators: Methodology and User Guide. OECD Statistics Working Papers.

Nelson, T.A., Boots, B. (2008). Detecting spatial hot spots in landscape ecology. Ecography, Vol. 31 (5), 556–566.

Newman, M.E.J. (2003). The structure and function of complex networks. SIAM review, 2003 – JSTOR.

Newman, M.E.J. (2004). Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 066133.

Newman, M.E.J., Leicht, E.A. (2007). Mixture models and exploratory analysis in networks. PNAS, Vol. 104 (23), 9564–9569.

Ng, A.Y., Jordan, M., Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems (NIPS), 2002.

Norg, I., Groenen, P. (1997). Modern multidimensional scaling theory and applications. New York: Springer Verlag.

Olex, A.L., John, D.J., et al. (2007). Additional limitations of the clustering validation method figure of merit. 45th ACM Southeast Annual Conference, Winston-Salem, NC.

Ord, J.K., Getis, A. (2001). Testing for local spatial autocorrelation in the presence of global autocorrelation. Journal of Regional Science, Vol. 41 (3), 411–432.

Palla, G., Derényi, I., Farkas, I., Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435, 814–818.

Pipino, L.L., Funk, J.D., Wang, R.Y. (2006). Journey to Data Quality. MIT Press Ltd, 2006.

Pison, G., Struyf, A., Rousseeuw, P.J. (1999). Displaying a clustering with CLUSPLOT. Comput. Stat. Data Anal., 30, 381–392 ftp://ftp.win.ua.ac.be/pub/preprints/99/Disclu99.pdf.

Premo, L.S. (2004). Local spatial autocorrelation statistics quantify multi=scale patterns in distributional data: an example from the Maya Lowlands. Journal of Archaeological Science, Vol. 31, 855–866.

Raftery A.E., Dean, N. (2006). Variable Selection for Model-Based Clustering. Journal of the American Statistical Association, Vol. 101 (473), 168–178.

Rahm, E., Do, H.H. (2000). Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23 (4), 3–13.

Roth, V., Lange, T., Braun, M., Buhmann, J. (2002). A Resampling Approach to Cluster Validation.
http://informatik.unibas.ch/personen/roth_volker/PUB/compstat02.pdf

Rousseeuw, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math, 20, 53–65.

Rousseeuw, P.J., Ruts, I., Tukey, J.W. (1999). The bagplot: A bivariate boxplot. Am. Stat., 53 (4), 382–387.

Rousseeuw, P.J., Debruyne, M., Engelen, S., Hubert, M. (2006). Robustness and outlier detection in chemometrics, Critical Reviews in Analytical Chemistry, 36, 221–242.

Runkler, T.A. (2000). Information Mining - Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse. Vieweg, Wiesbaden, 2000.

Saeys, Y., Inza, I., Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. Bioinformatics. Vol. 23 (19), 2507–17.

Saitta, S., Raphael, B., Smith, I.F.C. (2007). A Bounded Index for Cluster Validity. In: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition, LNAI 4571, Springer Verlag, Heidelberg, pp. 174–187, 2007.

Saitta, S., Raphael, B., Smith, I.F.C. (2008). A Comprehensive Validity Index for Clustering. accepted for publication in the Journal of Intelligent Data Analysis, 2008.

Salvador, S., Chan, P. (2004). Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. Proc. 16th IEEE Intl. Conf. on Tools with AI, 576–584, 2004.

Saraiya, P., North, C., Duca, K. (2005). An insight-based methodology for evaluating bioinformatics vizualizations. IEEE Trans, on Visualization and Computer Graphics, Vol. 11, 443–456.

Scharl, T., Leisch, F. (2009). gcExplorer: Interactive Exploration of Gene Clusters. Bioinformatics, Vol. 25 (8), 1089–1090.

Schölkopf, B., Smola, A., Müller, K.-R. (1999). Kernel Principal Component Analysis, In: Bernhard Schölkopf, Christopher J. C. Burges, Alexander J. Smola (Eds.), Advances in Kernel Methods-Support Vector Learning, 1999, MIT Press Cambridge, MA, USA, 327-352. ISBN 0-262-19416-3.

Scholz, M., Kaplan, F., Guy, C.L., Kopka, J., Selbig, J. (2005). Non-linear PCA: a missing data approach. Bioinformatics, 21, 3887–3895.

Schonlau, M. (2002). The clustergram: a graph for visualizing hierarchical and non-hierarchical cluster analyses. The Stata Journal, 2002, 2 (4), 391–402.

Schonlau, M. (2004). Visualizing Hierarchical and Non-Hierarchical Cluster Analyses with Clustergrams. Computational Statistics: 2004, 19 (1), 95–111.

Shi, J., Malik, J. (2000). Normalized cuts and image segmentation. IEEE Trans. PAMI, 22 (8), 888–905.

Shi,T., et al. (2005). Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. Modern Pathology., 18, 547–557.

Shi, T., Horvath, S. (2006). Unsupervised Learning with Random Forest Predictors. Journal of Computational and Graphical Statistics. Vol. 15 (1), 118–138(21).

Sietz, D., Lüdeke, M.K.B., Walther, C. (2011). Categorisation of typical vulnerability patterns in global drylands. Global Environmental Change, 21, 431–440.

Silver, M. (1995). Scales of Measurement and Cluster Analysis: An Application Concerning Market Segments in the Babyfood market. The Statistician, Vol. 44 (1), 101–112.

Smyth, C.W., Coomans, D.H. (2006). Parsimonious Ensembles for Regression. The 38th Symposium on the Interface of Statistics, Computing Science and Applications: Massive Data Sets and Streams Interface Foundation of North America, Pasadena, California 54 – 54.

Smyth, C.W., Coomans, D.H., Everingham, Y.L. (2006a). Clustering noisy data in a reduced dimension space via multivariate regression trees. Pattern Recognition, Vol. 39, 424–431.

Smyth, C.W., Coomans, D.H., Everingham, Y.L., Hancock, T.P. (2006b). Auto-associative Multivariate Regression Trees for Cluster Analysis. Chemometrics and Intelligent Laboratory Systems, Vol. 80, 120–129.

Smyth, C.W., Coomans, D.H. (2007). Predictice weighting for cluster ensembles. Journal of Chemometrics, Vol. 21, 364–375.

Spaans, M., Heiser, W.J. (2005). Instability of hierarchical cluster analysis due to input order of the data: The PermuCLUSTER solution. Psychological Methods, 10 (4), 468–476.

Steinley, D. (2004). Properties of the Hubert-Arabie adjusted Rand index. Psychological Methods, 9, 386–396.

Steinley, D. (2006). K-means clustering: A half-century synthesis. British Journal of Mathematical and Statistical Psychology, 59, 1–34.

Steinley, D. (2008). Stability analysis in K-means clustering. British Journal of Mathematical and Statistical Psychology, 61, 255–273.

Steinley, D., Brusco, M.J. (2007). Initializing K-means batch clustering: A critical evaluation of several techniques. Journal of Classification, 24, 99–121.

Steinley, D., Brusco, M.J. (2008a). A new variable weighting and selection procedure for K-means cluster analysis. Multivariate Behavioral Research, Vol. 43, 77–108.

Steinley, D., Brusco, M.J. (2008b). Selection of Variables in Cluster Analysis: An Empirical Comparison of Eight Procedures. Psychometrika, Vol. 73 (1), 125–144.

Strehl, A., Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3, 583–617.

Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T, Zeileis, A. (2008). Conditional variable importance for random forests. BMC Bioinformatics 2008, 9, 307.

Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T. (2007). Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution. BMC Bioinformatics 2007, 8, 25.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information and Computer Sciences, 43, 1947–1958.

Tan, J., Zhang, J., Li, W. (2010). An Improved Clustering Algorithm Based on Density Distribution Function. Computer and Information Science, Vol. 3 (3), August 2010, 23–29. URL (31-08-2010): http://ccsenet.org/journal/index.php/cis/article/viewFile/6891/5426.

Tanay, R., Sharan, R., Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. Bioinformatics Vol. 18 (9), S136–S144.

Tenenbaum, J.B., de Silva, V., Langford, J.C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, Vol. 290, 2319–2323.

Tian, T., James, G., Wilcox, R. (2009). A Multivariate Adaptive Stochastic Search Method for Dimensionality Reduction in Classification. Annals of Applied Statistics, 4, 339–364.

Tibshirani, R., Walther, G., Hastie, T. (2001). Estimating the number of clusters in a dataset via gap statistic. Journal of Royal Statistical Society B 2001, 63, 411–423.

Tibshirani, R., Walther, G. (2005). Cluster Validation by Prediction Strength. Journal of Computational & Graphical Statistics, 14, 511–528.

Tsai, C.Y., Chiu, C.C. (2008). Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. Computational Statistics & Data Analysis, Vol. 52 (10), 4658–4672.

van der Laan, M. (2006). Statistical Inference for Variable Importance. International Journal of Biostatistics, 2 (1), 1008–1008.

Varshavsky, R., Gottlieb, A., Linial, M., Horn, D. (2006). Novel unsupervised feature filtering of biological data. Bioinformatics, Vol. 22, e507–e513.

Varshavsky, R., Gottlieb, A., Horn, D., Linial, M. (2007). Unsupervised feature selection under perturbations: meeting the challenges of biological data. Bioinformatics, Vol. 23 (24), 3343–3349.

Vesanto, J. (1999). SOM-based data visualization methods, Intelligent Data Analysis, 3 (2), 111–126.

Vesanto, J., Alhoniemi, E. (2000). Clustering of the Self-Organizing Map. IEEE Trans. On Neural Networks, Vol. 11, 586–600.

Vinh, N.X., Epps, J., Bailey, J. (2009). Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary? Proceedings of the 26th International Conference on Machine Learning, 2009.

Xing, E.P. (2003). Feature Selection in Microarray Analysis, in D.P. Berrar, W. Dubitzky and M. Granzow (Eds.), A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, 2003.

Xu, R., Wunsch, D. (2008). Clustering. IEEE Press Series on Computational Intelligence. John Wiley and Sons.

Yan, D., Huang, L., Jordan, M.I. (2009). Fast approximate spectral clustering. International Conference on Knowledge Discovery and Data Mining Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data Paris, France Pages 907–916.

Yeung, K.Y., Haynor, D.R., Ruzzo, W.L. (2001). Validating clustering for analysis for clustering gene expression data. Bioinformatics, Vol. 17 (4), 309–318.

Yeung, K.Y., Ruzzo, W.L. (2001). Principal component analysis for clustering gene expression data. Bioinformatics, Vol. 17 (9), 763–774.

Yiang, M.K.A., Kumar, A. (2005). A comparative analysis of an extended SOM network and K-means analysis. Journal International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol. 8 (1), 9–15.

Yu, L. (2007). Feature Selection for Genomic Data Analysis. In H. Liu, editor, Computational Methods for Feature Selection, Chapman and Hall/CRC Press, 2007.

Wagstaff, K.L., Laidler, V. (2005). Making the Most of Missing Values: Object Clustering with Partial Data in Astronomy. Astronomical Data Analysis Software and Systems XIV; ASP Conference Series 2005, P 2.1.25.

Waller, N.G., Kaiser, H.A., Illian, J.B., Manry, M. (1998). Cluster analysis with Kohonen neural networks. Psychometrika, Vol. 63, 5–22.

Winters-Hilt, S., Yelundur, A., McChesney, C., Landry, M. (2006). Support Vector Machine Implementations for Classification & Clustering. BMC Bioinformatics 2006, 7(Suppl 2):S4 doi:10.1186/1471-2105-7-S2-S4.

Winters-Hilt, S., Merat, S. (2007). SVM clustering. BMC Bioinformatics 2007, 8 (Suppl 7):S18 doi:10.1186/1471-2105-8-S7-S18.

Wu, C.-J., Kasif, S. (2005). GEMS: a web server for biclustering analysis of expression data. Nucleic Acids Research 2005 33(Web Server Issue):W596-W599.

Wu, K.L., Yang, M.S. (2005). A cluster validity index for fuzzy clustering, Pattern Recognition Lett., Vol. 26, 1275–1291.

Wu, K.L., Yang, M.S., Hsieh, J.N. (2009). Robust cluster validity indexes. Pattern Recognition. Vol. 42 (11), 2541–2550.

Zadeh, R.B., Ben-David, S. (2009). A Uniqueness Theorem for Clustering. Proceedings of UAI 2009.

Appendix A: The R software environment

R is a software environment for data manipulation, calculation, and graphical display, and serves both as an environment and a programming language. R is available as Free Software in source code form under the terms of the Free Software Foundation's GNU General Public License. R runs on a wide variety of platforms (Unix, Linux,Windows, MacOS, FreeBSD). Sources and binaries of R can be downloaded at http://www.r-project.org. Installation of R is very simple and a variety of packages can be added directly from the web site (e.g. Brock et al., 2008). R has a very active development community and many resources can be found including user guides, manuals, script samples, newsgroups, and mailing lists (e.g. Venables et al., 2002). Further an extensive amount of publications like Paradis (2002) or Maindonald (2008) exists. R is a command line application. Its integrated object oriented language allows for efficient data manipulation. Whereas use of R does require programming, scripts can be developed and used to automate analyses and provide additional functionality. Graphical user interface (GUI)s have been developed for certain applications to avoid user programming (see, for example, Rcommander).
R has an amazing variety of functions for cluster analysis, which is illustrated at the web-page http://cran.r-project.org/web/views/Cluster.html. In this background document we will present a number of examples implemented in R. See also appendix A, which illustratively highlights some functionality of R for performing cluster analysis.

Citing R:

R Development Core Team (2005). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL: http://www.R-project.org.

Brock, Pihur, Datta Su., Datta So, *clValid: An R Package for Cluster Validation*, Journal of Statistical Software, Volume 25, Issue 4, 2008

Maindonald, *Using R for Data Analysis and Graphics - Introduction, Code and Commentary*, Centre for Mathematics and Its Applications, Australian National University. 2008

Paradis, E., *R for Beginners*, Montpellier, 2002

Venables, Smith and the R Development Core Team *An Introduction to R,* Network Theory Limited, Bristol, 2002

Appendix B: Cluster analysis in R[27]

R has an amazing variety of functions for performing cluster analysis. In this appendix three of the many approaches will be described: hierarchical agglomerative, partitioning, and model based. While there are no best solutions for the problem of determining the number of clusters to extract, several approaches are given below.

Data preparation

Prior to clustering data, you may want to remove or estimate missing data and rescale variables for comparability.

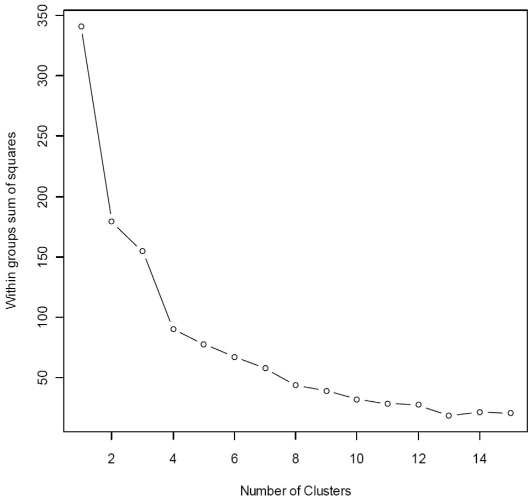
```
# Prepare Data
mydata <- na.omit(mydata) # listwise deletion of missing
mydata <- scale(mydata) # standardize variables
```

Partitioning

K-means clustering is the most popular partitioning method. It requires the analyst to specify the number of clusters to extract. A plot of the within groups sum of squares by number of clusters extracted can help determine the appropriate number of clusters. The analyst looks for a bend in the plot similar to a scree test in factor analysis. See Everitt & Hothorn (pg. 251).

Determine number of clusters

```
# Determine number of clusters
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
   centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```



[27] This appendix is taken from the information about QuickR (see http://www.statmethods.net/advstats/cluster.html). See also http://inference.us/SolutionPlatform/Documents/R/Cluster%20Analysis.pdf

K-Means cluster analysis

```
# 5 cluster solution
fit <- kmeans(mydata, 5) # 5 cluster solution

# get cluster means
aggregate(mydata,by=list(fit$cluster),FUN=mean)

# append cluster assignment
mydata <- data.frame(mydata, fit$cluster)
```

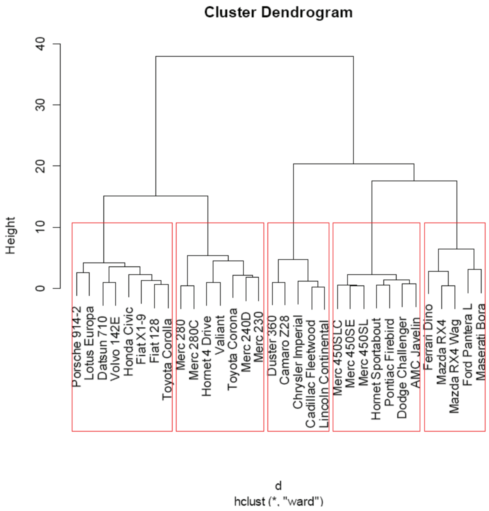
A robust version of **K-means** based on mediods can be invoked by using **pam( )** instead of **kmeans( )**. The function **pamk( )** in the **fpc** package is a wrapper for pam that also prints the suggested number of clusters based on optimum average silhouette width.

Hierarchical agglomerative

There are a wide range of hierarchical clustering approaches, and Ward's method described below is a popular one.

Ward hierarchical clustering

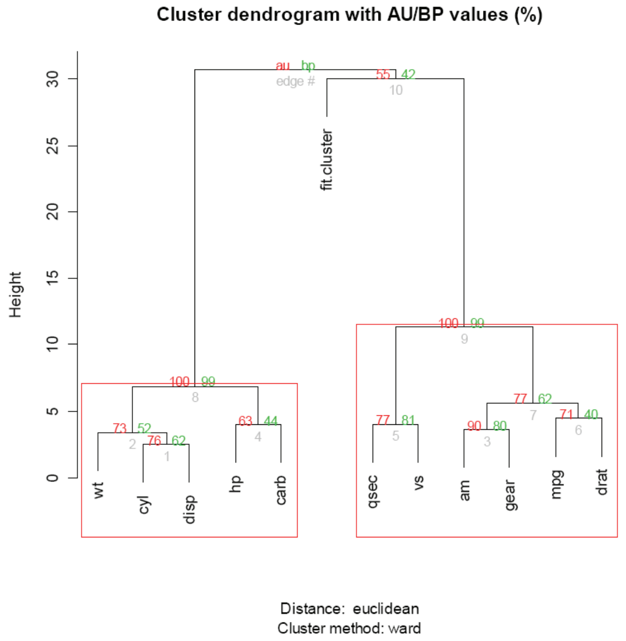
```
# Ward Hierarchical Clustering

# distance matrix
d <- dist(mydata, method = "euclidean")

fit <- hclust(d, method="ward")

# display dendogram
plot(fit)

# cut tree into 5 clusters
groups <- cutree(fit, k=5)

# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=5, border="red")
```



The **pvclust( )** function in the **pvclust** package provides p-values for hierarchical clustering based on multiscale bootstrap resampling. Clusters that are highly supported by the data will have large p values. Interpretation details are provided Suzuki[28]. Be aware that **pvclust** clusters columns, not rows. Transpose your data before using.

Ward hierarchical clustering with bootstrapped p values

```
# Ward Hierarchical Clustering with Bootstrapped p values

library(pvclust)
fit <- pvclust(mydata, method.hclust="ward",
   method.dist="euclidean")

# dendogram with p values

plot(fit)

# add rectangles around groups highly supported by the
data
pvrect(fit, alpha=.95)
```



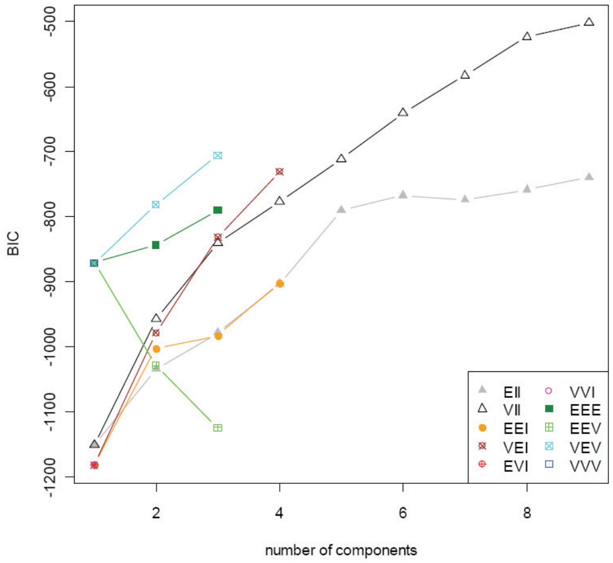
Model based approaches

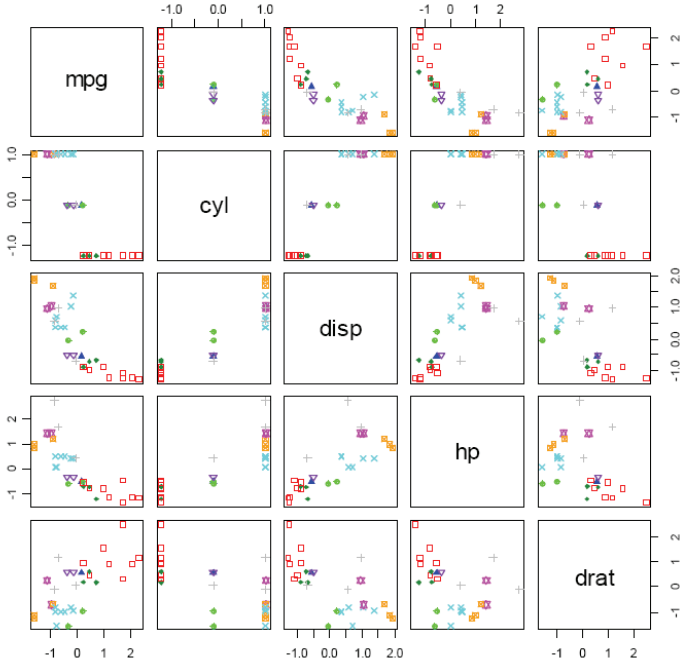
Model based approaches assume a variety of data models and apply maximum likelihood estimation and Bayes criteria to identify the most likely model and number of clusters. Specifically, the **Mclust( )** function in the **mclust** package selects the optimal model according to BIC for EM initialized by hierarchical clustering for parameterized Gaussian mixture models. (phew!). One chooses the model and number of clusters with the largest BIC. See help(mclustModelNames) to details on the model chosen as best.

[28] See http://www.is.titech.ac.jp/~shimo/prog/pvclust/

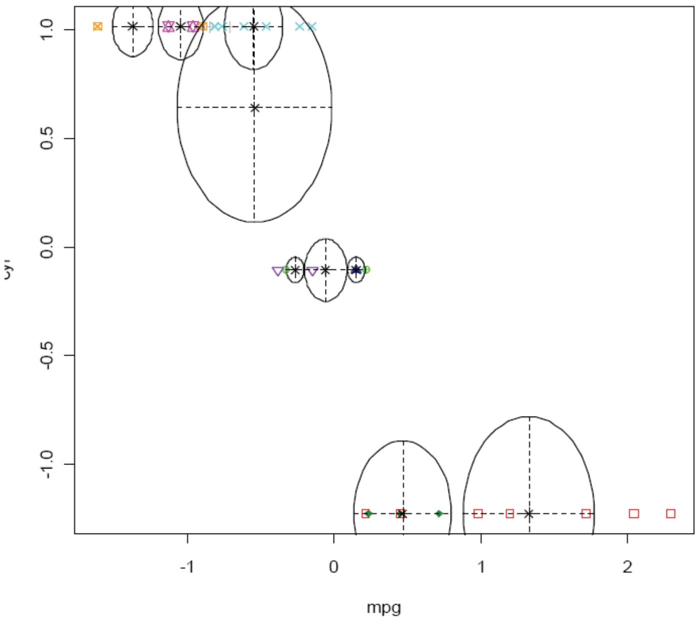
Model based clustering

```
# Model Based Clustering
library(mclust)
fit <- Mclust(mydata)

# plot results
plot(fit, mydata)

# display the best model
print(fit)
```

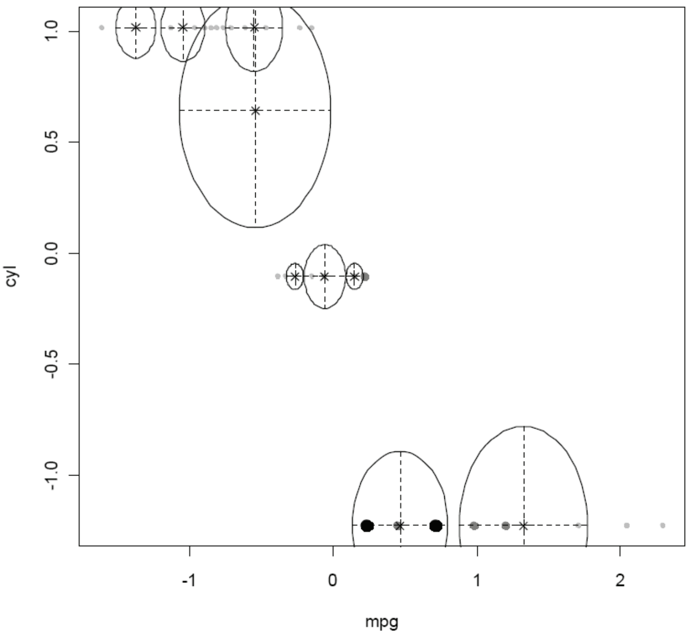




1,2 Coordinate Projection showing Classification



1,2 Coordinate Projection showing Uncertainty



Plotting cluster solutions

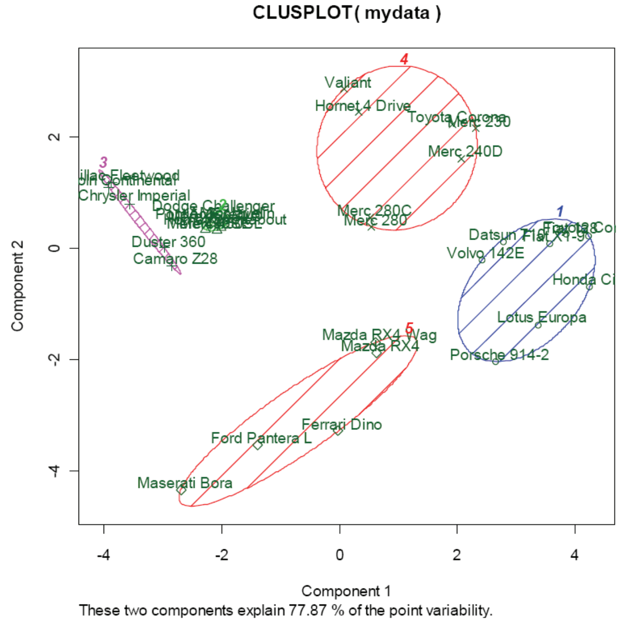
It is always a good idea to look at the cluster results.

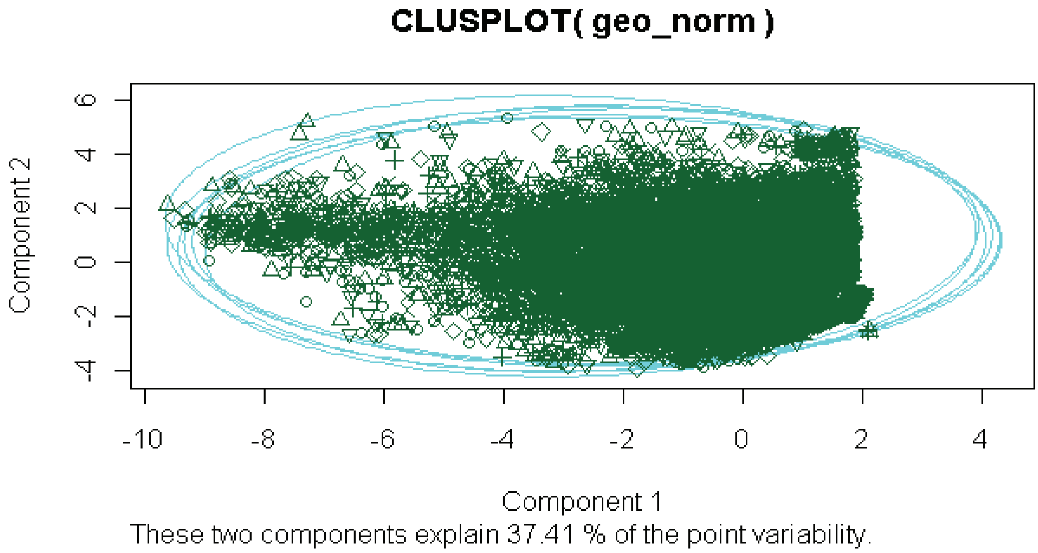
K-Means clustering with 5 clusters

```
# K-Means Clustering with 5 clusters
fit <- kmeans(mydata, 5)
```

Cluster plot against 1st 2 principal components

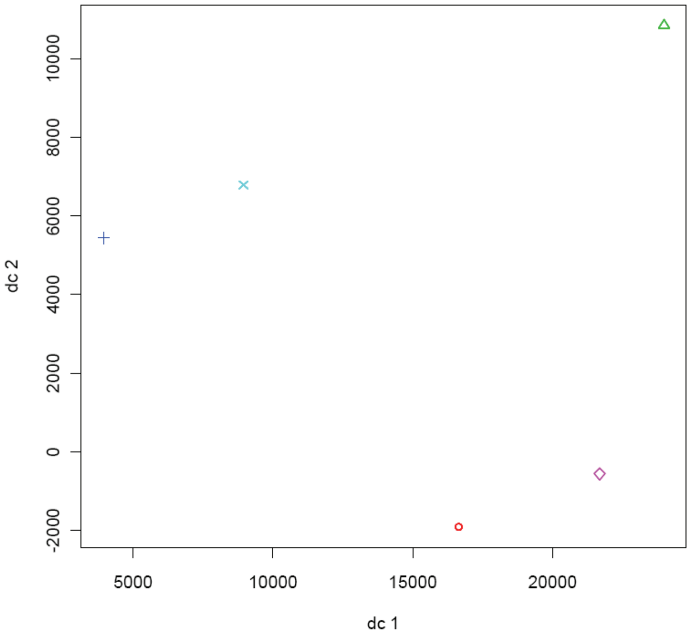
```
# Cluster Plot against 1st 2 principal components

# vary parameters for most readable graph
library(cluster)
clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)
```





Centroid plot against 1st 2 discriminant functions

```
# Centroid Plot against 1st 2 discriminant functions
library(fpc)
plotcluster(mydata, fit$cluster)
```



Validating cluster solutions

The function **cluster.stats()** in the **fpc** package provides a mechanism for comparing the similarity of two cluster solutions using a variety of validation criteria (Hubert's gamma coefficient, the Dunn index and the corrected rand index)

comparing 2 cluster solutions

```
# comparing 2 cluster solutions
library(fpc)
cluster.stats(d, fit1$cluster, fit2$cluster)
```

where **d** is a distance matrix among objects, and **fit1$cluster** and **fit$cluste**r are integer vectors containing classification results from two different clusterings of the same data.

Example R-script for clustering

The following R-Script is divided into four functions, called “consistency”, “sPIKcentres”, “initial” and “clus_graphs”. In the first function calls the loop for the overall repeating of the pair wise dissimilarity calculation and the loop for the size of the clustered partition. Further it performs the two of clusterings. The second function makes the dissimilarity calculation itself. The “initial” function is responsible for initialization of kmeans with hclust. And the last function delivers graphical representations of the cluster result.
In the last part of the script the user settings have to be chosen. The script can be used for calculation of the consistency measure and for the clustering of the subsequent best number of clusters.
The format of data has to be: rows represent the objects and columns represent the features of the objects.

```
## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
consistency <- function(Xdata,NmaxCluster,master,cm)  {

 Ndata <- dim(Xdata)[1]             # total number of datapoints
 Clust_cl <- matrix(0,Ndata,2)     # storing the cluster-class results for the two trial-clusterings
 X0 <- matrix(0,NmaxCluster,master)         # storing info on every run for consist.meas.
 C0 <- matrix(0,NmaxCluster,1)              # Init.matrix with average consist.meas.
 G0 <- matrix(0,Ndata,1)                    # Matrix for best cluster result
 ResMat <- list(MeanC=C0,SpecR=X0,Gold=G0) # global list for returning after calculation
 ifelse(cm,whl<-2,whl<-1)
 ifelse(cm,NminCluster<-2,NminCluster<-NmaxCluster)

 for (iOuter in 1:master) {              # outer-loop for comparing pairs of clusterings
  for (iClus in NminCluster:NmaxCluster) {            # Number of Clusters to be analysed
   for (iInner in 1:whl) {
    N_sel <- max(NmaxCluster,round(Ndata/200,0))
    ss  <- sample(1:Ndata)                  # random permutation of data set
    sss <- ss[1:N_sel]                      # First N_sel indices of random permutation
    Xdata_sel <- Xdata[sss,]
    while( length(unique(rowSums(Xdata_sel)))<iClus ) {
     ss  <- sample(1:Ndata)   ;  sss <- ss[1:N_sel]   ;   Xdata_sel <- Xdata[sss,] }
    centro    <- initial(Xdata_sel,iClus)
    indRand <- sample(1:Ndata)                        # reshuffling
    Xdata_shuffle <- Xdata[indRand,]                  # shuffled data
    cl_kmeans <- kmeans(Xdata_shuffle,centro,iter.max=50) # clustering with centro initializatin
    Clust_cl[indRand,iInner] <- cl_kmeans$cluster  # assign classes as indexed by non-shuffled data
   }
   ifelse(cm , {                         # Evaluate dissimilarities for the clustering-pairs
    ResMat$SpecR[iClus,iOuter] <- sPIKcentres(Xdata,Clust_cl[,1],Clust_cl[,2],Iheur=1)
    } , {
    for (j in 1:iClus) {                 # withinclustersum ~~~~~~~~~~~~~
     clu_diff <- 0
     clu_diff <- Xdata_shuffle[which(cl_kmeans$cluster==j),]-(matrix(1,cl_kmeans$size[j],1)
        %*%colMeans(Xdata_shuffle[which(cl_kmeans$cluster==j),]))
     ResMat$SpecR[iClus,iOuter] <- ResMat$SpecR[iClus,iOuter] + sum(clu_diff*clu_diff) }
    ifelse(ResMat$SpecR[iClus,iOuter]==min(ResMat$SpecR[iClus,1:iOuter]) , gold <- Clust_cl[,1]
    }    )  }   }
    ifelse (cm , { for (iClus in 2:NmaxCluster) { ResMat$MeanC[iClus,] <-
        with(ResMat,mean(SpecR[iClus,1:master])) } },# average-value for consistency measure
                 { ResMat$Gold <- gold } )
        return(ResMat)
}     # end function consistency

## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
sPIKcentres <-   function(dataCl,clust1,clust2,Iheur=1)  {
```

```
  Ncl1 <- max(clust1)      # maximum number of clusterclasses
  Ncl2 <- max(clust2)      # maximum number of clusterclasses
  Nclmin <- min(Ncl1,Ncl2) # minimum of Ncl1 and Ncl2

  ## Determine cluster centers       --> # matrix of cluster-centres for clustering 1 and 2
  cent1=rbind() ; for (i in 1:Ncl1) { ifelse(length(which(clust1==i))<2 , cent1 <-
          rbind(cent1,dataCl[which(clust1==i),]) , cent1 <-
          rbind(cent1,colMeans(dataCl[which(clust1==i),])) )}
  cent2=rbind() ; for (i in 1:Ncl2) { ifelse(length(which(clust2==i))<2 , cent2 <-
          rbind(cent2,dataCl[which(clust2==i),]) , cent2 <-
          rbind(cent2,colMeans(dataCl[which(clust2==i),])) )}

  ## Determine the distance matrix  of cluster-centers
  Distmat <- matrix(0,Ncl1,Ncl2)
  Distmat <- as.matrix(dist(rbind(cent1,cent2)))[1:Ncl1,(1:Ncl2)+Ncl1]
  ## Determination of association on basis of distances between clusters
  match.listb <- array(0,length<-Ncl2)     # initialising list for renaming clusters
  xft_tmp <- Distmat                      # storing Distmat in intermediate matrix
  xft_max <- max(xft_tmp)+1        # setting an upperlimit to values of xft_tmp
  for (d2 in 1:Nclmin) {
    cc <- which(xft_tmp==min(xft_tmp),arr.ind=T)[1,2]    # in which column is minimum (ref to clu1)
   rr <- which(xft_tmp==min(xft_tmp),arr.ind=T)[1,1]    # in which row is minimum (ref to clu2)
   match.listb[cc] <- rr   ## the cc-th cluster of clus.2  corresponds the to rr-th  cluster of the clus.1
   xft_tmp[rr,] <- xft_max ; xft_tmp[,cc] <- xft_max }
  match.listb[which(match.listb==0)] <-  max.col(-t(Distmat[,which(match.listb==0)]))
  clust2A <-  match.listb[clust2]    # second clustering in terms of its association with the first clus.
  res <- length(which(clust2A==clust1))/(length(clust1))  # count of fraction of replicates
  return(res) }

## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
initial <- function(Xdata_sel,k) { # function for initializing Kmeans

  geo_dist <- (dist(Xdata_sel))                      # distance matrix of part of data set
  cl_hcl   <- hclust(geo_dist,method="ward")# hclust with method: ward
  ser      <- as.vector(cutree(cl_hcl,k))           # cut the tree into k clusters
  cluster  <- list()                        # initializing to empty list
  for (i in 1:k) { cluster[[i]] <- which(ser==i) }
  centro <- matrix(ncol=ncol(Xdata_sel),nrow=k)   # storing cluster-centers
  for (i in 1:length(cluster)){
   for (j in 1:ncol(Xdata_sel)){
     centro[i,j] <- mean(Xdata_sel[cluster[[i]],j])
   }  }
   return(centro)  }

## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
clus_graphs <- function(gold,clu,clu_dim) {

  ## worldmap
  world <- matrix(scan("~/AT_CLUSTERUNG/R-Script+Data/geo_maske.dat"),ncol=1)  ## land mask
  for (z in 1:clu_dim[1]) {world[clus_dat[z,1]] <- gold[z]}
  x11(11,8) ; par(mar=c(2,2,2,1))
  is.na(world)<-which(world==0,arr.ind=T)       ## all zeros out
  z.a <- matrix(world,720,360)[,360:1]
  for (i in 0:(clu-1)) z.a[(i*20):(i*20+20),51:70]<-i+1
  farb<-c(rgb(0,0,0),rgb(1,0.6,0),rgb(1,1,0.3),rgb(0.5,0.5,0.5),rgb(0,1,0),rgb(0.5,0,0.5)
          ,rgb(1,0,0.3),rgb(0,0,1),rgb(0.2,1,1),rgb(1,0.5,0))
  farb <- farb[c(9,4,7,3,8,6,1,5,2,10)]
  image(1:720,1:360,z.a,col=c(grey(0.9),farb[1:clu]),xlim=c(0,720),ylim=c(50,360),
          main=paste("run.ident: ",round(min(clus_res$SpecR[clu,]),4),sep=""))
```

```
  x11(11,4);par(mfrow=c(1,clu),mar=c(2.5,1.8,1.4,0.3))
  size <- array(0,clu)
  for (j in 1:clu) {size[j]<-length(which(gold==j))}
  for (k in 1:clu) {
    bpdata <- as.data.frame(clus_dat[,feat])
    bpdata[which(gold!=k,arr.ind=T),]<-NA
    bpdata <- na.omit(bpdata)
    boxplot(bpdata, whisklty=0, staplelty=0, col=farb[k], outline=F, main=paste("C",k,": ",
             size[k]))->boxinfo
    for (dd in 1:ncol(bpdata)) {
      cen <- quantile(bpdata[,dd],  probs=c(5,95)/100)
      segments(dd,boxinfo$stats[4,dd],dd,as.numeric(cen[2]),col="black",lwd=1,lty=3)
      segments(dd,boxinfo$stats[2,dd],dd,as.numeric(cen[1]),col="black",lwd=1,lty=3)
      points(dd,as.numeric(cen[1]),col="black",pch=1)
      points(dd,as.numeric(cen[2]),col="black",pch=1)
    }
     mean.cl <- c(colMeans(bpdata,na.rm=T))
    points(c(1:length(feat)),mean.cl,pch=1,col=9,cex=1.4)
  } }
###############################################################
## PARAMETERS THAT HAVE TO BE DEFINED BY USER ~~~~~~~~
##
  namIndicat <- "choose name"
  namIndDir =   "choose directory"
  colIndFile <- 9
  featurenames <- c("choose list of feature names")
  feat <- c(3:9)                ## feature columns - for clustering !
  NmaxCluster <- 8              ## choose as upper boundary for consistency measure calculation or already
                                ## as value for best cluster result
  cm = T                        ## consistency measure calculation or only best cluster number clustering
##
##########################

namIndFile = paste(namIndDir,namIndicat,sep="")    ## reading data ~~~~~~~~~~~~~~~~
clus_dat <- matrix(scan(namIndFile,sep=""),ncol=colIndFile,byrow=T)
clu_dim  <- dim(clus_dat)

is.na(clus_dat) <- which(clus_dat==-9999,arr.ind=T) ## erase missing values ~~~~~~~~~~
clus_dat <- na.omit(clus_dat)

x11(7,4);par(mar=c(2.1,4,2.3,0.5),mfrow=c(3,3))        ## Histogramm of Cluster Data ~~~~~
for (i in feat)  hist(clus_dat[,i],main=featurenames[i])

ifelse(cm,master<-200,master<-50)
clus_res <- consistency(clus_dat[,feat],NmaxCluster,master,cm) ## Clustering ~~~~~~~~~

## Ploting of Result ~~~~~~~~~~~~~
if(cm) {x11(6,4);plot(c(2:NmaxCluster),clus_res$MeanC[2:NmaxCluster],cex.main=0.9,xlab="#
Cluster",ylab=paste(master,"-Loops"),panel.first=grid())} else {
          clus_graphs(clus_res$Gold,NmaxCluster,clu_dim)}
```

Appendix C: Data for comparing clustering methods

(see http://www.ima.umn.edu/~iwen/REU/REU_cluster.html#code)

Matlab code for generating random datasets

- An example `.m' file that creates a 2D dataset with 3 clusters. It can also be modified to generate other artificial data (with different numbers of clusters, dimensions, and underlying distributions).
- The following matlab package contains a file called "generate_samples.m" for generating hybrid linear models. It is part of the larger GPCA package. In order to avoid intersection of subspaces (so that standard clustering could be applied) one needs to set the parameter avoidIntersection = TRUE (and also have affine subspaces instead of linear).

Other data and data repositories

- Clustering datasets at UCI Repository
- Complete UCI Machine Learning Repository
- Yale Face Database B
- Some processed face datasets saved as Matlab data can be found here. Two matrices, X and Y, are included. If you plot Y(1:3,:) you will see three clearly separated clusters. The first 64 points are in one cluster, the next 64 points in another cluster, etc.. The original files are on the Yale Face Database B webpage (above). The folder names are yaleB5_P00, yaleB8_P00, yaleB10_P00. They have been processed following the steps described in Section 4.2.2 of the following paper. The matlab code used for processing them is here.
- Here is an example of spectral clustering data. It contains points from 2 noisy circles: after loading the `.mat' file type "plot(X(:,1),X(:,2),'LineStyle','.');" to see them. You can embed them into 2D space for clustering with EmbedCircles.m. Note that changing sigma in this file will lead to different problems.
- See also http://dbkgroup.org/handl/generators/

Appendix D: On determining variable importance for clustering

A plethora of methods has been proposed to select informative subsets of variables/features in the context of clustering analysis, as illustrated by recent literature on feature/variable selection (cf. Saeys et al., 2007, Steinley and Brusco, 2008b, Varshavsky et al. 2006, 2006).

Below we discuss three straightforward (univariate) methods which can be applied easily to express variable importance in a clustering context. In presenting the methods, we restrict ourselves to continuous variables.
We notice beforehand that the proposed techniques are univariate and consider each variable separately, thereby ignoring variable dependencies. This may lead to worse clustering performance when compared to other more advanced feature selection techniques (see e.g. Saeys et al. 2007).

A. ANOVA-based method (for complete cluster-partitioning)

This method is based on comparing what a specific variable/feature contributes to the within-cluster variability as compared to the between cluster variability. The resulting importance-index is expressed as the ratio BSS(j)/WSS(j) (see also Dudoit et al., 2002), defined by

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_{k=1}^{n}\sum_{i=1}^{N_k}(\bar{x}_{k.}(j) - \bar{x}_{..}(j))^2}{\sum_{k=1}^{n}\sum_{i=1}^{N_k}(x_{k,i}(j) - \bar{x}_{k.}(j))^2} \qquad (1)$$

where BSS refers to the between sums of squares variability and WSS to the within sum of squares variability. The ratio is used as an indication of the contribution of the variable j to the overall clustering.
Here *j* refers to the features/variables, *k* to the clusters, and *i* to the N_k objects within the *k*-th cluster. $x_{k,i}(j)$ refers to the value of the *j*-th variable (feature/component) of object *i* in cluster *k*; $\bar{x}_{..}(j)$ refers to the *j*-th component of the overall mean (population mean), while $\bar{x}_{k.}(j)$ refers to the *j*-th component of the cluster-mean of the *k*-th cluster.

Variables with the *highest* BSS(j)/WSS(j) are considered to have the largest 'explanatory performance' in respect to the 'unexplained one', and therefore are labeled as more important. See also the following textbox, which puts some caution in using these kind of indicators.

Remark: On the relation with ANOVA:

(a) Note that the total sums of squares can be written as the sum of the sums of squares of all variables/components, and be split into a within- and between-cluster part:

$$
\begin{aligned}
TSS = \sum_{j=1}^{p} TSS(j) &= \sum_{j=1}^{p}\sum_{k=1}^{n}\sum_{i=1}^{N_k}(x_{k,i}(j)-\bar{x}_{..}(j))^2 = \\
&= \sum_{j=1}^{p}\sum_{k=1}^{n}\sum_{i=1}^{N_k}(x_{k,i}(j)-\bar{x}_{k.}(j)+\bar{x}_{k.}(j)-\bar{x}_{..}(j))^2 \\
&= \sum_{j=1}^{p}\sum_{k=1}^{n}\sum_{i=1}^{N_k}(x_{k,i}(j)-\bar{x}_{k.}(j))^2+(\bar{x}_{k.}(j)-\bar{x}_{..}(j))^2 = \\
&= \sum_{j=1}^{p}\left[\sum_{k=1}^{n}\left(\sum_{i=1}^{N_k}(x_{k,i}(j)-\bar{x}_{k.}(j))^2+\sum_{i=1}^{N_k}(\bar{x}_{k.}(j)-\bar{x}_{..}(j))^2\right)\right]= \\
&= \sum_{j=1}^{p}\sum_{k=1}^{n}\left(WSS_k(j)+BSS_k(j)\right)= \\
&= \sum_{j=1}^{p}\left(WSS(j)+BSS(j)\right)
\end{aligned}
$$

where BSS(j) refers to the explained part and WSS(j) to the unexplained part of the sums of squares. The *k-means method* is intended to minimize the total within-sum of squares *WSS (=* Σ_j *WSS(j))* (unexplained) and thus in fact maximizes the in-between differences *BSS ((=* Σ_j *BSS(j))* (explained). This however does not imply that the various components *WSS (j)* are minimized individually (or, equivalently, the *BSS(j)* are maximized individually), since trade-offs between the various *WSS (j)* can be involved in minimizing their sum.

(b) The ratio *BSS(j)/WSS(j)* is in fact directly related to the *F*-ratio in the context of an *ANOVA* for the specific *j*-th variable $x_{k,i}(j)$. The F-ratio is $MSS^{between}(j)/MSS^{within}(j)$ where the various mean-sum of squares are defined as $MSS^{between}(j)=BSS(j)/(n-1)$ and

$MSS^{within}(j)=WSS(j)/(N-n)$ where $N=\sum_{k=1}^{n}N_k$.

The F-ratio test is applied to test whether the underlying cluster-means $\mu_{k.}(j)$ of $x_{k,i}(j)$ are all equal for *k=1, ..., n*, in which case F should be nearly equal to 1. Notice that BSS(j)/WSS(j)=(n-1)/(N-n) × F.

(c) One should however be careful to interpret this ratio completely in terms of *ANOVA*, since the underlying assumptions – concerning independence, normality and equal variance - for *ANOVA* are typically not valid in a clustering context where the clusters have been determined deliberately so as to minimize the within sum-of-squares (cf. Milligan and Mahajan (1980). Milligan and Cooper (1987)). Compare also Hartigan (1975) and Aldenderfer and Blashfield (1976) who illustrate the statistical inappropriateness of the use of (M)ANOVAs for indicating existence of clusters.

B. t-test based method (cluster-wise)

Another way to express the variable importance of the *j*-th variable in a specific cluster is by using the t-statistic, in fact checking to what extend the mean-value of the specific variable - when constrained to this cluster - differs from the overall mean-value. The corresponding importance index can be expressed as[29]:

[29] As implemented in the TwoStep cluster method in SPSS.

$$t_k(j) = \frac{\bar{x}_{k.}(j) - \bar{x}_{..}(j)}{\hat{s}_k(j)} \qquad (2)$$

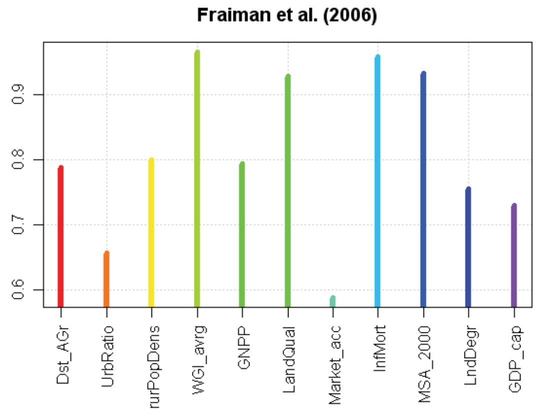
where $\hat{s}_k(j)$ is the standard deviation, defined as:

$$\hat{s}_k(j) = \sqrt{\frac{\sum_{i=1}^{N_k}(x_{k,i}(j) - \bar{x}_{k.}(j))^2}{(N_k - 1)}} \qquad (3)$$

The idea is that the importance of a variable for a cluster can be measured by the absolute value of this t-statistic, where variables with larger absolute t-statistics are considered as more important then variables for which the t-statistic is smaller. This measure is therefore initially related to a *specific* cluster (cluster-wise). A measure for the overall importance of the j-th variables for *all* clusters can e.g. be obtained by summing the absolute value $|t_k(j)|$ for all clusters k=1, …, n. Another possibility is to consider the maximum-value of the $|t_k(j)|$ over all clusters k=1, …, n., as a measure for the variable importance. See also Gat-Viks et al. (2003) who apply an ANOVA based test of equality of means amongst the cluster members.

C. 'Fraiman' index (for complete cluster-partitioning)

Fraiman et al. (2008) propose to 'blind' (subset of) variables, by fixing them at their mean-value, and to repeat the clustering analysis subsequently. Then the pairwise agreement (e.g. by means of the adjusted Rand index introduced by Hubert and Arabie (1985)) is determined between the partition thus obtained and the original partition with all variables fully included. This index serves as an indication for the importance of the blinded variable(s). The adjusted Rand index is a value between 0 and 1, where large values (near 1) mean that there is a large agreement between the partitions with and without blinding the specific variable. To identify the most important variables one therefore should look for variables with *small* Fraiman-indices.



Fraiman-measure to identify the importance of the different variables for the total cluster partition (low values indicate high importance).

Fraiman el al. 2008 show that this univariate procedure will falter if there are strong correlations between variables, since the effects of omitting one variable will be compensated by the other (non-blinded) related variable. This will typically result in a large agreement of the clustering partitions in the blinded and non-blinded case.

Therefore, in case of dependencies Fraiman et al. (2008) propose an alternative measure, where the blinded variable is not replaced by its marginal mean, but by its conditional mean over the set of other (non-blinded) variables.

Intermezzo: Promising alternatives

"Ensemble learning" methods that generate many classifiers and aggregate their results have been proposed during the last decade as efficient methods for analyzing the structure in data. Especially the procedure of *random forests (RF)*, which uses a multitude of regression trees on different bootstrap samples of the data (cf. Breiman (2001)) is a popular and user-friendly method. This method renders a measure for the variable importance of the involved (predictor) variables, and gives also a measure of the internal structure of the data (proximity of different data points to one another).
Although this method was first established for classification and regression problems (i.e. forms of supervised learning) the random-forest idea can also be applied for clustering purposes (unsupervised learning). The trick for this is to distinguish two datasets: the original dataset is called "class 1", while a synthetic dataset, using information on the marginal distributions of the original data, is constructed which is called "class 2". Next one uses the random-forest machinery to classify the combined data with a random forest. The underlying idea is that real data points that are similar to one another will tend to be classified in the same terminal node of the tree, as measured by the proximity matrix that can be returned using the RF-technique. Thus the proximity matrix can be taken as a similarity measure[30], which can be applied for dividing the original matrix into groups for visual exploration on basis of clustering or multi-dimensional scaling. See the example in Liaw and Wiener (2002) as a work-out how to perform this analysis with the randomForest package in R.
Along similar lines this method has been further applied and analysed by Horvath and Shi in a series of papers (Shi et al. 2005, 2006). They underline the attractiveness of the method since it enables handling mixed variable types, is invariant to monotonic transformation of the input variables and is robust to outlying observations. Moreover the RF-based dissimilarity easily deals with a large number of variables.

The above reframing of clustering in terms of random forest procedure offers a link to recent interesting literature (Strobl et al. 2007, 2008) on measuring the importance of variables in a random forest context explicitly accounting for the (conditional) effects of correlated variables. These results suggest ways to do this also for clustering, but this will not be worked out here. See also R-software like part(y)itioning (Hothorn et al. 2006) which can be applied in this context.

Another interesting related approach which deserves further exploration is offered by Questier et al. (2005), Smyth et al. (2006a) who put forward an extension of classification and regression trees, namely multivariate regression trees[31], for (supervised and unsupervised) feature selection as well as for cluster analysis. The idea is to use the original data (x) as explanatory variables (x) and also as response variables (y=x), giving rise to so-called Auto-Associative Multivariate Regression Trees. The suitability of this approach for clustering is further explored in Smyth et al. (2006b), while in Smyth et al. (2007) proposals are given to enhance the performance of the method by weighing the resulting cluster ensemble appropriately on basis of the prediction quality of the individual model. Also suggestions are

[30] Concerning this similarity measure provided by the random forest method, one should realize that ideally the choice of the (dis)similarity measure ideally should be determined by the kind of patterns one hopes to find, which makes that there are situations where other dissimilarities are preferable.
[31] R-software has been developed for multivariate regression trees, namely MVPART

given for determining the variable importance and the number of clusters. For R-software on multivariate regression trees see the CRAN package mvpart[32].

Appendix E: Commonly used internal validation indexes

In the sequel we present various ***internal validation indices*** (see also Günter, S, Bunke, H., 2003):

- *Silhouette index*: this composite index reflects the compactness and separation of clusters. A larger Silhouette index indicates a better overall quality of the clustering result (Kaufman & Rousseeuw, 1990).
  The Silhouette index (SI) calculates for each point a width depending on its membership in any cluster. This silhouette index is then the average of the silhouette widths of all points/objects:
  $$SI = \frac{1}{N}\sum_{i=1}^{N}\frac{(b_i - a_i)}{\max(a_i, b_i)}$$
  where b_i is the minimum of the average distances between the specific point *i* and the points in the other clusters, and a_i is the average distance between the point *i* and all other points in the cluster where *i* is member of. The values *s(i)=[b(i)-a(i)]/max[a(i),b(i)]* vary between -1 and 1, where values close to -1 mean that the point is on average closer to another cluster than the one it belongs to, in fact indicating that the object *i* is 'misclassified'. Values close to 1 mean that the average distance to its own cluster is significantly smaller than to any other cluster, indicating that object *i* is 'well classified'. When the width is near zero it is not clear whether the object should have been assigned to its current cluster or to the neighbouring cluster. The higher the silhouette index, the more compact and separated are the clusters. Kaufman and Rousseeuw, 1990, give guidance for the desirable size of the silhouette width; they consider a reasonable classification to be characterized by an average silhouette width above 0.5. Small silhouette width below 0.2 should be interpreted as a lack of substantial cluster structure.
- *Davies-Bouldin index*: This measure tries to maximize the between-cluster distance while minimizing the distance between the cluster centroid and the other points. It expresses the *average similarity* between each cluster and its most similar one. Small values correspond to clusters that are compact and have well-separated centres. Therefore its minimum value determines the optimal number of clusters.
- *Calinski-Harabasz index*: This index measures the between-cluster isolation and the within-cluster compactness, in terms of:
  $$CH(K) = \frac{Trace(S_B)}{K-1}\bigg/\frac{Trace(S_W)}{K-1}$$
  with N being the number of objects and S_B and S_W being the between and within-class scatter matrix
  $$S_W = \sum_{i=1}^{K}\sum_{j=1}^{N}\gamma_{ij}(x_j - m_j)(x_i - m_i) \quad ; \quad S_B = \sum_{i=1}^{K} N_i(m_i - m)(m_i - m)^T$$
  where $\Gamma=\{\gamma_{ij}\}$ is a partition matrix, with $\gamma_{ij}=1$ if x_j belongs to cluster *i* and 0 otherwise, where moreover $\sum_{i=1}^{K}\gamma_{ij} = 1$ for all *j*. $M=[m_1, \ldots, m_K]$ is the cluster prototype or centroid

[32] http://cran.nedmirror.nl/web/packages/mvpart/index.html

matrix, and $m_i = \frac{1}{N_i}\sum_{j=1}^{N} \gamma_{ij} x_j$ is the mean for the *i*-th cluster with N_i objects. The optimal number of clusters is determined by maximizing the CH-index.

- *Dunn index*: this index is defined as the ratio between the minimum distance between two clusters and the size of the largest cluster. Depending on the choice of the distance measure and the size of the cluster, various Dunn indices can be defined. Maximizing this index reflects to a certain extent the maximization of the inter-cluster-distances while simultaneously minimizing the intra-cluster distances.
- *RMSSTD index* (Root Mean Square Standard Deviation): This index is designed for hierarchical clustering, but can equally well be used for any clustering algorithm, and measures the homogeneity of the formed clusters (or the variance of clusters) at each step of the hierarchical clustering algorithm. A lower RMSSTD value indicates better clustering.
- *C index*: This index (Hubert and Schultz, 1976) is defined as follows:

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}}$$

where S is the sum of distances over all pairs of objects from the same cluster. Let *r* be the number of those pairs. Then $S_{\min}$ is the sum of the *r* smallest distances if all pairs of objects are considered (i.e. also objects that can belong to different clusters). Similarly $S_{\max}$ is the sum of the *r* largest distances out of all pairs. Hence a small value of C indicates a good clustering.

- *Maulik-Bandyopadhyay index*: This index is a combination of three terms

$$MB_k = \left(\frac{1}{k} \cdot \frac{E_1}{E_k} \cdot D_k\right)^p$$

where the intra-cluster distance is defined by $E_k = \sum_{i=1}^{k} \sum_{x \in c_i} \left\| x - z_i \right\|$ and the inter-cluster distance by $D_k = \max_{i,j=1}^{k} \left\| z_i - z_j \right\|$, where z_i is the centre of cluster c_i. *p* is chosen to be two and the number of clusters *k* is determined by maximizing MB_k.

- The *Cophenetic correlation coefficient (CPCC)* is an index to validate hierarchical clustering structures, and is based on the proximity matrix P={p_{ij}}, of the data X. It measures the degree of similarity between P and the cophenetic matrix Q={q_{ij}}, the elements of which express the proximity level where pairs of data points are grouped in the same cluster.

CPCC is defined as:

$$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} p_{ij} q_{ij} - \mu_P \mu_Q}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} p_{ij}^2 - \mu_P^2\right) \cdot \left(\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{ij}^2 - \mu_Q^2\right)}}$$

Where μ_P and μ_Q are the means of P and Q:

$$\mu_P = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} p_{ij} \quad ; \quad \mu_Q = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{ij}$$

with M=N(N-1)/2. The value of CPCC lies in the range of [-1,1] with an index value close to 1 indicating a significant similarity between P and Q. However for group average linkage (UPGMA) even large CPCC values (such as 0.9) cannot assure sufficient similarity between the two matrices.

Remark: Also for Fuzzy clustering internal validation indices have been proposed, such as the partition coefficient (PC) and partition entropy (PE), the (extended) Xie-Beni index and the Fukuyama-Sugeno index, c.f. Pal-Bezdek (1995), Hammah and Curran (2000), Wu and Yang, 2005; cf. section 10.4.3 in Xue and Wunsch (2008). Wang and Zhang (2007) performed an extensive evaluation of the fuzzy clustering indices, while Zhang et al. 2008 tested a newly proposed index. They conclude that cluster validation is a very difficult task and that ‘no matter how good your index is, there is a dataset out there waiting to trick it (and you)’ (Pal and Bezdek (1997)). Wu et al. (2009) recently analyse the robustness of the cluster indices for noise and outliers, and propose ways to robustify them.

PIK Report-Reference:

No. 1 3. Deutsche Klimatagung, Potsdam 11.-14. April 1994
Tagungsband der Vorträge und Poster (April 1994)

No. 2 Extremer Nordsommer '92
Meteorologische Ausprägung, Wirkungen auf naturnahe und vom Menschen beeinflußte Ökosysteme, gesellschaftliche Perzeption und situationsbezogene politisch-administrative bzw. individuelle Maßnahmen (Vol. 1 - Vol. 4)
H.-J. Schellnhuber, W. Enke, M. Flechsig (Mai 1994)

No. 3 Using Plant Functional Types in a Global Vegetation Model
W. Cramer (September 1994)

No. 4 Interannual variability of Central European climate parameters and their relation to the large-scale circulation
P. C. Werner (Oktober 1994)

No. 5 Coupling Global Models of Vegetation Structure and Ecosystem Processes - An Example from Arctic and Boreal Ecosystems
M. Plöchl, W. Cramer (Oktober 1994)

No. 6 The use of a European forest model in North America: A study of ecosystem response to climate gradients
H. Bugmann, A. Solomon (Mai 1995)

No. 7 A comparison of forest gap models: Model structure and behaviour
H. Bugmann, Y. Xiaodong, M. T. Sykes, Ph. Martin, M. Lindner, P. V. Desanker, S. G. Cumming (Mai 1995)

No. 8 Simulating forest dynamics in complex topography using gridded climatic data
H. Bugmann, A. Fischlin (Mai 1995)

No. 9 Application of two forest succession models at sites in Northeast Germany
P. Lasch, M. Lindner (Juni 1995)

No. 10 Application of a forest succession model to a continentality gradient through Central Europe
M. Lindner, P. Lasch, W. Cramer (Juni 1995)

No. 11 Possible Impacts of global warming on tundra and boreal forest ecosystems - Comparison of some biogeochemical models
M. Plöchl, W. Cramer (Juni 1995)

No. 12 Wirkung von Klimaveränderungen auf Waldökosysteme
P. Lasch, M. Lindner (August 1995)

No. 13 MOSES - Modellierung und Simulation ökologischer Systeme - Eine Sprachbeschreibung mit Anwendungsbeispielen
V. Wenzel, M. Kücken, M. Flechsig (Dezember 1995)

No. 14 TOYS - Materials to the Brandenburg biosphere model / GAIA
Part 1 - Simple models of the "Climate + Biosphere" system
Yu. Svirezhev (ed.), A. Block, W. v. Bloh, V. Brovkin, A. Ganopolski, V. Petoukhov, V. Razzhevaikin (Januar 1996)

No. 15 Änderung von Hochwassercharakteristiken im Zusammenhang mit Klimaänderungen - Stand der Forschung
A. Bronstert (April 1996)

No. 16 Entwicklung eines Instruments zur Unterstützung der klimapolitischen Entscheidungsfindung
M. Leimbach (Mai 1996)

No. 17 Hochwasser in Deutschland unter Aspekten globaler Veränderungen - Bericht über das DFG-Rundgespräch am 9. Oktober 1995 in Potsdam
A. Bronstert (ed.) (Juni 1996)

No. 18 Integrated modelling of hydrology and water quality in mesoscale watersheds
V. Krysanova, D.-I. Müller-Wohlfeil, A. Becker (Juli 1996)

No. 19 Identification of vulnerable subregions in the Elbe drainage basin under global change impact
V. Krysanova, D.-I. Müller-Wohlfeil, W. Cramer, A. Becker (Juli 1996)

No. 20 Simulation of soil moisture patterns using a topography-based model at different scales
D.-I. Müller-Wohlfeil, W. Lahmer, W. Cramer, V. Krysanova (Juli 1996)

No. 21 International relations and global climate change
D. Sprinz, U. Luterbacher (1st ed. July, 2n ed. December 1996)

No. 22 Modelling the possible impact of climate change on broad-scale vegetation structure - examples from Northern Europe
W. Cramer (August 1996)

No. 23 A methode to estimate the statistical security for cluster separation
F.-W. Gerstengarbe, P.C. Werner (Oktober 1996)

No. 24 Improving the behaviour of forest gap models along drought gradients
H. Bugmann, W. Cramer (Januar 1997)

No. 25 The development of climate scenarios
P.C. Werner, F.-W. Gerstengarbe (Januar 1997)

No. 26 On the Influence of Southern Hemisphere Winds on North Atlantic Deep Water Flow
S. Rahmstorf, M. H. England (Januar 1977)

No. 27 Integrated systems analysis at PIK: A brief epistemology
A. Bronstert, V. Brovkin, M. Krol, M. Lüdeke, G. Petschel-Held, Yu. Svirezhev, V. Wenzel (März 1997)

No. 28 Implementing carbon mitigation measures in the forestry sector - A review
M. Lindner (Mai 1997)

No. 29 Implementation of a Parallel Version of a Regional Climate Model
M. Kücken, U. Schättler (Oktober 1997)

No. 30 Comparing global models of terrestrial net primary productivity (NPP): Overview and key results
W. Cramer, D. W. Kicklighter, A. Bondeau, B. Moore III, G. Churkina, A. Ruimy, A. Schloss, participants of "Potsdam '95" (Oktober 1997)

No. 31 Comparing global models of terrestrial net primary productivity (NPP): Analysis of the seasonal behaviour of NPP, LAI, FPAR along climatic gradients across ecotones
A. Bondeau, J. Kaduk, D. W. Kicklighter, participants of "Potsdam '95" (Oktober 1997)

No. 32 Evaluation of the physiologically-based forest growth model FORSANA
R. Grote, M. Erhard, F. Suckow (November 1997)

No. 33 Modelling the Global Carbon Cycle for the Past and Future Evolution of the Earth System
S. Franck, K. Kossacki, Ch. Bounama (Dezember 1997)

No. 34 Simulation of the global bio-geophysical interactions during the Last Glacial Maximum
C. Kubatzki, M. Claussen (Januar 1998)

No. 35 CLIMBER-2: A climate system model of intermediate complexity. Part I: Model description and performance for present climate
V. Petoukhov, A. Ganopolski, V. Brovkin, M. Claussen, A. Eliseev, C. Kubatzki, S. Rahmstorf (Februar 1998)

No. 36 Geocybernetics: Controlling a rather complex dynamical system under uncertainty
H.-J. Schellnhuber, J. Kropp (Februar 1998)

No. 37 Untersuchung der Auswirkungen erhöhter atmosphärischer CO_2-Konzentrationen auf Weizenbestände des Free-Air Carbondioxid Enrichment (FACE) - Experimentes Maricopa (USA)
T. Kartschall, S. Grossman, P. Michaelis, F. Wechsung, J. Gräfe, K. Waloszczyk, G. Wechsung, E. Blum, M. Blum (Februar 1998)

No. 38 Die Berücksichtigung natürlicher Störungen in der Vegetationsdynamik verschiedener Klimagebiete
K. Thonicke (Februar 1998)

No. 39 Decadal Variability of the Thermohaline Ocean Circulation
S. Rahmstorf (März 1998)

No. 40 SANA-Project results and PIK contributions
K. Bellmann, M. Erhard, M. Flechsig, R. Grote, F. Suckow (März 1998)

No. 41 Umwelt und Sicherheit: Die Rolle von Umweltschwellenwerten in der empirisch-quantitativen Modellierung
D. F. Sprinz (März 1998)

No. 42 Reversing Course: Germany's Response to the Challenge of Transboundary Air Pollution
D. F. Sprinz, A. Wahl (März 1998)

No. 43 Modellierung des Wasser- und Stofftransportes in großen Einzugsgebieten. Zusammenstellung der Beiträge des Workshops am 15. Dezember 1997 in Potsdam
A. Bronstert, V. Krysanova, A. Schröder, A. Becker, H.-R. Bork (eds.) (April 1998)

No. 44 Capabilities and Limitations of Physically Based Hydrological Modelling on the Hillslope Scale
A. Bronstert (April 1998)

No. 45 Sensitivity Analysis of a Forest Gap Model Concerning Current and Future Climate Variability
P. Lasch, F. Suckow, G. Bürger, M. Lindner (Juli 1998)

No. 46 Wirkung von Klimaveränderungen in mitteleuropäischen Wirtschaftswäldern
M. Lindner (Juli 1998)

No. 47 SPRINT-S: A Parallelization Tool for Experiments with Simulation Models
M. Flechsig (Juli 1998)

No. 48 The Odra/Oder Flood in Summer 1997: Proceedings of the European Expert Meeting in Potsdam, 18 May 1998
A. Bronstert, A. Ghazi, J. Hladny, Z. Kundzewicz, L. Menzel (eds.) (September 1998)

No. 49 Struktur, Aufbau und statistische Programmbibliothek der meteorologischen Datenbank am Potsdam-Institut für Klimafolgenforschung
H. Österle, J. Glauer, M. Denhard (Januar 1999)

No. 50 The complete non-hierarchical cluster analysis
F.-W. Gerstengarbe, P. C. Werner (Januar 1999)

No. 51 Struktur der Amplitudengleichung des Klimas
A. Hauschild (April 1999)

No. 52 Measuring the Effectiveness of International Environmental Regimes
C. Helm, D. F. Sprinz (Mai 1999)

No. 53 Untersuchung der Auswirkungen erhöhter atmosphärischer CO_2-Konzentrationen innerhalb des Free-Air Carbon Dioxide Enrichment-Experimentes: Ableitung allgemeiner Modellösungen
T. Kartschall, J. Gräfe, P. Michaelis, K. Waloszczyk, S. Grossman-Clarke (Juni 1999)

No. 54 Flächenhafte Modellierung der Evapotranspiration mit TRAIN
L. Menzel (August 1999)

No. 55 Dry atmosphere asymptotics
N. Botta, R. Klein, A. Almgren (September 1999)

No. 56 Wachstum von Kiefern-Ökosystemen in Abhängigkeit von Klima und Stoffeintrag - Eine regionale Fallstudie auf Landschaftsebene
M. Erhard (Dezember 1999)

No. 57 Response of a River Catchment to Climatic Change: Application of Expanded Downscaling to Northern Germany
D.-I. Müller-Wohlfeil, G. Bürger, W. Lahmer (Januar 2000)

No. 58 Der "Index of Sustainable Economic Welfare" und die Neuen Bundesländer in der Übergangsphase
V. Wenzel, N. Herrmann (Februar 2000)

No. 59 Weather Impacts on Natural, Social and Economic Systems (WISE, ENV4-CT97-0448) German report
M. Flechsig, K. Gerlinger, N. Herrmann, R. J. T. Klein, M. Schneider, H. Sterr, H.-J. Schellnhuber (Mai 2000)

No. 60 The Need for De-Aliasing in a Chebyshev Pseudo-Spectral Method
M. Uhlmann (Juni 2000)

No. 61 National and Regional Climate Change Impact Assessments in the Forestry Sector - Workshop Summary and Abstracts of Oral and Poster Presentations
M. Lindner (ed.) (Juli 2000)

No. 62 Bewertung ausgewählter Waldfunktionen unter Klimaänderung in Brandenburg
A. Wenzel (August 2000)

No. 63 Eine Methode zur Validierung von Klimamodellen für die Klimawirkungsforschung hinsichtlich der Wiedergabe extremer Ereignisse
U. Böhm (September 2000)

No. 64 Die Wirkung von erhöhten atmosphärischen CO_2-Konzentrationen auf die Transpiration eines Weizenbestandes unter Berücksichtigung von Wasser- und Stickstofflimitierung
S. Grossman-Clarke (September 2000)

No. 65 European Conference on Advances in Flood Research, Proceedings, (Vol. 1 - Vol. 2)
A. Bronstert, Ch. Bismuth, L. Menzel (eds.) (November 2000)

No. 66 The Rising Tide of Green Unilateralism in World Trade Law - Options for Reconciling the Emerging North-South Conflict
F. Biermann (Dezember 2000)

No. 67 Coupling Distributed Fortran Applications Using C++ Wrappers and the CORBA Sequence Type
T. Slawig (Dezember 2000)

No. 68 A Parallel Algorithm for the Discrete Orthogonal Wavelet Transform
M. Uhlmann (Dezember 2000)

No. 69 SWIM (Soil and Water Integrated Model), User Manual
V. Krysanova, F. Wechsung, J. Arnold, R. Srinivasan, J. Williams (Dezember 2000)

No. 70 Stakeholder Successes in Global Environmental Management, Report of Workshop, Potsdam, 8 December 2000
M. Welp (ed.) (April 2001)

No. 71 GIS-gestützte Analyse globaler Muster anthropogener Waldschädigung - Eine sektorale Anwendung des Syndromkonzepts
M. Cassel-Gintz (Juni 2001)

No. 72 Wavelets Based on Legendre Polynomials
J. Fröhlich, M. Uhlmann (Juli 2001)

No. 73 Der Einfluß der Landnutzung auf Verdunstung und Grundwasserneubildung - Modellierungen und Folgerungen für das Einzugsgebiet des Glan
D. Reichert (Juli 2001)

No. 74 Weltumweltpolitik - Global Change als Herausforderung für die deutsche Politikwissenschaft
F. Biermann, K. Dingwerth (Dezember 2001)

No. 75 Angewandte Statistik - PIK-Weiterbildungsseminar 2000/2001
F.-W. Gerstengarbe (Hrsg.) (März 2002)

No. 76 Zur Klimatologie der Station Jena
B. Orlowsky (September 2002)

No. 77 Large-Scale Hydrological Modelling in the Semi-Arid North-East of Brazil
A. Güntner (September 2002)

No. 78 Phenology in Germany in the 20th Century: Methods, Analyses and Models
J. Schaber (November 2002)

No. 79 Modelling of Global Vegetation Diversity Pattern
I. Venevskaia, S. Venevsky (Dezember 2002)

No. 80 Proceedings of the 2001 Berlin Conference on the Human Dimensions of Global Environmental Change "Global Environmental Change and the Nation State"
F. Biermann, R. Brohm, K. Dingwerth (eds.) (Dezember 2002)

No. 81 POTSDAM - A Set of Atmosphere Statistical-Dynamical Models: Theoretical Background
V. Petoukhov, A. Ganopolski, M. Claussen (März 2003)

No. 82 Simulation der Siedlungsflächenentwicklung als Teil des Globalen Wandels und ihr Einfluß auf den Wasserhaushalt im Großraum Berlin
B. Ströbl, V. Wenzel, B. Pfützner (April 2003)

No. 83 Studie zur klimatischen Entwicklung im Land Brandenburg bis 2055 und deren Auswirkungen auf den Wasserhaushalt, die Forst- und Landwirtschaft sowie die Ableitung erster Perspektiven
F.-W. Gerstengarbe, F. Badeck, F. Hattermann, V. Krysanova, W. Lahmer, P. Lasch, M. Stock, F. Suckow, F. Wechsung, P. C. Werner (Juni 2003)

No. 84 Well Balanced Finite Volume Methods for Nearly Hydrostatic Flows
N. Botta, R. Klein, S. Langenberg, S. Lützenkirchen (August 2003)

No. 85 Orts- und zeitdiskrete Ermittlung der Sickerwassermenge im Land Brandenburg auf der Basis flächendeckender Wasserhaushaltsberechnungen
W. Lahmer, B. Pfützner (September 2003)

No. 86 A Note on Domains of Discourse - Logical Know-How for Integrated Environmental Modelling, Version of October 15, 2003
C. C. Jaeger (Oktober 2003)

No. 87 Hochwasserrisiko im mittleren Neckarraum - Charakterisierung unter Berücksichtigung regionaler Klimaszenarien sowie dessen Wahrnehmung durch befragte Anwohner
M. Wolff (Dezember 2003)

No. 88 Abflußentwicklung in Teileinzugsgebieten des Rheins - Simulationen für den Ist-Zustand und für Klimaszenarien
D. Schwandt (April 2004)

No. 89 Regionale Integrierte Modellierung der Auswirkungen von Klimaänderungen am Beispiel des semi-ariden Nordostens von Brasilien
A. Jaeger (April 2004)

No. 90 Lebensstile und globaler Energieverbrauch - Analyse und Strategieansätze zu einer nachhaltigen Energiestruktur
F. Reusswig, K. Gerlinger, O. Edenhofer (Juli 2004)

No. 91 Conceptual Frameworks of Adaptation to Climate Change and their Applicability to Human Health
H.-M. Füssel, R. J. T. Klein (August 2004)

No. 92 Double Impact - The Climate Blockbuster 'The Day After Tomorrow' and its Impact on the German Cinema Public
F. Reusswig, J. Schwarzkopf, P. Polenz (Oktober 2004)

No. 93 How Much Warming are we Committed to and How Much Can be Avoided?
B. Hare, M. Meinshausen (Oktober 2004)

No. 94 Urbanised Territories as a Specific Component of the Global Carbon Cycle
A. Svirejeva-Hopkins, H.-J. Schellnhuber (Januar 2005)

No. 95 GLOWA-Elbe I - Integrierte Analyse der Auswirkungen des globalen Wandels auf Wasser, Umwelt und Gesellschaft im Elbegebiet
F. Wechsung, A. Becker, P. Gräfe (Hrsg.) (April 2005)

No. 96 The Time Scales of the Climate-Economy Feedback and the Climatic Cost of Growth
S. Hallegatte (April 2005)

No. 97 A New Projection Method for the Zero Froude Number Shallow Water Equations
S. Vater (Juni 2005)

No. 98 Table of EMICs - Earth System Models of Intermediate Complexity
M. Claussen (ed.) (Juli 2005)

No. 99 KLARA - Klimawandel - Auswirkungen, Risiken, Anpassung
M. Stock (Hrsg.) (Juli 2005)

No. 100 Katalog der Großwetterlagen Europas (1881-2004) nach Paul Hess und Helmut Brezowsky
6., verbesserte und ergänzte Auflage
F.-W. Gerstengarbe, P. C. Werner (September 2005)

No. 101 An Asymptotic, Nonlinear Model for Anisotropic, Large-Scale Flows in the Tropics
S. Dolaptchiev (September 2005)

No. 102 A Long-Term Model of the German Economy: lagom$^{d\_sim}$
C. C. Jaeger (Oktober 2005)

No. 103 Structuring Distributed Relation-Based Computations with SCDRC
N. Botta, C. Ionescu, C. Linstead, R. Klein (Oktober 2006)

No. 104 Development of Functional Irrigation Types for Improved Global Crop Modelling
J. Rohwer, D. Gerten, W. Lucht (März 2007)

No. 105 Intra-Regional Migration in Formerly Industrialised Regions: Qualitative Modelling of Household Location Decisions as an Input to Policy and Plan Making in Leipzig/Germany and Wirral/Liverpool/UK
D. Reckien (April 2007)

No. 106 Perspektiven der Klimaänderung bis 2050 für den Weinbau in Deutschland (Klima 2050) - Schlußbericht zum FDW-Vorhaben: Klima 2050
M. Stock, F. Badeck, F.-W. Gerstengarbe, D. Hoppmann, T. Kartschall, H. Österle, P. C. Werner, M. Wodinski (Juni 2007)

No. 107 Climate Policy in the Coming Phases of the Kyoto Process: Targets, Instruments, and the Role of Cap and Trade Schemes - Proceedings of the International Symposium, February 20-21, 2006, Brussels
M. Welp, L. Wicke, C. C. Jaeger (eds.) (Juli 2007)

No. 108 Correlation Analysis of Climate Variables and Wheat Yield Data on Various Aggregation Levels in Germany and the EU-15 Using GIS and Statistical Methods, with a Focus on Heat Wave Years
T. Sterzel (Juli 2007)

No. 109 MOLOCH - Ein Strömungsverfahren für inkompressible Strömungen - Technische Referenz 1.0
M. Münch (Januar 2008)

No. 110 Rationing & Bayesian Expectations with Application to the Labour Market
H. Förster (Februar 2008)

No. 111 Finding a Pareto-Optimal Solution for Multi-Region Models Subject to Capital Trade and Spillover Externalities
M. Leimbach, K. Eisenack (November 2008)

No. 112 Die Ertragsfähigkeit ostdeutscher Ackerflächen unter Klimawandel
F. Wechsung, F.-W. Gerstengarbe, P. Lasch, A. Lüttger (Hrsg.) (Dezember 2008)

No. 113 Klimawandel und Kulturlandschaft Berlin
H. Lotze-Campen, L. Claussen, A. Dosch, S. Noleppa, J. Rock, J. Schuler, G. Uckert (Juni 2009)

No. 114 Die landwirtschaftliche Bewässerung in Ostdeutschland seit 1949 - Eine historische Analyse vor dem Hintergrund des Klimawandels
M. Simon (September 2009)

No. 115 Continents under Climate Change - Conference on the Occasion of the 200th Anniversary of the Humboldt-Universität zu Berlin, Abstracts of Lectures and Posters of the Conference,
April 21-23, 2010, Berlin
W. Endlicher, F.-W. Gerstengarbe (eds.) (April 2010)

No. 116 Nach Kopenhagen: Neue Strategie zur Realisierung des 2°max-Klimazieles
L. Wicke, H. J. Schellnhuber, D. Klingenfeld (April 2010)

No. 117 Evaluating Global Climate Policy - Taking Stock and Charting a New Way Forward
D. Klingenfeld (April 2010)

No. 118 Untersuchungen zu anthropogenen Beeinträchtigungen der Wasserstände am Pegel Magdeburg-Strombrücke
M. Simon (September 2010)

No. 119 Katalog der Großwetterlagen Europas (1881-2009) nach Paul Hess und Helmut Brezowsky
7., verbesserte und ergänzte Auflage
P. C. Werner, F.-W. Gerstengarbe (Oktober 2010)

No. 120 Energy taxes, resource taxes and quantity rationing for climate protection
K. Eisenack, O. Edenhofer, M. Kalkuhl (November 2010)

No. 121 Klimawandel in der Region Havelland-Fläming
A. Lüttger, F.-W. Gerstengarbe, M. Gutsch, F. Hattermann, P. Lasch, A. Murawski,
J. Petraschek, F. Suckow, P. C. Werner (Januar 2011)

No. 122 Adaptation to Climate Change in the Transport Sector: A Review
K. Eisenack, R. Stecker, D. Reckien, E. Hoffmann (Mai 2011)

No. 123 Spatial-temporal changes of meteorological parameters in selected circulation patterns
P. C. Werner, F.-W. Gerstengarbe (November 2011)

No. 124 Assessment of Trade-off Decisions for Sustainable Bioenergy Development in the Philippines: An Application of Conjoint Analysis
L. A. Acosta, D. B. Magcale-Macandog, W. Lucht, K. G. Engay, M. N. Q. Herrera,
O. B. S. Nicopior, M. I. V. Sumilang, V. Espaldon (November 2011)

No. 125 Historisch vereinbarte minimale mittlere Monatsabflüsse der Elbe im tschechisch-deutschen Grenzprofil bei Hřensko/Schöna – Eine Analyse der Niedrigwasseraufhöhung im Grenzprofil infolge des Talsperrenbaus im tschechischen Einzugsgebiet der Elbe
M. Simon, J. Böhme (März 2012)

No. 126 Cluster Analysis to Understand Socio-Ecological Systems: A Guideline
P. Janssen, C. Walther, M. Lüdeke (September 2012)