MAGISTERARBEIT

Integrating Statistical Basefunctionality in Interactive Visual Data Analysis

Ausgeführt am VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH

unter der Leitung von
Priv.-Doz. Dipl.-Ing. Dr.techn. Helwig Hauser,
und der Mitbetreuung von
Ao.Univ.-Prov. Dipl.-Ing. Dr.techn. Peter Filzmoser und
Dipl.-Ing. Harald Piringer,

eingereicht
an der Technischen Universität Wien,
Fakultät für Informatik

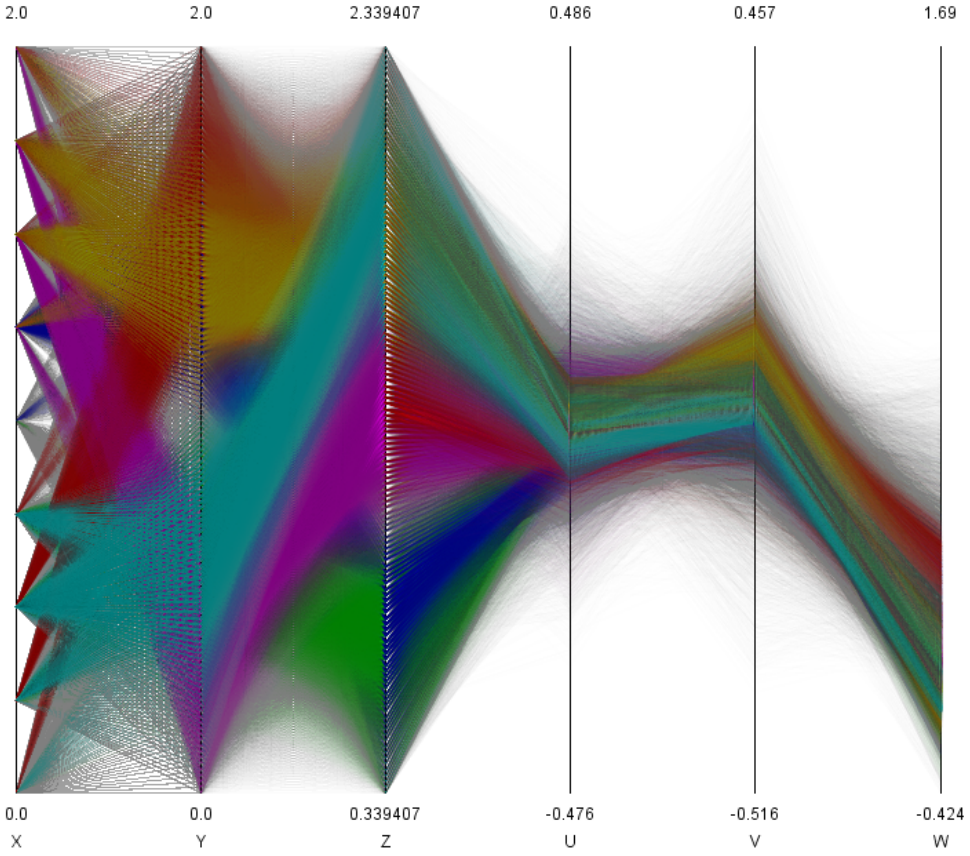
von
Jürgen Platzer
2632 Wimpassing, Penk 78
Matrikelnummer: 0025360

Wien, März 2007 Jürgen Platzer

Jürgen Platzer

Integrating Statistical Basefunctionality in Interactive Visual Data Analysis

MASTER'S THESIS



http://www.VRVis.at/vis/resources/DA-JPlatzer/

Juergen.Platzer@VRVis.at

Abstract, Kurzfassung

(engl.) Both information visualization and statistics analyse high dimensional data, but these sciences provide different ways to explore datasets. The information visualization is a branch of the field of computer graphics and creates graphics of the datasets that in general contain more than three dimensions to provide insight to the behaviour of the data. Because of the high dimensionality the data items usually do not show any inherent spatial reference, which poses a special challenge to visualize the entire data. Additionally interaction possibilities are provided to adapt the graphics to the needs of the user. This allows the visual exploration and the extraction of the intrinsic information of the data.

In contrast to that statistics execute algorithms that provide numerical summaries of the analysis of the datasets. Based on the knowledgeable theory of data exploration the results of those methods allow making statements about the datasets and provide a hint for their validity.

As both sciences pursue the same aims, it is a consistent consequence to combine methods of information visualization and statistics to achieve a more efficient exploration of multivariate data, which is also called data mining. Therefore this work surveys the most important tools provided by both disciplines to analyse high dimensional data. Furthermore existing applications using techniques of the field of statistics and of the information visualization are presented.

But the main contribution of this work is to provide statistical methods for visual data mining applications. Therefore a library was compiled that contains routines, which are of high importance for information visualization techniques and allow a fast modification of their results, to integrate possible adaptations in the visualization. The library is able to work on datasets containing millions of data items and hundreds of dimensions.

In addition an example application is introduced that demonstrates a possible interweaving between statistical methods and information visualization techniques. Tasks like the detection of outliers, the grouping of data items and attributes as well as the reduction of the dimensionality were incorporated.

(deut.) Sowohl die Informationsvisualisierung als auch die Statistik beschäftigen sich mit der Analyse von hochdimensionalen Daten, wobei beide Wissenschaften unterschiedliche Wege beschreiten. Die Informationsvisualisierung ist ein Teilgebiet der Computergraphik und erstellt aus Datensätzen, die üblicherweise mehr als drei Dimensionen aufweisen, Grafiken, die Einsicht in das Wesen der Daten geben sollen. Aufgrund der hohen Dimensionalität weisen die Datenpunkte oft keinen inhärenten räumlichen Bezug auf, weshalb die besondere Herausforderung in der Darstellung der Gesamtheit der Daten liegt. Zusätzlich werden Interaktionsmöglichkeiten zur Verfügung gestellt, um die Grafiken an die Bedürfnisse des Benutzers anzupassen. Somit ist es möglich, die Daten visuell zu erforschen und die wesentlichen Informationen zu extrahieren.
Im Gegensatz dazu bedient sich die Statistik der Ausführung von Algorithmen, die numerische Zusammenfassungen des zu untersuchenden Verhaltens der Daten erstellen. Basierend auf den fundierten theoretischen Betrachtungen der Datenanalyse erlauben diese Ergebnisse, Aussagen über die analysierten Datensätze zu treffen und zusätzlich festzustellen, mit welcher Wahrscheinlichkeit diese Aussagen Gültigkeit besitzen.
Da beide Wissenschaften die selben Ziele verfolgen, ist es eine logische Konsequenz Methoden der Statistik mit den Techniken der Informationsvisualisierung zu kombinieren, um bessere und effizientere Analysen der Daten vornehmen zu können. Diese Arbeit gibt daher einen Überblick über die wichtigsten Werkzeuge, welche von der Statistik und der Informationsvisualisierung für die Exploration von hochdimensionalen Daten bereitgestellt wird. Außerdem werden bereits existierende Anwendungen, die Techniken aus beiden Disziplinen vereinen, vorgestellt.
Das primäre Ziel dieser Arbeit ist es aber, statistische Methoden für Applikationen der Informationsvisualisierung zur Verfügung zu stellen. Dafür wurde eine Bibliothek an Routinen zusammengestellt, die zum einen als besonders wichtig für die visuelle Datenexploration gelten und zum anderen eine Modifikation ihrer Parameter und eine rasche Neuberechnung zulassen, so dass Änderungen für die Visualisierung übernommen werden können. Diese Bibliothek ist darauf ausgerichtet Datensätze, die Millionen von Datenpunkten und Hunderte von Dimensionen enthalten, zu bearbeiten. Zusätzlich wird in einer Beispielapplikation eine mögliche Verflechtung zwischen statistischen Routinen und verschiedenen Visualisierungsformen demonstriert. Hierbei wurde besonderes Augenmerk auf die Erkennung von Ausreißern, das Gruppieren von Datenpunkten und Dimensionen sowie die Dimensionsreduktion gelegt.

Table of Contents

Chapter 1

Introduction

In the second half of the past century the capabilities of computers grew tremendously and according to this development the size of the datasets that could be handled increased significantly. Nowadays it is commonplace to work with millions of data items that are defined in thousands of dimensions. Datasets of this scale are gathered in digital libraries, during simulations, for the analysis of genetic data or by surveys. Consequently the exploration of the given mass of data items and the drawing of conclusions is a crucial working area, which will furthermore gain importance.

The next sections discuss common tasks in a data exploration process and outline approaches from the fields statistics and information visualization that allow the user to accomplish them. Furthermore a collaboration between those sciences is encouraged and an approach how this can be fulfilled is outlined.

1.1 Common Data Exploration Tasks

An example posing challenges for the exploration and the filtering of the main information of the data is the examination of unstructured text documents. As the number of documents in digital libraries as well as in the world wide web is increasing rapidly their analysis and categorization gained importance in recent years. Common tasks are the detection of spam emails or the classification of documents according to their relevance for a given search key. To accomplish this a single text is stored as a vector of word counts. Because ten thousands of vocables are considered such a dataset contains a large number of attributes.

An initial analysis that can be accomplished on such a dataset is the detection of the main categories of documents and their properties. Categories of documents could be fairy tales, scientific publications as well as newspaper articles. Their features may be exceptional frequencies of uncommon vocables or word combinations as well as a significant pattern of

word counts. A similar task is the identification of documents that are displaced, because their content does not match the topic of a digital library, or that are exceptional with respect to a user defined property.

Of course categorizations can not be fulfilled by examining the dataset itself in a text representation. Thus techniques have to be applied that allow the extraction of the needed information from the data. Both the field of statistics as well as the information visualization provide functionality to deal with the complex process of exploring multivariate data, which is also called data mining.

The statistical analysis of data is applied for centuries. Thus a huge variety of methods and a profound theoretic foundation for the data exploration has been introduced. But foremost with the rising capabilities of computers the creation of statistical routines that analyse large datasets was initiated. These algorithms provide numeric summaries that capture the information of interest. Examples are statistical estimators specifying location or spread of data as well as models for predicting values of variables.

For the analysis of documents routines can be applied that find the major groups in the data by focussing on objects showing the same patterns. These procedures are called clusterings and provide a summary of the trends in the data by calculating the centers of the detected groups holding the average behaviour of the data items of a cluster. Thus the user can identify the most important word count patterns in the dataset and can assign the documents captured by one cluster to a certain text category.

Information visualization techniques try to achieve this result by applying other means. As the visualization is a branch of computer graphics images and animations based on the properties of the data are created. The key element is that the extraordinary pattern recognition capabilities of the human visual system are used to identify outlying values, groups in the data or other features of special interest. Because the field of information visualization analyses multivariate datasets, there is usually no inherent reference to the three dimensional space. Thus it is a tremendous challenge to capture the high dimensional information in a 2D illustration. A variety of techniques has been developed that focus on different aspects of the data to accomplish this task. But the created graphics can also be interactively modified to the needs of the user. Zooming and selection techniques allow focusing on a subset of data items and thus the inspection of their behaviour.

Thus a selection approach can be applied to interactively browse the properties of documents in a digital library. By defining constraints that a text must fulfil, for example certain word count values, subgroups in the data can be identified and explored.

But as both sciences have their strengths, each technique has also disadvantages to overcome. Visualization approaches clearly suffer that the user is not able to interpret high dimensional features, because humans are used to think in three dimensional spaces. Furthermore

restrictions in the screen space do not allow showing all data items or all dimensions of a dataset without cluttering. In contrast to that statistical routines in general perform well if the data applies to a theoretic distribution like the multivariate normal distribution. Of course this constraint is not fulfilled by the datasets that should be analysed. Another issue is that algorithms like a chosen cluster procedure may not be adequate for the structure of the data. But it is not trivial to find out, whether this is the case, and which alternative algorithm could be applied.

These observations can be easily demonstrated by the task of exploring text documents. The high dimensionality makes a visual examination of these datasets containing up to a million of documents difficult, because the presentation of all dimensions exceeds the visualization space. Even the illustration of the most important dimensions does not allow an efficient pattern recognition and the pruned dimensions may hold essential information. The numerical inspection is also limited due to the curse of dimensionality, which leads in this case to a sparse data space. The large number of vocables includes words that are only used by a small subset of documents. This circumstance causes data mining algorithms to fail, because large groups of dimensions hold the same information for the majority of the data items.

As these examples show, a combination of different approaches could be useful to overcome the disadvantages of the applied techniques. This concept is also encouraged by the fact that both sciences show similar aims but use different ways to reach them. Statistical routines use the possibilities of today's computers to process a high number of calculations. The results are presented in numerical outputs that the user has to interpret to obtain new knowledge. The visualization uses the fast computers to create interactive graphic interfaces that can be adapted to the user's needs within fractions of a second. This procedure allows new insights into the data by identifying patterns visually. Consequently if one approach fails, the other can compensate this failure, because it relies on a different system.

But although statistics and visualization developed techniques to identify groups or outlying values in high dimensional data, only hesitantly approaches were made, where the strengths of these fields are combined to achieve more efficient data mining applications. This work is intended to contribute to the development of tools connecting the strong theoretic fundamentals and efficient calculation of numerical facts from the field of statistics with the capabilities of visual representation and the interactive nature of information visualization.

1.2 Integrating Statistical Functionality into Information Visualization

One way to achieve an efficient combination of statistical routines and visualization techniques is to compile and adapt the statistical functionality in a library so that information visualization approaches can call those routines according to their needs. For this purpose special attention has to be paid on the incorporation of so called hooks of interaction, which allow immediate updates of numerical summaries that are used again for the visualizations. Thus the interaction techniques applied in the views of a visual data mining application have to be translated to common function calls of statistical routines. This is a necessity to allow the interactive collaboration of statistical functionality and user interaction, which is crucial to exploit the previously outlined compensation of possible failures of one technique.

To accomplish this approach of integration the statistical procedures that should be considered for such a library have to be determined. As the field of statistics provides a large range of methods to analyse data, research has been made to evaluate which functionality is most relevant for information visualization applications. Hence popular visual data exploration tools like SpotFire [7] were tested with respect to the provided statistical routines. Furthermore the recent publications of the field of information visualization that focus on statistical calculations were considered to create a list of useful methods for a visual data mining tool.

In the scope of this work the tasks clustering and outlier detection were focused, because the detection of the main trends in the data and the segregation of objects that obviously are not part of the data itself are central working steps in a data mining application. Furthermore for these operations the benefit of a collaboration of both sciences is tremendous. Because the work with high dimensional data poses challenges for visualization techniques as well as for statistical routines a dimension reduction technique is also a central functionality of the created library. As those statistical routines need an adequate data preparation, a set of transformations is implemented, that maps the data values to certain intervals or distributions. Besides these tasks standard calculations like statistical moments, correlation and distance measures as well as hypothesis tests were realized.

To give an explanation of the usefulness of this statistical functionality within a visual data mining application several examples are given, that suggest possible integration scenarios and their advantages in comparison to the separated use of the techniques. But also a sample application that picks up some of these proposed ideas was implemented and demonstrates how the statistical routines interweave efficiently with visualization approaches. This system especially features on interactive visual outlier detection, feature subset selection and clustering.

Although the techniques of both sciences are enhanced to provide functionality for collaboration, this work is not intended to improve statistical routines by modifications or to intro-

duce new methods that are especially suitable for visual exploration. The same applies to the techniques of information visualization. The work shows how the combination of both fields can help to overcome the drawbacks of single procedures. It is not denied, that both fields are introducing new developments that compensate the shortcomings of techniques, by the means of their science. But nevertheless an interactive collaboration between a statistical and a visualization approach may solve those problems efficiently.

1.3 Organization of this work

To provide an overview about the main techniques of information visualization and statistical routines, the next section outlines the major contributions of these fields for data exploration. This state of the art report concentrates mainly on clustering, outlier detection and dimension reduction. For those tasks also existing applications are introduced, that focus on the collaboration between statistics and visualization.

In section 3 the most important statistical definitions and algorithms, which are compiled in the statistics library are stated.

The strengths and weaknesses of statistical routines and information visualization techniques are analysed as well as an outline of the benefits of possible collaborations between both fields are presented in section 4. To accomplish the latter also specific examples based on existing functionality are depicted.

The statistical library containing the base functionality for information visualization applications is introduced in section 5. There an explanation is given why the implemented routines were chosen. Furthermore the functionality of the sample application is outlined. Proof of concept cases, where this application demonstrates the work of the integrated statistical routines in common visualizations on real datasets, are discussed in section 6.

Attention on issues of the implementation of the library as well as on the work with large datasets is paid in section 7. Also the use of parameters in the corresponding function calls and the runtime of central routines are stated.

In chapter 8 the work is summarized and the main contributions are accentuated. Finally the work is concluded in section 9, where necessary developments for the successful integration of statistical functionality in information visualization applications are addressed.

Chapter 2

State of the Art

Both areas - Visualization and Statistics - provide different ways to analyse data. In this section short overviews about the routines and techniques of both fields are given. Afterwards an extensive discussion of the state of the art concerning the combination of information visualization and statistical data exploration algorithms is presented.

2.1 Visualization Techniques

In the last 20 years visualization was a tremendously growing research field, where a huge variety of methods for presentation and visual exploration of datasets was developed. According to the increasing number of operations that computers can execute in short time the possibilities of visualizations grew and the interaction between humans and the visual representation gained focus. Additionally the explosion of the data size posed the challenge of providing different views that are linked together as well as allowing interactive modifications that give feedback in a few seconds, while handling millions of data items. This short overview introduces the most popular techniques for the visual data exploration of high dimensional data. The discussed visualization types focus on different aspects of datasets and thus are often used in combination to provide different insights in the processed data. To give a clear discussion of the various techniques a subdivision of the approaches into four categories according to [55] is made.

2.1.1 Geometric Projection Techniques

Geometric projection techniques map dimension values on screen space positions. Primitives, like points or polylines that represent data items are drawn so that they apply to the mapped attribute values of the objects.

An intuitive way to visualize data are two-dimensional plots, where the values of two attributes are mapped on the x and on the y axis respectively and the data items are represented as points in the area spanned by the two perpendicular axes. This visualization is called scatterplot and has the advantage, that structures in the shown dimension pair can be detected very fast, because the human is used to think in spaces with Cartesian coordinates. The same applies to three dimensional scatterplots [92], where interaction techniques have to be integrated to allow the user the navigation through the 3D space. Without interaction misleading conclusions can be drawn caused by occlusions or effects of the perspective projection introduced to depict the three dimensional space. While patterns like correlations, outlying data items or groups can be easily detected, the main problems of this technique are the overplotting, meaning that the user can not identify how many objects are depicted at a plotting coordinate, and the representation of only two variables and thus only two dimensional patterns.

To provide insight in a high dimensional dataset the scatterplot-matrix [27] was introduced, which shows all attributes by plotting the data points in a scatterplot view for each dimension pair. The scatterplots are placed as tiles of a matrix, where each view of a row holds the same attribute mapped on its y axis. Analogously each visualization in a column displays the same variable on its x axis. As the plots in the main diagonal would show the identity one dimensional visualizations like histograms or boxplots can be used to depict the distribution of the dimension values. Although scatterplot-matrices are capable to represent all attributes of a multivariate dataset, multivariate patterns can not be detected immediately.

Prosection views [44] are an extension of the projection technique of scatterplots by additionally using selections that represent sections of the data space with a low dimensional object. This combination of projections and sections is able to display structures of higher dimensionalities. Nevertheless the same intuitive data exploration, that the scatterplot technique provides, is only possible for experienced users.

The most popular geometric projection visualization for displaying high dimensional data is the parallel coordinates [60] view. To achieve this, the p attributes of a p dimensional dataset are drawn as parallel vertical lines that are uniformly spaced on the xy-plane. The values of each variable are linearly mapped on y positions for each axis separately, so that the minima of the dimension values are on the lower or upper end and the maxima on the upper respectively the lower end of the drawn axis. A data item is represented by a polyline connecting the values of the object on each dimension by intersecting the corresponding axis at the appropriate y position.

The major advantage of this technique is the possibility to visualize high dimensional datasets. The number of the dimensions that can be shown is only limited by the resolution of the view in the horizontal direction. The drawback that arises from the plot of more than 15 dimensions is that correlations between dimensions and patterns in the dataset itself can no

longer easily be perceived. Also text information showing the names of the attributes as well as their minima and maxima can not be presented for each dimension.

The drawbacks of the parallel coordinates are that an attribute can only have two neighbouring dimensions and that a high number of data items can make it impossible to detect any pattern. While for the first disadvantage a simple change in the order of drawn axes can be performed by the user, the second drawback can not be solved as easily. Solutions for this problem are introduced by the use of hierarchical parallel coordinates [41], as discussed in further detail in section 2.3.1 and by heuristics that reveal structures in parallel coordinates views as discussed by Johansson et al [63].

2.1.2 Icon Based Techniques

Icon based techniques map the attribute values of the data items on the properties of icons, in a way so that the observer can easily detect differences between them. One of the most famous approaches are the Chernoff Faces [25] introduced by Herman Chernoff in 1973. They use two dimensional line primitives to picture a face per data point, where the values of the data item influence the shape of the face as well as the facial expression. 18 independent features like the length of the nose or the size of the eyes make the representation of that many dimensions possible. Additionally the faces can be positioned according to two dimension values on the *xy* visualization plane. This representation of data items is justified by the fact that humans are used to recognize faces and to interpret their expressions. Furthermore it is assumed that users employ more intense with Chernoff faces as with comparable iconic techniques. Admittedly studies [83] [79] show this technique does not present significant advantages over other iconic graphics. Additionally the representation of each data item by a Chernoff Face introduces a tremendous limitation of the number of data points that can be depicted.

In contrast to Chernoff Faces a visualization of a higher number of data items can be achieved by Stick Figures [90]. The values of two dimensions are used again to position the icons on the visualization space. The remaining attribute values are mapped on the angles of joints and the length of the limbs represented as lines. Large datasets lead to dense Stick Figure visualizations, which are similar to textures. Monotonic areas of these textures can be interpreted as clusters, whereas outliers may be detected as single icons in monotonic areas that have a significant different shape as their neighbouring Stick Figures.

Star glyphs [109] are a further popular icon based technique, that maps the attribute values of data items on the lengths of lines that leave a point position in evenly spaced directions. The outer ends of those lines are connected by a polyline. The behaviour of data items is discriminated by examining the convexity of the outline of these glyphs. Thus outliers can be detected by comparing the mean glyph shape with those of single data items. Also the main

trends can be analysed easily by identifying the most frequently appearing convexity patterns.

While the star glyphs can also be seen as a variation of the parallel coordinates, shape coding [15] is similar to pixel based techniques, because it uses a small two dimensional array of pixels to depict a data item. The number of pixels corresponds to the number of attributes in which the object is defined. The colour of the pixels corresponds to the dimension values of the data point. Usually a border separates the data item representations from each other.

2.1.3 Pixel Based Techniques

Pixel-based visualizations try to represent each data item by one pixel. If the number of pixels is not sufficient for all objects, several data points are mapped to a pixel. This has the advantage that a large number of data items can be displayed without cluttering. The values of a dimension are used for a colour mapping, while each attribute is depicted in a different window. Thus the user can apply ordering with respect to the values of one variable. The other windows depict the data items according to their sorting position in this attribute. This allows the detection of functional dependencies and dimension similarities. To perceive those patterns easily several pixel arrangement heuristics have been introduced. While line-by-line or column-by-column alignments are suboptimal, screen-filling curves provide a clustering behaviour of data items and thus easier pattern detection. An extensive discussion about ordering techniques is given by Keim [71].

Alternatively to the visualization of each dimension in a separate window, a grouping technique was introduced, where all dimension values of a data item are depicted in a two dimensional pixel array [71]. Those arrays can be ordered like the single pixels, but to improve the pattern recognition they should be separated from each other by a border. This approach reduces the number of items that can be visualized in one view, and is similar to the shape coding.

A popular extension of this basic pixel-based approach is the pixel bar charts technique [70], for which the screen space is divided horizontally and vertically according to the number of categories that two variables hold. Those attributes are mapped on the x and the y axis respectively so that the widths and the heights of the introduced regions correspond to the number of data items that belong to them. If no categories are present a binning can be introduced. The pixels within a region are coloured according to a user defined attribute. The values of further two dimensions can be incorporated by arranging the data values within the regions. The user can interactively specify which variables are used for each of those mapping operations, which represents a powerful interaction tool for the visual data exploration.

Pixel bar charts combine the concept of bar charts and pixel based visualizations. Therefore no aggregated data is shown in the bars and the screen space is optimally used, while

simple bars represent only one value and do not fully cover the visualization space. Thus this approach can be seen as a generalization of traditional bar charts, because the interior pixels of the bars are used to visualize single data items of a multidimensional dataset. Additionally the major advantage of the pixel based visualizations that can depict a large amount of data without overplotting applies.

2.1.4 Hierarchical Techniques

Hierarchical visualization techniques create illustrations where the dimensions are interleaved. Therefore a subset of attributes is chosen to allocate the screen space with a common visualization technique. The further variables are iteratively nested into this view.

The concept of Worlds within Worlds [38] applies this approach by using two- or three dimensional coordinate systems to present a subspace of a high dimensional dataset. Into this initial view other visualizations are embedded to grant the user insight in all dimensions of interest. Therefore in an already existing coordinate system a new system can be created, which is called the inner world whereas the system that surrounds the inner world is referred as outer world. The position of the origin of the inner world specifies the values of the dimensions mapped on the axes of the outer world. This nested structure of coordinate systems allows the visualization of high dimensional data, but for exploration purposes interaction techniques like the change of the allocation of axes, the manipulation of the positions of the origins of inner worlds as well as the necessary navigation techniques through visualizations of a three dimensional space have to be integrated.

Dimensional stacking [78] is another approach which embeds dimensions within the visualization of other attributes. Therefore an iterative discretization of the screenspace is performed. The first dimension is chosen for the horizontal axis, and according to its values the visualization space is divided into sections delimited by vertical lines. The next attribute is chosen to divide the vertical axis which creates rectangular tiles representing each possible value combination of the first two dimensions. The rectangles are iteratively subdivided in the same manner by using the remaining dimensions. This allows the visualization of a large number of attributes. Each final rectangle is coloured according to the number of data items that shows the corresponding values of this introduced subsection. This visualization is adequate for categorical data. Continuous values have to be binned for this technique. It is recommended that the outer dimensions use small numbers of bins, while the inner attributes can be depicted with further detail.

2.1.5 Interaction Techniques

Besides the different types of visualizations also concepts for interactive operations have been developed, that allow the modification and adaptation of the data illustration. These techniques are the key elements that make the exploration of datasets possible. According to the visual information seeking mantra [108] the visualizations themselves are used to give a first overview of the data. Afterwards interactive zooming and filtering operations are applied to examine interesting data items and to visually exclude objects of low importance. Details-on-demand operations can be applied on the selected set of data points to retrieve their attribute values themselves or alternatively numerical summaries.

Zooming operations help the user to scrutinize interesting patterns in further detail by enlarging their visualization. Therefore it is useful to apply techniques that also keep track of the overall context, in which the examined portion of the data is seen. The so called Fisheye Views [104] provide such smooth distortion techniques that magnify the data items of interest and reduce the zoom factor step by step for objects farther away from the inspected position.

To filter out uninteresting data items the popular concept of drawing selections is applied, which is commonly performed via mouse interactions. Therefore the user highlights a set of data items that is of special interest. In contrast to applying a query on the data items so that only those are shown that fulfil a set of criteria, this interactive filtering approach provides instant visual feedback. This means during the process of creating the selection the user already perceives, whether the highlighted data points match the search properties. Thus an information drill down process can be realized, meaning that the user iteratively applies filtering operations by interactive manipulations of the visualization until only data objects of interest are selected. Thereby selections that are already drawn on a subset of data items represent a refinement of the previous filtering and thus are no complete reformulation of the constraints, that the data points have to fulfil [11].

Finally the selected data items could be extracted as well as exported into a file, so that a re-use of this subset for further analysis is possible. To comprehend how those data objects were identified, all interaction steps have to be recorded and possibly even stored with the data subset.

2.2 Statistical Methods for data exploration

The creation of statistics and the analysis of sets of data values, the so called samples, is a task many scientists were working on for more than 300 years. With the introduction of the computer the working field and thus the number of statistical algorithms has grown tremendously.

For the scope of this work the statistical routines that could enhance the visual data exploration are of special interest. Therefore the statistical outlier detection is of high importance, because the distinction between the intrinsic data and the outlying values is an initial task for the data analysis. Afterwards a popular processing step is to characterize groups of data to get a better overview and to determine which trends are dominant. Therefore clustering processes provide a big variety of grouping heuristics. But often the dimensionality is too high for an efficient clustering that can be analysed and interpreted. Therefore routines for the dimension reduction are in use, to capture the main part of the information of the dataset in a lower dimensional subspace. A slightly different procedure of determining an alternative data representation with small number of attributes is the feature subset selection, where the most informative variables are chosen to make further processing steps more efficient.

Consequently this section gives an overview about the fields of statistical outlier detection, dimension reduction, subset selection and clustering.

2.2.1 Outlier Detection

Outliers are data items, that seem to be significantly different from the remainder of the data based on a measure defined by the user [14]. This property indicates that those objects have to be treated separately. Thus a detection of outliers is a crucial task in data exploration. In contrast to the detection of clusters in the data, outliers are a group of data points that can be heterogenic, which means that they do not show the same pattern in general, while cluster members are similar to each other. Furthermore the number of outliers is usually rather small, so that the search is concentrated on a minority of the data.

The visualization of one, two or three dimensional data allows an easy and fast identification of outlying values by the human pattern recognition skills. The detection of multivariate outliers with a dimensionality higher than 3 is no longer tractable by means of simple visualization. A very popular way to detect high dimensional outliers that is in use in various information visualization applications is to detect extreme values per attribute via robust statistical measures [63]. An example showing that it is not valid to identify multivariate outliers by features in lower dimensional subspaces is discussed in section 4.2.3.

Depending on the heuristic that is applied for outlier detection, different definitions of outliers are proposed. This section gives an overview over the main categories of outlier identification and their most prominent algorithms.

Distribution based techniques

Statistical methods are often based on theoretic distributions. The most prominent high dimensional distribution is the multivariate normal distribution. Its shape can be easily described by

two parameters. The first parameter specifies the location of the distribution. Therefore usually the mean vector of the given sample is used to identify the center of the dataset. To explain the spread as well as the shape and orientation of the point cloud the covariance matrix is calculated, which holds the variances of the individual attributes and the covariances describing the linear relationship between the dimensions of the dataset. If these parameters of the multivariate normal distribution are given, the Mahalanobis distance [82] can be computed for each data item, which considers the shape of the distribution and its location and thus indicates how far a data item is displaced from the center of the dataset. Consequently this distance measures the outlyingness of data points, if there is just one group of objects that shows approximately multivariate normal distribution.

But to create reasonable results the center and the covariance matrix for a dataset has to be estimated robustly. Therefore two prominent estimators have been developed: the Minimum Volume Ellipsoid (MVE) [99] and the Minimum Covariance Determinant (MCD) [99]. MVE considers those data items for the calculation of the mean vector and the covariance matrix, which lie in the hyperellipsoid with minimum volume containing at least the half of the data points. MCD uses the same approach but the criterion for the considered hyperellipsoid is that the determinant of the covariance matrix based on the objects in the hyperellipsoid has to be minimal. Because it is not feasible to investigate all possible subsets of a large datset to find the optimal ellipsoids, the heuristics MINVOL [102] and FAST-MCD [101] for a fast calculation of sufficiently good solutions have been introduced. The latter is outlined in section 3.2. The Mahalanobis distance that is based on those robust estimates for location and spread is called robust distance.

The disadvantage of this distribution based approach is that the dataset must be nearly multivariate normal distributed. Otherwise the results of the robust distance are not reliable. Thus datasets with different distributions have to be transformed, which can be cumbersome. Also groups in the data can falsify the introduced outlyingness measure.

Distance based techniques

Distance based techniques rely on the calculation of the k nearest neighbours of a data item. Thus an object can be considered as outlier, if the distance to its k nearest neighbour (D^k) is larger than a user specified threshold d [74]. This definition introduces a simple and intuitive heuristic to identify outlying data items. But the major drawback of this detection rule is that the user has to specify a distance limit that depends on the number of used dimensions and the range of values per dimension, what does not allow the reuse of a given value for different datasets and different considered attributes. Furthermore a try and error approach is required to figure out correct threshold values. Consequently the rule was adapted, so that those n objects with the highest D^k values were considered as outliers [96]. The parameter n can be specified

by calculating a percentage of the total number of data items in the dataset. An important advantage of the D^k measure is that it introduces a ranking of outlyingness and also allows a fuzzy decision boundary between outliers and "normal" data items.

Because the calculation of D^k for all data points in a large dataset is cumbersome, a partitioning algorithm was introduced, which calculates the maximum values of D^k per introduced data subset. If there are maximum values that are smaller than the highest n D^k values, that are evaluated at that time, a pruning takes place, so that partitions with a small maximum D^k are no longer examined [96].

Density based techniques

Density based algorithms in general apply a volume parameter and the so called *MinPts* parameter that steers the minimum number of points within a given volume. Together they define a threshold that decides if objects or regions are merged to a new cluster. Data items that could not be assigned to a cluster and thus are outside of dense regions are considered as outliers. A cluster approach that is based on this concept is discussed in section 2.2.2.

An example that deals with this concept is the Local Outlier Factor (LOF) [21], which uses the *MinPts* parameter to calculate a value for each data item indicating its outlyingness. Data items located deep within clusters show an LOF of approximately 1, while high LOF values are assigned to isolated objects. Thus based on the density of neighbouring data points a continuous measure allows a fuzzy outlier detection, which has the drawback, that the calculation of the LOF values for all data items in a large dataset is computationally expensive.

Other outlier detection techniques

Depth-based techniques try to compute the multivariate depth of data items, which indicates the location of data points. Objects with the highest depth are considered to be near the center of the dataset, while low depth values indicate that a data item is at the border of the data cloud and thus a potential outlier. But the calculation of a measures like the half-space depth [113] is computationally expensive for high dimensions. Thus only for 2 dimensional data efficient depth-based algorithms exist [103], [65].

For sparse high dimensional data density based as well as distance based methods become inefficient. The higher the number of dimensions the lower is the density of the given data items and thus the difference between isolated data points and members of clusters becomes smaller. The same applies to techniques based on a k nearest neighbours approach, because the more dimensions are considered, the higher are the distance values, which means that the difference of D^k values between possible outliers and "normal" objects is represented in the last values of floating point numbers. This phenomenon is called the curse of dimensionality. Therefore an

evolutionary search technique for an efficient detection of lower dimensional projections that allow an easy detection of outliers was introduced by Aggarwal et al [9].

2.2.2 Clustering

A clustering algorithm partitions a dataset into groups, the so called clusters. The data items that belong to the same cluster are more similar to each other than to members of other clusters. Thus a clustering introduces a simplification of the dataset, which outlines the main patterns in the data.

A central decision, that has to be made for grouping the data, is the choice of a similarity measure between two objects. Certainly this definition depends on the context, for which the clustering is used. A common solution is to calculate distance measures like the Manhattan or Euclidean distance, which describe the dissimilarity between data items.

In this section the most important categories of clustering algorithms and examples of their representatives are mentioned. Because of the vast number of cluster procedures developed in the last 30 years it is far beyond the scope of this work, to refer to all outstanding developments in this field. Therefore respective survey papers are recommended [16] [62].

Hierarchical Clustering

Hierarchical clustering algorithms create nested cluster structures by a merging or by a division process. A merging procedure builds the clusters in a bottom-up manner by assigning each data item to a cluster. In an iterative process the most similar clusters are merged to a new cluster until the whole dataset is represented by one single cluster. A divisive approach creates this group hierarchy in the inverse and thus top-down way.

To achieve the merging or the division of clusters, the algorithms commonly operate on a $n \times n$ matrix holding dissimilarity values between each pair of the given n data items, which is called the connectivity matrix. Furthermore besides measures that distinct between objects also the differences between clusters have to be considered. Therefore the so called linkage metrics have been introduced, which are heuristics to calculate the distance between subsets. The most popular linkage metrics are single link [110], average link [118] and complete link [73]. Single link calculates the distance between two sets A and B as the minimum distance between a pair of objects x and y, where $x \in A$ and $y \in B$. This heuristic suffers unwanted chaining effects, meaning that elongated clusters may be created. In contrast to that complete link methods consider the maximum distance between each pair of members of the subsets of interest. Therefore the variety of possible cluster shapes is limited in comparison to single link algorithms. A compromise between those two linkage metrics is the average link, which considers the mean of distances between all pairs of x and y.

The hierarchy created by a hierarchical clustering algorithm can be depicted in a dendrogram, which is a tree like structure, where each node represents a cluster and each edge represents a subset relationship between those groups. The root node stands for the whole dataset, while the nodes of the level d, introduce a partition of the data into d clusters, if in each iteration two clusters have been merged respectively if a splitting procedure created two subclusters. The dendrogram representation allows an interactive choice of the granularity of the partition, which is a significant advantage in comparison to partitional clusterings, where the number of clusters has to be predefined by the user. A drawback of the hierarchical approach is that hierarchies that have been created during the clustering are not changed anymore, so that no further optimization takes place.

Popular hierarchical cluster algorithms are BIRCH [123] and CURE [45]. CURE represents clusters by well scattered data items, which allows also the detection of clusters showing non-spherical shapes and reduces the influence of outliers. Sampling techniques make the efficient processing of large datasets possible. In comparison to that BIRCH is designed to work with high numbers of data items. The similarities of the objects are stored in a height-balanced tree, which can be incrementally updated. Consequently the first scan of the data items already provides a good clustering solution, which can be even improved by further iterations.

Partitional Clustering

Partitional clustering algorithms introduce groups of the data that are iteratively optimized with respect to an objective function measuring the quality of a cluster result. Usually an initial solution is created that is improved by reassigning data items to different clusters until a local optimum of the energy function is reached. The main drawback of those cluster approaches is that the computed local optimum can be far away from the best solution and the quality of the cluster result strongly depends on the initial solution, for which only heuristics exist, which do not guarantee a worst case boundary for the quality of the final partitions. Nevertheless partitional clusterings are very popular because of their simplicity and their intuitive and easy to interpret solutions.

The algorithm that is probably the most used clustering approach is k means clustering [50] [51]. It divides the dataset into a user defined number of partitions, which is represented by the variable k. The data items are assigned to the cluster with the nearest cluster center. Afterwards the centers, also called centroids in this case, are updated by calculating the mean vector of all members of a cluster. These operations are performed iteratively until no cluster center changes its position significantly. This procedure detects in general spherical shaped groups in the data. An elaborate description of the algorithm can be found in section 3.3.1.

The main drawbacks of this approach are. that outlying objects have strong influence on the solution, that only numerical data can be clustered and that the user has to specify the

optimal parameter k, which in general can only be achieved by several runs of the algorithm. But because of its popularity a huge number of publications propose adaptations of the k means clustering to overcome its disadvantages. To create better initial solutions that influence the quality of the final result, several k means procedures are applied on subsets of the data and the centers of the best solution are used as starting point for the clustering on the whole dataset [20]. An extension for categorical data was introduced by Huang [56]. To reduce the computational costs of the k means clustering, geometrical acceleration techniques were developed [89]. The adapted algorithm x means [88] proposes besides an acceleration approach also an estimate of the parameter k. Additionally to allow hyperelliptic shaped clusters an approach using the Mahalanobis distance as distance measure was tested [81].

Furthermore the fuzzy k means clustering [17] was introduced, which avoids that data items are assigned to only one cluster, although they are located at the border of it. Thus fuzzy clustering computes a membership value indicating the reliability of a cluster assignment. While data items near a cluster center have high membership values for the corresponding cluster, objects on cluster boundaries may be associated with several clusters. The algorithm is discussed in further detail in section 3.3.2.

Besides the k means approach the k medoids clustering has gained high popularity. In contrast to k means the cluster representatives (medoids) are chosen so that they are located in the densest area of the cluster, which reduces the influence of outliers. Popular algorithms are Partitioning Around Medoids (PAM) [69] and Clustering LARge Applications (CLARA) [69].

An also important category of partitional clustering procedures are expectation maximization (EM) [32] approaches, that assume that data items are samples independently drawn from a mixture model, consisting of several (unimodal) distributions. Thus those algorithms are also called probabilistic clustering. An EM approach makes an initial guess of the probability functions of the distributions and iteratively improves this solution. The result can be easily interpreted as the distributions of the mixture model are explained by the computed parameters.

Density-based Clustering

Density-based cluster algorithms are able to detect clusters of arbitrary shapes. In general the user has to specify volume and/or "number of objects" parameters that implicitly compute a density threshold, deciding which regions are connected to clusters. The major drawbacks of those approaches are the interpretation of the created cluster structures and the setting of a reasonable density threshold. The concept of a density-based clustering is shortly described on the basis of the algorithm Density Based Spatial Clustering of Applications with Noise (DBSCAN) [37].

DBSCAN needs two parameters that influence the shape as well as the number of created clusters. The first parameter ϵ defines those objects as neighbours of a given data item, which

have a distance smaller than ϵ. The second parameter $MinPts$ is used to define core objects, which are data items that have more neighbours as $MinPts$. As starting point each core object represents a cluster. In the first iteration the neighbours of the core objects are added to their clusters. Afterwards all neighbours of the cluster members are added, which allows a growing process according to the data density. Data items that are not added to a cluster are considered as outliers.

Other Clustering Approaches

Besides these main categories a huge variety of other methodologies like graph-theoretic or grid-based clusterings as well as search-based or evolutionary approaches for the detection of groups in data have been developed. Applications in combination with dimension reduction techniques were proposed for the clustering of very high dimensional data. Also the influence of outlying objects on the group finding process gave rise to fuse clustering techniques with outlier detection approaches [16] [62].

2.2.3 Dimension Reduction Techniques

The concept of dimension reduction is a well known technique used in the fields of pattern recognition, compression or analysis of functional dependencies. In information visualization the projection of multidimensional data points into a two or three dimensional space is a typical task, which allows the intuitive representation of data items in scatterplots.

But also other statistical procedures like clustering can benefit of a dimension reduction, because the most important information of the multivariate dataset is gathered in a low dimensional subspace. This allows a faster execution of clustering algorithms and forces those routines to focus on the main information, meaning that high frequent noise in the data is already extracted. Furthermore correlations between dimensions are summarized, what avoids the domination of the clustering by a group of dimensions showing the same pattern.

This section gives a short overview of the most prominent dimension reduction techniques. An extensive survey on this field is given by Carreira-Perpinan [23].

Principal Component Analysis (PCA)

The most prominent and widespread dimension reduction technique is the Principal Component Analysis (PCA) [61] [66]. The PCA finds uncorrelated directions describing the maximum variance contained in the data. Those directions are called the principal components and are linear combinations of the data dimensions. The first principal component represents the direction in which a given data cloud has the highest elongation and thus the highest variance. The i-th

principal component is perpendicular to their $i-1$ predecessors and describes equal or less variance than the $(i-1)$-th principal component. Thus the rejection of the last principal components which mainly capture high frequent noise in the data, introduces a reasonable dimension reduction. A mathematical discussion of the PCA is given in section 3.4.

Its popularity is based on the simple concept, the good results in practical applications and its efficiency, because the PCA can be calculated in polynomial time. But the simple model used for the calculation of the principal components has the drawback that only linear subspaces of the data can be created. Also the number of principal components that should be used to keep the majority of the information of the data is not obvious. But heuristics for finding the cut off point at the principal component, where the highest relative difference in the explained variance values takes place, help to overcome this drawback.

Projection Pursuit

The projection pursuit [57] [67] has in contrast to the PCA the objective to search for "interesting" projections of the data points. The aim is to employ the user's excellent pattern recognition skills to find structures in visualizations of the data items. Therefore projection pursuit procedures try to find one, two or three dimensional projections of the data that show patterns like clusters, gaps or outliers significantly. To achieve this, an unsupervised search procedure is started optimizing a given objective function, the so called projection index, which measures the quality of a linear orthogonal projection of the data. Because the distribution of the projected data should contain anomalies, a projection is of high interest, if it deviates from a normal distribution. The reason for this type of measure is that a normal distribution can be easily described by the mean vector and the covariance matrix and that low dimensional projections of high dimensional data items usually create normal distributed patterns.

The projection pursuit suffers the same problem as the PCA because only linear projections can be created. Furthermore the search for projections of interest is computationally expensive, because a multitude of directions has to be checked. Thus a sampling of directions has to be introduced as well as search heuristics that improve solutions which are stuck in local optima.

In contrast to the automatic detection of interesting projections the grand tour [13] [22] creates an animation of mappings. Therefore the p dimensional data is rotated and usually mapped in a two dimensional visualization area. The task of identifying projections of interest is handed over to the user, but the drawback is, that an exhaustive search of all possible mappings of a multivariate dataset may not have a time limit.

Kohonen's Self-Organizing Map (SOM)

A very popular method for dimension reduction which is also used in information visualization applications is Kohonen's Self-Organizing Map (SOM) [76]. SOMs try to capture the structure of a p dimensional dataset by applying an unsupervised learning scheme, which initiates a set of p dimensional reference vectors, which are associated with units of a two dimensional lattice, at random positions in data space. Afterwards for each data item the nearest reference vector is attracted to the position of the data point. The reference vectors of neighbouring units are also influenced, so that linked units in the two dimensional lattice represent similar positions in data space. By an iterative execution of these update calculations the reference vectors improve their fit of the main agglomerations of data points in data space.

The main advantage of this dimension reduction technique is that it combines a topology preserving mapping of multivariate data on a plane with a clustering procedure. This is also expressed by visualization techniques for SOMs discussed in section 2.3.1. Furthermore the reference vectors represent the distribution of the data. But in contrast to clustering procedures like the k means clustering the SOM procedure is not guaranteed to converge. It is also not possible to define an objective function that could be optimized.

Based on the SOM algorithm many new approaches for topology preserving mappings of high dimensional data on a two dimensional lattice were made. Examples for these are the incremental grid growing [19], the growing hierarchical self-organizing map [33] or the growing cell structures [40]. An alternative to the SOM procedures is the Generative Topographic Mapping (GTM) [18], which also creates topologically continuous maps. GTMs are based on solid statistical theories, but they are not suitable for drastic dimension reductions, where hundreds and thousands of dimensions should be handled.

Multi Dimensional Scaling (MDS)

Multi Dimensional Scaling (MDS) [77] tries to find a function, that maps the p dimensional data items to a lower dimensional space in a way that the relation between the data points is kept as accurate as possible. This means that data items that have similar values in the data space should be mapped closely to each other. To find the optimum mapping function a search procedure in the space of functions of interest is started. The objective function that is minimized considers the difference of distances between the mapped data items and their actual dissimilarity in the dataset.

This procedure allows creating any reasonable linear or non linear mapping of the data. But one drawback of MDS is that, if the data has clusters, the mapped data items may show large differences between the clusters but the spread within the groups may not be represented in the correct magnitude. Thus this approach often pronounces the global structure of the data,

but local properties may be neglected. A further challenge by using MDS is the selection of the correct dimensionality of the mapping. It is recommended to try several settings [23]. In contrast to the PCA it is also not possible to reduce the dimensionality of an MDS mapping by omitting one of the coordinates of the created subspace.

Efficient algorithms for MDS apply spring models that iteratively adapt the lower dimensional representatives of the data items to their distance relation in the data space [84] [24].

2.2.4 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) [59] is a recent development which creates linear combinations of the original attributes, so that these artificial variables - the so called independent components - are maximally statistically independent from each other. With the FastICA [86] an efficient algorithm for the calculation of the independent components is provided.

ICA can be used to detect interesting projections of the data, showing significant structures. Similarly to the projection pursuit it is assumed that interesting mappings do not show a normal distribution. Thus ICA can be applied to detect all independent components, which are nongaussian projections of the data. Components that show normal distribution can be seen as the noise in the data. This concept allows the reduction of the dimensionality and the compression of the major information in the dataset by a set of artificial attributes. The main disadvantage of this technique is that the number of ”uninteresting” projections can not be determined in advance and thus the possible reduction of the attributes can not be influenced by the user. Furthermore in contrast to the PCA neither a hint concerning the quality of the dimension reduction is given nor can a statement about the importance of one of the computed artificial attributes be made.

2.2.5 Feature Subset Selection

In contrast to dimension reduction techniques, where new variables are created that can be used for visualization or clustering, the feature subset selection chooses the most important attributes for a user defined purpose. This has the advantage that the variables defining the lower dimensional representation of the data items can be easily interpreted.

In the context of unsupervised learning feature subset selection is a powerful tool that is primarily applied to clustering. But also for information visualization this technique gained more and more focus along with the increasing number of dimensions that has to be displayed within one view.

In general the importance of a variable is ranked with respect to a criterion. Often used measures for these rankings are saliency, entropy, density or reliability. A dimension is consid-

ered as salient, if it covers a high variance or a large range of data values. The entropy criterion has maximum values for uniformly distributed attributes, while the density measures how many variables are correlated with the current dimension of interest. In contrast to that a feature is called reliable, if it is measured in high quality and thus the errors are small compared to the range of the data values [47] [85].

2.3 Combination of Statistical Methods and Information Visualization

An increasing number of information visualization applications provide techniques for clustering and dimension reduction, but they differ in the way how these statistical routines are incorporated in the visualizations. The simplest approach is certainly to integrate the results of procedures like clustering in standard views as parallel coordinates or scatterplots. This allows the user to present and explore the outcome of the statistical routines, but it is not possible to interact with those algorithms. Thus if the user discovers possible mistakes in the result of a statistical technique, it has to be started again with different settings, which yields to a new solution that is not based on the previous outcome and may show completely different issues. Consequently this section categorizes proposals for combining information visualization with statistics in two classes, describing the graphical representation of results of statistical routines and their exploration on the one hand, and the integration of statistical functionality in an interactive visual data mining process on the other hand.

2.3.1 Visualizing results of statistical procedures

The majority of visualizations that are applied to explore statistical results concentrate on clusterings and dimension reductions. Although there are applications that also discuss the identification of outliers, no publications were found, that solely focus on the visual analysis of a statistical outlier detection routine.

Visualizing clustering results

The outcome of a clustering procedure can be easily incorporated into standard visualization techniques. The representations of the data items can be coloured according to their cluster memberships. Additionally - depending on which clustering technique was used - the representatives of a cluster (mostly the cluster centers) could be accentuated.

Furthermore approaches exist to use the statistical properties of the computed groups in the data to separate them accordingly in the visualization. To achieve this, the inverse of the

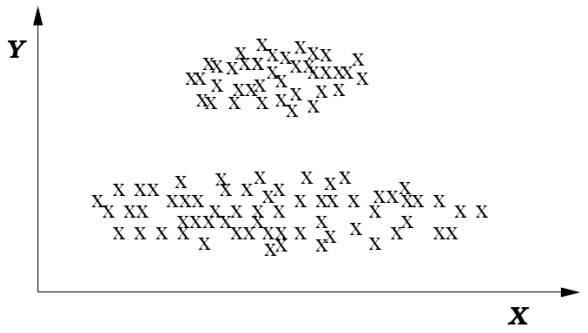


Figure 2.1: An example, for which the PCA would project the data items on a direction parallel to the x axis, while the LDA approach would choose a direction parallel to the y axis (image courtesy by Dy et al [36]).

population weighted average covariance matrix of all clusters and the covariance matrix based on the cluster centers are applied. The first matrix is used to eliminate distortions introduced by the cluster shapes. The second matrix spreads the data items so that the cluster centers become more distinct. Thus the data items projected on the first two principal components of this matrix product show the maximum possible separation of clusters created by a linear combination of original data attributes. Consequently this projection represents the best two dimensional visualization of high dimensional clusters [36]. The statistical fundamentals for this approach are introduced by the field of Linear Discriminant Analysis (LDA) [42]. The advantage of this concept is demonstrated by an example shown in figure 2.1, where the LDA approach would project the two groups on a direction parallel to the y axis, which results in a separation of the groups in the projection. The PCA is not able to achieve this result. A drawback of this approach is that outlying cluster centers have tremendous influence on the results and may create distortions that can lead to wrong conclusions.

An application that incorporates a variety of cluster algorithms and a visual exploration tool for the cluster results is gCluto [97]. It provides agglomerative, partitional, graph partitional and bootstrap clustering procedures. For those routines an interface is implemented that allows the power user to steer the common parameters of the clustering as well as provides the non-experienced user to create useful partitions by standard settings. For the exploration of the introduced groups of data items two visualization techniques are realized. The so called matrix visualization shown in figure 2.2 (a) illustrates the data as a table, where the data values themselves are replaced by colours. High positive values are usually mapped to red, negative entries are painted green, while items near 0 are represented by white cells. Cluster borders are indicated by black lines that separate the rows of the matrix visualization, which correspond to the data items. A dendrogram structure that is plotted on the left side of the matrix allows investigating the cluster structure in the data. Data items belonging to one cluster can be sum-

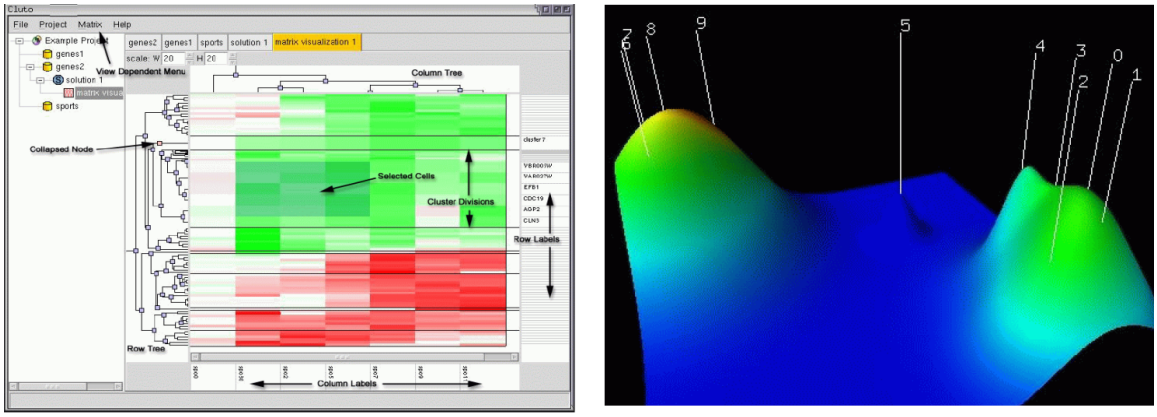


Figure 2.2: Screenshots from gCluto showing the matrix visualization (a) and the height field view (b) (images courtesy by Rasmussen et al [97]).

marized by collapsing the cluster. Afterwards the cells in the matrix representing the cluster are coloured according to the mean values of its objects. The same approach is applied for the columns, which represent the attributes of the dataset. Additionally zooming and selection techniques can be applied. A details-on-demand approach shows the values of cells and their cluster assignment.

The second visual exploration tool is a height field visualization (figure 2.2 (b)), for which an MDS is applied to project the data items into a two dimensional space. Afterwards height values that correspond to the similarity of the data items within a region are calculated for each position in the plane. For the visualization a free formed surface is created that represents clusters as mountains and their boundaries as valleys. Their position to each other indicates the similarity between the clusters. Additionally a colour mapping according to the standard deviation of the data items per cluster is applied. The cluster centers are accentuated by labels that are connected to their position on the surface by lines. As interaction technique the user can navigate through the three dimensional visualization.

While gCluto developed its visualizations for a set of different cluster procedures, applications like H-Blobs [111] focus on the properties of a certain category of clusterings. H-Blobs introduces a visualization that is designed for an adapted hierarchical clustering approach. The clusters are illustrated in three dimensional scatterplots by surfaces that are created by placing blob primitives at the cluster centers. Those primitives introduce ellipsoidal gaussian fields, on which an isosurface generation is based. Because of applying a hierarchical approach also a level-of-detail adaptation is possible. Thus if a lower hierarchy containing a higher number of partitions is chosen, also the precision of the data visualization is increased.

The same level-of-detail procedure is used for the hierarchical parallel coordinates [41].

In this application the use of hierarchical clustering to enhance the parallel coordinates visualization concept is introduced. After the clustering process a level in the dendrogram and consequently a number of clusters is chosen and visualised with the parallel coordinates. But a cluster is no longer represented by the polyline illustrations of its data items but by variable-width opacity bands. The width of the bands is defined on each axis by the minimum and maximum attribute values of the cluster objects. Between the axes the width is linearly interpolated. The maximum opacity of a band depends on the size of the cluster. While the borders of the band are transparent, the center has the maximum opacity value. For the area between linear interpolation for this property is performed.

Besides the main applications introducing modifications of cluster routines and visualizations to allow a combination of those techniques also the effective exploration of clustering results with common information visualization functionality is discussed. The strengths of parallel coordinates and its interaction techniques for the analysis of partitioned data is treated by Wegman et al [119]. Furthermore the use of the grand tour focussing on the generation of intuitive plots of projections of the data is outlined. Kandogan [68] outlined the usefulness of star coordinates for the investigation of multivariate features like trends as well as outliers. An overview concerning the interactive use of dendrogram structures with focus on the exploration of genomic microarray data is given by Seo et al [106].

Feature Subset Selections and Visualization

In this section three examples of feature subset selection applications which incorporate visualization techniques are discussed. These tools do not consider a supervised learning approach, where it is possible to verify the quality of a chosen set of attributes with respect to an error measure.

Dy et al [36] introduced the so called Visual Feature Subset Selection using EM Clustering (Visual-FSSEM). For this application EM clustering is used to find groups using a specified subset of attributes in which the p dimensional data items are defined. To avoid the examination of all 2^p possible dimension subsets, a greedy heuristic is applied so that iteratively that attribute is eliminated that introduces the smallest deterioration of the clustering result with respect to a user defined objective. The provided quality criteria for this incremental subset creation are the scatter separability or the maximum likelihood [35]. On each of the introduced feature subsets an EM clustering is applied. The result is visualized in scatterplots using the LDA projection approach outlined in 2.3.1. The user chooses depending on the visual presentation of the results the optimum feature subset.

With respect to visualization of very high dimensional data the Dimension Ordering, Spacing and Filtering Approach (DOSFA) [120] was proposed, which combines interactive multivariate visualization techniques with heuristics to determine the importance of attributes.

Based on the thesis that variables showing similar patterns should be visualized closely to each other (e.g. as neighbouring axis in parallel coordinates) [12], a dimension ordering is introduced with the aim to minimize the dissimilarities between adjacent dimensions in visualizations like the star glyph or the parallel coordinates. Therefore a hierarchical clustering is performed on the dimensions via the Visual Hierarchical Dimension Reduction (VHDR) [121] system, which is discussed in section 2.3.2. The clustering result is visualized by the InterRing [122] technique, a radial tree visualization tool, that allows the interactive exploration of the cluster hierarchy. Afterwards the attribute order is established by sorting the clusters of each hierarchy level according to a similarity criterion or according to their variance of data values. To compare the clusters representatives are calculated by averaging the dimensions within the groups. The user has the possibility to re-order the attributes or clusters of variables via the InterRing interface. Additionally to the ordering a spacing between the dimensions is established, placing similar attributes closely to each other in the multidimensional views to meet the requirements of the Gestalt Law on proximity [75]. The user can influence the spaces between depicted dimensions by zooming and panning operations as well as by distortion techniques that emphasize selected dimensions, while the context is kept. This facilitates the recognition of dimension groups and the perception of patterns in very high dimensional data. A feature subset selection can be applied by using a so called dimension filtering heuristic. This approach chooses an attribute of a possible set of very similar variables. Dimensions that are of low interest for the user can be neglected for the visualization. The InterRing interface therefore allows the selection of dimensions or groups of attributes for visualization.

A different approach to identify attributes of special interest for the user is introduced by the Rank-by-Feature Framework [107]. In this application one and two dimensional axis parallel projections are ranked by the importance of the structure that they contain. For one dimensional projections measures can be selected that test the attribute for uniform or normal distribution as well as for the number of possible outliers or unique values. Depending on the needs of the users one of theses ranking criteria can be chosen which leads to an ordering of the variables according to their importance. The validity of this ranking can be examined by inspecting histogram and boxplot visualizations of the projections. For the two dimensional mappings the functional relationship between the illustrated dimensions can be examined by correlation coefficients or by the least squares errors of a linear respectively a curvilinear regression. If the projections should be ranked according to two dimensional distribution patterns, then the number of data items in a user defined region or a test for uniformity can be evaluated. Afterwards the introduced ranking can be visually examined by a scatterplot view, showing a user selected two dimensional projection. The application eases the analysis of interesting dimensions and two dimensional structures, because especially for high dimensional datasets a pre-selection of attribute pairs with respect to an importance criterion is crucial to avoid the investigation of all possible projections. Visualization serves as mean of validation and exploration, which on the

one hand allows the verification, if the chosen importance measures achieves correct results, and on the other hand makes the immediate investigation of the found structures possible.

Visualization of dimension reduction techniques

As dimension reduction techniques like PCA and MCD introduce a mapping of the data into a lower subspace, all visualization techniques for multivariate data can be applied to illustrate the projected data items. This does not apply to the results of SOMs, because they represent the multivariate datasets by reference vectors, which are linked to positions in a two dimensional grid. For the visualization of this grid a huge variety of contributions has been made in the field of machine learning. This section summarizes the most important publications.

One of the most cited visualization techniques for a SOM is the U-Matrix [114], where the sum of distances of each reference vector in data space to those of neighbouring units is calculated. This measure is used for colour mapping, so that each unit represents an array of pixels with the same appearance in a two dimensional visualization. Because of the presentation of the data in a map the metaphor of a landscape is stressed, where clusters are represented as valleys, cluster boundaries can be identified as mountains and outliers are indicated by funnels. A similar approach is used by the P-Matrix [115], for which a density estimate is performed at the position of each reference vector so that the colours for the units can be mapped according to the density in data space. This leads to the representation of clusters by plateaus, while ditches identify their borders. A combination of the distance-based U-Matrix and the density-based P-Matrix was introduced by the U*-Matrix [116] that aims to depict clusters by coherent colour regions and to accentuate cluster boundaries.

Other approaches like the hit histogram [94] map the number of data items that are associated with a reference vector on a symbol that is bright if nearly no object is near the vector and fully coloured otherwise. Thus light areas indicate sparsely populated regions in data space, while dark areas identify high density. Smoothed data histograms [87] introduced a parameter to reduce the noise of visualizations created by hit-histograms. This parameter can be used to steer the level-of-detail, meaning that it also decides how accurate the visualization shows the actual data structures. Contour plots can even simplify those representations and exaggerate dense regions.

For the colouring of the SOM units also clusterings on reference vectors are common [117]. The most popular approaches use k means or hierarchical cluster algorithms. The same colour is assigned to the units of a cluster. The main drawback of this approach is that the number of groups in the data has to be specified by the user. Based on the clustering result also a flow field representation of the map was proposed, where for each unit an arrow showing to the nearest cluster center is drawn [93].

2.3.2 Interactive collaboration between Information Visualization and Statistical Procedures

A tight collaboration between statistical routines and interactive visualization techniques is rarely achieved. In this section three examples that accomplish this degree of integration are discussed.

Visual Hierarchical Dimension Reduction (VHDR) [121] applies a hierarchical clustering on the dimensions of a dataset. The introduced nested attribute group structure is afterwards visualized by InterRing [122]. This circular, space-filling visualization technique illustrates the root of the tree structure by an inner ring, which is surrounded by ring segments that illustrate the nodes and thus the dimension groups of the clustering result. Segments representing child nodes of the same cluster are similarly coloured. Distortion techniques, as well as rotations, zoom and panning operations allow the exploration of the dimension relationships. Selection techniques can be used to highlight variable groups of interest. Also the manipulation of the dendrogram hierarchy is possible. After the exploration and the modification of the clustering result a representative dimension per cluster or the center of all dimensions in a group is chosen. These attributes are applied for visualizations of the data items and further operations in the data mining process. Consequently this approach introduces a subset selection procedure, that is based on a statistical routine and incorporates the user's knowledge and experience, which is communicated via an information visualization interface.

The coordination of computational and visual techniques for feature selections and clusterings is also discussed by Guo [46]. In the introduced application the feature subsets are evaluated according their "goodness for clustering", meaning that only those combinations of dimensions, that show significant cluster structures in their spanned subspace, are of interest. As quality measure the so called Maximum Conditional Entropy (MCE) is calculated for each pair of variables. An ordering of the attributes, so that similar dimensions are positioned next to each other, is accomplished by a hierarchical clustering. The MCE values of the ordered dimensions are afterwards illustrated in a colour coded matrix, so that blocks of bright colours indicate subspaces that contain strong group structures of data objects, which is shown in figure 2.3 (a). The user can interactively select the attributes of interest in the matrix visualization or apply a heuristic that determines these data subspaces based on the MCE information. Afterwards a clustering procedure that allows the detection of groups with arbitrary shapes is applied on the dimension subset. To achieve this, a hierarchical clustering procedure was enhanced by concepts from density and graph based techniques. The additionally introduced parameters can be steered via interactive visualizations. Firstly the clustering divides the detected subspace into hyper-cells, for which the numbers of data items that lie in them are considered as density measure. The user can set a cut-off value that excludes cells with lower density values from the actual clustering to speed up the group finding process. For the interactive tuning of this

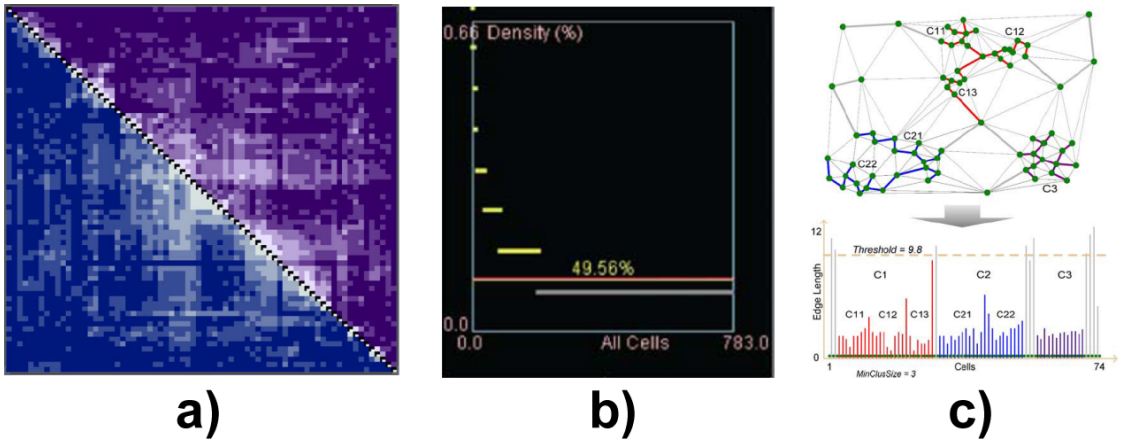


Figure 2.3: Interactive visualizations for the integration of user interaction in feature subset selection and clustering tasks: the matrix visualization for the MCE values (a), the graph showing the density of cells and a cut-off limit (b) and an illustration of a possible cluster structure, which is summarized by an interactive visualization steering the number of clusters (c). (Images courtesy by Guo [46])

parameter a graph showing the decreasingly ordered density measures is used. The cut-off limit is drawn as a horizontal line that also specifies the number of considered cells by the x position of its crossing point with the graph (figure 2.3 (b)). Afterwards the hierarchical clustering is applied on the selected cells. Thereby the cells are also ordered by a minimum spanning tree approach. A plot showing each cell versus the distances to its two neighbouring cells shows valleys as clusters and ridges as cluster boundaries. The user can steer the number of cell groups by introducing a limit for the distances. Distances that exceed this limit introduce a new cluster boundary (figure2.3 (c)). Consequently this parameter can be used for manipulating the level-of-details, because the lower the distance limit is set, the higher is the number of detected clusters and the more precise is their fit to the data structures. All these visualizations for parameter tuning are linked with a parallel coordinate plot that illustrates the detected clusters.

Based on the clustering procedure OptiGrid [54], which is designed to detect groups in very high dimensional datasets, the HD-Eye approach [55] integrates innovative visualization methods into the clustering procedure to achieve better results. OptiGrid uses a density function that identifies agglomerations of data items and applies splitting procedures, where the data space is divided into half spaces by separators, which can be geometric objects like hyper planes. Therefore only separators are chosen that pass through low density regions. The introduced subspaces are processed iteratively until no further subdivision can be applied. The HD-Eye approach integrates the human pattern recognition skills to define data projections, that allow the visual detection of gaps in the data, and finally the user can define separators for the partitioning process. For the quality appraisal of a projection iconic visualizations are created that indicate,

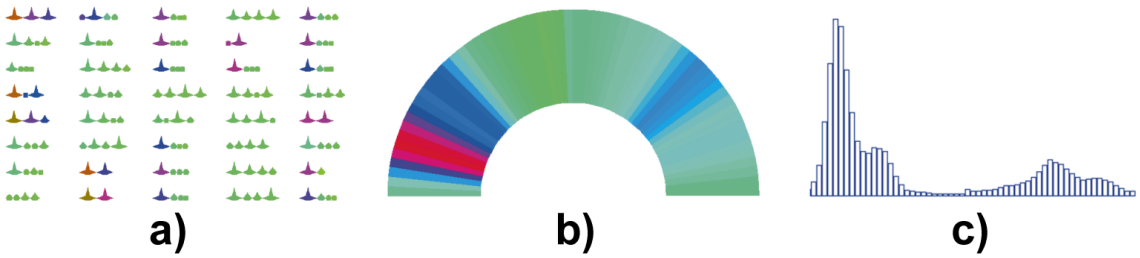


Figure 2.4: Visualizations as user guidance for the HD-Eye approach: Iconic visualizations (a) showing good separation properties of projections by large spikes, while the color indicates the number of data items that can be well divided into subcells. Colour-density (b) and curve-density plots (c) indicate for the projections where separators can be introduced. (Images courtesy by Hinneburg et al [55])

whether the mapped data items can be separated in groups (figure 2.4 (a)). If the user chooses a projection of interest, colour-density (figure 2.4 (b)) and curve density plots (figure 2.4 (c)) that stress the agglomerations of objects by colour mappings respectively by histogram like visualizations are used to define separators that divide the examined space into subspaces, which represent new cells for the OptiGrid cluster algorithm. Consequently the pattern recognition skills of the user are applied to create the best possible subdivisions while algorithms provide pre-selected projections and decide, whether further splitting operations are possible.

Chapter 3

Statistical Fundamentals

This section discusses the statistical basics that are of high importance for this work. Therefore instructions for the calculations and formulas that are necessary for understanding the usage and the implementation of the routines integrated in the statistical library are presented. The explanation of these fundamentals is divided into the six topics statistical moments, correlation and covariance, clustering, principal component analysis (PCA), linear regression and finally theoretic distributions and statistical tests. The reason why these functionalities are chosen as well as their usefulness for information visualization applications are outlined in section 5.

3.1 Statistical Moments

Statistical moments [82] are estimates of parameters concerning the location, the scatter or the shape of the distribution a given set of values, the so called sample, comes from. For the calculation of the center of N data values x_i the estimators arithmetic mean, median and α - trimmed mean exist. For the median as well as for the α - trimmed mean the sorted values are needed. Thus $x_{(i)}$ represents the i-th smallest element in the given formulas of table 3.1.

Arithmetic mean	$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$
Median	$median(x_1, \ldots, x_N) = \tilde{x} = \begin{cases} x_{(\lfloor \frac{N}{2} \rfloor + 1)} & \text{if N is uneven.} \\ (x_{(\lfloor \frac{N}{2} \rfloor)} + x_{(\lfloor \frac{N}{2} \rfloor + 1)})/2 & \text{if N is even.} \end{cases}$
α - trimmed mean	$m(\alpha) = \frac{1}{N - \lfloor N * \alpha \rfloor}(x_{(\lfloor N*\alpha \rfloor + 1)} + \cdots + x_{(N - \lfloor N*\alpha \rfloor)})$

Table 3.1: The moments describing the location of a given sample.

For the estimation of the magnitude of the spread of the values around the center the most popular moments are the variance, the standard deviation or the mean of absolute deviations. But also the robust measures median of absolute deviations (MAD) and α - trimmed standard deviation should be considered, because they are resistant against the influence of extreme values. To compare the MAD and α - trimmed standard deviation with the standard deviation, their values have to be scaled. The MAD is multiplied with $\frac{1}{0.675}$ and the α - trimmed standard deviation with a constant that depends on α. (Those coefficients are evaluated by mapping the MAD respectively the α - trimmed standard deviation on the value of the standard deviation for a standard normal distributed set of values.) The formulas of these moments are summarized in table 3.2.

Variance	$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2$
Standard deviation	$\sigma = \sqrt{\sigma^2}$
Mean of absolute deviations	$\frac{1}{N} \sum_{i=1}^{N} \lvert x_i - \bar{x} \rvert$
Median of absolute deviations (MAD)	$median_{1 \le i \le N} \ (\lvert x_i - median_{1 \le j \le N} \ (x_j) \rvert)$
α - trimmed standard deviation	$s(\alpha) = \sqrt{\frac{1}{N-\lfloor N*\alpha \rfloor - 1} \sum_{i=\lfloor N*\alpha \rfloor + 1}^{N-\lfloor N*\alpha \rfloor} (x_{(i)} - m(\alpha))^2}$

Table 3.2: The moments describing the spread of a given sample.

Higher order moments like the skewness and the kurtosis, that are shown in table 3.3, give hints concerning the shape of the distribution the data values come from. A negative skewness value indicates that the majority of the data values are smaller than their center, while the kurtosis informs about the deviation of the distribution of the data from the shape of the normal distribution. A kurtosis value of 0 means that, the data values are normally distributed. Negative values for the kurtosis indicate that there are less data items in the tails of the distribution compared to normal distribution. A positive kurtosis is a hint for the inverse case.

Skewness	$\left(\frac{1}{N} \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^3\right) / \sigma^3$
Kurtosis	$\left(\frac{1}{N} \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^4\right) / \sigma^4 - 3$

Table 3.3: The moments of higher order.

The α - percentiles [58], that are also called α - quantiles, are measures indicating the maximum value of the lowest $N * \alpha$ data values and thus communicate robust information about the shape of the data distribution. A computation scheme for the α - quantile q_α is given in table 3.4.

α - quantile	$q_\alpha = \begin{cases} \frac{x_{(N*\alpha)} + x_{(N*\alpha+1)}}{2} & \text{if } N * \alpha \text{ is whole-numbered.} \\ x_{(\lfloor N*\alpha \rfloor + 1)} & \text{else.} \end{cases}$

Table 3.4: α - quantiles calculation scheme.

3.2 Correlation and Covariance

The correlation [28] is a normalized measure in the intervall $(-1, 1)$ indicating linear dependencies between two samples with N values x_i and y_i. Three different methods for calculating a correlation coefficient are commonly in use, which are shown in table 3.5. The Pearson correlation r divides the covariance of the two samples by their standard deviations.

Pearson correlation	$r = \frac{\sum_{i=1}^{N} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2} \sqrt{\sum_{i=1}^{N} \left(y_i - \bar{y}\right)^2}}$
Spearman correlation	$r_s = \frac{\sum_{i=1}^{N} \left(R_i - \bar{R}\right)\left(S_i - \bar{S}\right)}{\sqrt{\sum_{i=1}^{N} \left(R_i - \bar{R}\right)^2} \sqrt{\sum_{i=1}^{N} \left(S_i - \bar{S}\right)^2}}$
Kendall correlation	$\tau = \frac{\text{conc.} - \text{disc.}}{\sqrt{\text{conc.} + \text{disc.} + \text{extra-y}} \sqrt{\text{conc.} + \text{disc.} + \text{extra-x}}}$

Table 3.5: Correlation measures

The Spearman correlation r_s has the same calculation scheme but instead of the data values themselves their ranks are used. The rank of a value is its position in the ascending sorted sample. R_i indicates the rank of the i-th value of the values x_i and S_i indicates the rank of the values y_i.

The Kendall correlation examines all $\frac{1}{2}N(N-1)$ pairs of data points (x_i, y_i). Thereby it is decided whether a pair is *concordant* (*conc.*) or *discordant* (*disc*). A pair is *concordant*, if

its x and y values have the same order relation (higher or lower) to each other. A pair is called *discordant*, if the x values have the opposite order relation as the y values. If the x values or the y values are equal an *extra-x-pair* respectively an *extra-y-pair* is counted. The counts of those four categories are used to calculate the Kendall correlation.

Because of the calculation schemes of the Spearman and Kendall correlation, they are also able to detect monotonic functional coherences like logarithmic or exponential dependencies. If only linear relationships should be examined in a robust way, a robust estimation of the covariance matrix can be applied, from which the correlation can be computed.

The covariance also describes the linear relationships between samples and is defined as

$$Covariance: \qquad \sigma_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N}\Big(x_i - \bar{x}\Big)\Big(y_i - \bar{y}\Big).$$

Thus the matrix holds the variances of the samples in the main diagonal and the covariances between the samples in the off diagonal entries. Consequently it is a symmetric matrix.

A robust estimation of the covariance matrix can be achieved by considering a reasonable subset for the covariance calculation that represents the majority of data items of the sample. One way to find this subset is the minimization of the determinant of the covariance matrix [100]. A proof for this theorem is shown in [101]. Certainly a complete search of possible subsets is not feasible especially for large datasets. Therefore the FAST-MCD [101] algorithm was introduced as a heuristic that approximates the best subset.

For a dataset containing N objects the procedure starts with the selection of $h \in (\frac{N+p+1}{2}, N)$ arbitrary p dimensional data items, which are considered to calculate the initial mean vector μ_1 and the initial covariance matrix Σ_1, which serve as location and spread estimates for the data. Those two estimates are used to calculate the Mahalanobis distance for all N data items.

$$MahalanobisDist(x) = (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1)$$

The next subset is formed of those data points having the h smallest Mahalanobis distances. This iterative process is carried on until the determinant of the covariance matrix of two consecutive subsets is equal.

Because this approximation scheme evaluates locally optimal solutions a multi start local search approach is applied on a small fraction of the data. Afterwards the best results are used on the whole dataset. Therefore parameters are introduced steering how many initial subsets are used and how long they are iterated. Furthermore the user can decide how many solutions should be considered for the whole dataset.

The subset creating the covariance matrix with the smallest determinant is applied to calculate the robust center of the data μ_{robust} and the robust covariance matrix Σ_{robust}, describing

the shape of the data cloud. On these robust estimates the calculation of the robust distance is based, which is defined analogous to the Mahalanobis distance.

$$RobustDist(x) = (x - \mu_{robust})^T \Sigma_{robust}^{-1} (x - \mu_{robust})$$

The robustness of the algorithm is steered by setting the size of the subset h. While a value of $h = 1$ yields to the calculation of the classic covariance matrix, a setting of $h = \frac{N+p+1}{2}$ leads to the maximum robustness which allows nearly 50 % outlying values.

3.3 Clustering

Cluster procedures search for groups in the data. For this work the k means [50] as well as the fuzzy k means [17] and a hierarchical clustering approach based on merging are of high importance.

3.3.1 *k* Means Clustering

The user has to specify the number of clusters k that should be created. There are several possibilities to choose the k initial cluster centers: k randomly selected data items or k randomly determined points in the hypervolume enclosing the dataset can be used. Because different initial settings can lead to significantly different results the starting centers already determine the quality of the final partitions. The clustering itself performs two steps that iteratively improve an overall energy function. The first task is to assign each data item to the cluster with the most similar (nearest) cluster center. The second step recomputes all cluster centers by calculating the means of all objects of a cluster.

This update procedure minimizes the sum of distances between the data items and their nearest cluster center. As convergence criterion a minimum update limit for the cluster centers can be set. If no center update exceeds this limit the algorithm stops. Another common stopping criterion is a maximum iteration number, that should be performed [62].

3.3.2 Fuzzy *k* Means Clustering

The fuzzy k means, also known as the fuzzy c-means (FCM) algorithm [17], is the most popular fuzzy clustering algorithm. Similar to the k means clustering the number of clusters k has to be specified. Afterwards the memberships for each data item to each cluster have to be initialized. The membership indicates how strong a data item is associated with a cluster. The membership values are stored in the matrix U, which has a dimensionality of $k \times N$, where N indicates the

number of data items. Typically the membership values u_{ij} lie within the intervall $[0, 1]$ and satisfy the constraints $\sum_{i=1}^{k} u_{ij} = 1$ for all j, meaning that the maximum possible membership of a data item is split up between the clusters.

The fuzzy k means optimizes the following objective function:

$$\sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^{m} \left\| x_j - c_i \right\|^2,$$

where x_j represents the j-th data item and c_i the i-th cluster center. m is the so called fuzzifier that influences the fuzzyness of the cluster result and can be set to values within the interval $(1, \infty)$. A value of m near 1 creates a hard clustering result. The higher the value of m is set, the more fluent are the transitions between clusters. Typically the fuzzifier is set to $m = 2$. To minimize the given objective function two steps are performed iteratively:

1.) The membership values are updated by calculating

$$u_{ij} = \Big(\sum_{r=1}^{k} \frac{\left\| x_j - c_i \right\|^{2/(m-1)}}{\left\| x_j - c_r \right\|^{2/(m-1)}} \Big)^{-1}.$$

2.) The cluster centers are recomputed by calculating the membership weighted mean of the data items according to

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}}.$$

As convergence criterion a minimum update limit for the memberships u_{ij} can be applied. If no membership value exceeds the given limit, the iteration stops. Additionally the maximum number of iterations can be set, to avoid long computation times.

Hierarchical Clustering

A hierarchical clustering that is based on a merging criterion starts by declaring each data item as initial cluster. Afterwards the two most similar clusters are merged to a new cluster. This procedure is iteratively repeated until only one cluster exists. A hierarchical clustering that is based on a splitting criterion starts with one cluster holding all data items and splits the existing clusters into two partitions until each data item lies in its own cluster.

The nested cluster structure that is created by these methods can be represented by a dendrogram, which is similar to a tree structure. Each node represents a cluster and has two

children unless it is a leaf node. Leaf nodes represent single data items. The nodes in a level divide the dataset in clusters. On level 1, which consists only of the root node, there is one cluster representing the whole dataset. On level 2 the dataset is partitioned in two clusters. On level N the dendrogram holds N clusters, each representing a data item of a dataset with N data points.

The most popular hierarchical clustering procedures are variants of the single-link or the complete-link algorithms. The single-link algorithms define the cluster similarity as the minimum of the distances between each pair of data items formed by one data point from each of the clusters that are compared. The problem that arises from this approach is called chaining and means that the cluster shapes can be elongated, if in each iteration an object is added to the cluster that is part of a chain of data items that points into a certain direction in the data space. The complete-link algorithms calculate the cluster similarities as the maximum distance that the previously mentioned data item pairs of the compared clusters have. This yields to more compact clusters [62].

3.4 Principal Component Analysis (PCA)

The principal component analysis [61] introduces a linear transformation of p dimensional data items on new axes, the so called principal components, where the first $m < p$ components hold the majority of the overall variance represented by all p components. Thus the linear transformation of the p original data dimensions $X_1, \ldots, X_p$ to the principal components $Y_1, \ldots, Y_p$ has the properties of:

- falling variances in the principal components: $Var(Y_1) \geq Var(Y_2) \geq \cdots \geq Var(Y_p)$
- preserving the overall variance in the data: $\sum Var(Y_i) = \sum Var(X_i)$
- no correlations between the values mapped on two principal components: $Cor(Y_i, Y_j) = 0$

The transformation matrix A that realizes the mapping from the original dimensions of the data to the principal components holds in its columns the eigenvectors of the covariance matrix Σ of the dataset in descending order of their eigen values. If α_i indicates the eigen vector with the i - highest eigen value the transformation can be written as:

$$\mathbf{Y} = \mathbf{A}^T\mathbf{X} = \begin{pmatrix} \alpha_1^T \\ \alpha_2^T \\ \vdots \\ \alpha_p^T \end{pmatrix} \mathbf{X}$$

3.5 Linear Regression

The multiple linear regression [82] [64] approximates the values y_i of an attribute by finding a linear function in p independent variables and applying it to their values. Thus the approximation of the values $\tilde{y_i}$ is computed by

$$\tilde{y_i} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}.$$

The regression parameters β_i are calculated by

$$\beta = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}^T y,$$

where $\mathbf{X}$ represents the design matrix holding 1 entries in the first column and the values of the p independent variables in the remaining columns. Because each row represents a data item with a leading 1, $\mathbf{X}$ has the dimensionality $N \times (p + 1)$. β represents a vector holding all regression parameters and the vector y contains the corresponding values of the data items in the attribute that should be estimated by the regression.

3.6 Theoretic Distributions and statistical Tests

Theoretic distributions [82] are characterized by their probability density function (pdf), a non-negative function, which has an integration value 1 over the range of $-\infty$ to $+\infty$. Additionally the distributions have different parameters that steer the shape of their pdf. The pdfs of the distributions of importance for this work and their parameters are shown in table 3.6. Each of those functions has parameters that influence the shape of the distribution. The normal and log normal distribution uses μ as setting for the center and σ for steering its spread. As standard settings $\mu = 0$ and $\sigma = 1$ are common. In contrast to that for the uniform distribution the interval limits a and b have to be set, for what the unit interval is taken as standard property. The exponential distribution has the parameter λ steering the magnitude of the decay and the maximum value of the pdf. The setting $\lambda = 1$ is used in this work, if nothing else is specified. The shape of the chi-squared distribution is influenced by the degrees of freedom DF. Furthermore its pdf refers to the gamma function

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

Besides the pdf, the cumulative distribution function (cdf) is helpful for calculating quantiles. The cdf is the integral of the pdf. Because the pdfs of the normal and log normal distribution can not be integrated analytically a numerical integration scheme has to be applied for

Distribution	pdf
Normal distribution	$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Log normal distribution	$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{(ln(x)-\mu)^2}{2\sigma^2}}$
Uniform distribution	$f(x;a,b) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{else.} \end{cases}$
Exponential distribution	$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{else.} \end{cases}$
Chi-squared distribution	$f(x;DF) = \begin{cases} \frac{1}{2^{DF/2}\Gamma(\frac{DF}{2})} x^{\frac{DF}{2}-1} e^{-\frac{x}{2}} & \text{if } x \geq 0. \\ 0 & \text{else.} \end{cases}$

Table 3.6: Probability functions of distributions used for this work.

the calculation of quantiles and distribution values. The pdfs of the remaining distributions are summarized in table 3.7, where the corresponding function of the chi-squared distribution refers to the incomplete gamma function

$$P(a,x) = \frac{1}{\Gamma(x)} \int_0^x e^{-t} t^{a-1} dt, \text{ for } a > 0.$$

Distribution	cdf
Uniform distribution	$F(x;a,b) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{else.} \end{cases}$
Exponential distribution	$F(x;a,b) = \begin{cases} 0 & \text{if } x < 0 \\ 1 - e^{-\lambda x} & \text{else.} \end{cases}$
Chi-squared distribution	$F(x;DF) = \begin{cases} 0 & \text{if } x < 0 \\ P(DF/2, x/2) & \text{else.} \end{cases}$

Table 3.7: Density functions of distributions used for this work.

To realize hypothesis testings that validate if a set of values comes from a theoretic distribution, the Kolmogorov-Smirnov test [29] has been implemented. Therefore an empirical cumulative distribution function is created of the sample holding for each value x the number of data values that are smaller or equal than x divided by N, which indicates the overall number of objects in the sample. This function is compared to the pdf of the given theoretic distribution. The largest absolute difference on a position x is stored as Kolmogorov-Smirnov statistic. Finally the significance for the test statistic is evaluated by calculating the Kolmogorov-Smirnov probability function. If this significance value exceeds a user defined limit (limits of 0.05 are common), the null hypothesis stating that the sample comes from the given theoretic distribution is kept.

This approach can also be used to test if two sets of values show the same distribution. Therefore the empirical cumulative distribution functions of those samples are compared.

Chapter 4

Integrating Statistical Functionality in Visualization

As previously outlined visualization techniques and statistical routines have similar objectives, but try to reach them using different means. This section discusses those issues by the tasks of group finding in high dimensional data and grouping of dimensions as well as for the detection of outlying values. For this purpose the strengths and weaknesses of the two disciplines are discussed theoretically and illustrated by concrete examples. The artificial datasets for these demonstrations are created with R [5] [31]. The visualizations are realized - if not differently stated - with a Java [6] application that uses the results of the functionality from the statistics library.

4.1 Statistical Techniques

Computational routines can fulfil a large number of calculations to filter information of special interest. Thus they are able to perform general purpose computations like the detection of the main trends or make numerical summaries of dimensions available. Nevertheless they lack the fast and immediate adaptation to the current dataset, which may not apply to the constraints that a statistical approach asks for, or which shows structures that may distort the results of an applied algorithm. The following paragraphs analyse these issues by selected tasks in the data mining process.

4.1.1 Grouping of Data Items

In data mining applications clustering techniques play a central role and are a very popular method for finding the essential information that is hidden in large high dimensional datasets.

Thus a huge variety of clustering routines has been developed and adapted to handle specific problems. Hence a clustering routine tuned for a certain task can introduce a meaningful partition of the data. But if there is no knowledge about the data and a general clustering procedure is used, no statement about the quality of the resulting clusters can be made. This issue can easily be discussed on the basis of the k means clustering algorithm, which is one of the most popular group finding heuristics.

The principal task that the user has to do before starting a k means procedure is to specify the number of groups k that should be considered. This is the first and already crucial step for obtaining a meaningful partitioning of the data. For this task in general a solution can only be found by try and error approaches or by experience values, if no information about the data is present in advance of the data mining process. The next drawback of the algorithm is that it converges to a local optimum, which means that the found solution can be far worse than the best possible k means clustering. To measure the quality of the found partitions the value of the k means objective function can be considered. But even several starts of k means clustering and the choice for the best solution can not guarantee a certain minimum quality. Furthermore the found clusters may not be suitable for the given dataset, because k means clustering creates in general spherical partitions, which may not represent the groups in the data properly.

As this example based on the k means approach shows, there are a number of uncertainties in applying a clustering procedure on a dataset. Thus the usage of those routines assumes the knowledge about the shape of the created clusters and the weaknesses of the used algorithm. Even clustering frameworks like gCluto [97], which provide a set of basic cluster routines with different objective functions and similarity measures, can not guarantee an optimal partitioning, even though standard settings and a clear graphical user interface differentiating between power users and users looking for fast results is provided. Certainly there are statistical methods like the Bootstrap Clustering [72] to test the reliability of clustering results by inspecting, if small changes in the data would yield to a significantly different partitioning. But this is no guarantee that a stable solution is also a reasonable grouping of data items. Also the field of cluster validity [48] [49] concentrates on the evaluation of the number of clusters and the settings for a clustering routine that are the best to fit the structure of the underlying data. However these methods require high computational effort and the interpretation of the results demands knowledge about the validation techniques.

4.1.2 Grouping of Dimensions and Feature Subset Selection

To partition a set of attributes into groups it is important to investigate the relationship between the dimensions. The correlation is a popular and easy to interpret mean of statistics that analyses the coherence between variables. It indicates linear consistencies in the values of the compared attributes and thus can be used as a similarity measure between dimensions. The use of the classic correlation can be strongly influenced by outlying values. Hence the computation of robust correlation coefficients, which are resistant against the impact of extreme values, is recommended. The comparison between the robust and the classic measures can also indicate, whether there are outlying data values, which do not correspond to the domain of the majority of the data.

To demonstrate the differences between the classic Pearson correlation and the two robust estimates according to Kendall and Spearman two examples are shown in figure 4.1. The first dataset consists of 500 bivariate standard normal distributed data items and 10 data items with the same properties but shifted from the origin. While the classic correlation already indicates a strong relationship between the dimensions, the robust coefficients state correctly that the attributes are not correlated. Furthermore the Spearman and Kendall correlation are able to indicate non linear dependencies between variables. This is shown in the second example, where the dimension Y holds the exponential values of X. Consequently the robust measures indicate perfect correlation. The Pearson coefficient also predicts a very strong dependency, but not that accurately. The reason for this is that the robust measures only consider the ranks of the dimension values for the calculation, which is also the reason for their robustness. If there is the need for a robust correlation coefficient that only detects linear correspondence patterns, a robust estimate of the covariance matrix can be considered. As the covariance matrix contains the variances of the attribute values and their covariances it provides important measures that roughly describe the shape of the high dimensional data cloud. But both estimates can also be used to calculate the correlations between the dimension pairs.

But although the comparison between robust and classic correlation measures can reveal the presence of outliers, this information can not help to identify or exclude those extreme values from further calculations. Also the detection of local correlation patterns can not be realized in this way.

As the correlation coefficients can be seen as a similarity measure for dimensions a hierarchical clustering based on the correlation matrix as starting point instead of a distance matrix between data items, is practical for the identification of groups in a set of dimensions. Therefore the correlation matrix holds the correlation coefficients for all possible dimension pairs. The hierarchical clustering provides a dendrogram structure as result. This allows the user to select the number of groups in the dimensions. As feedback concerning the quality of the groups the minimum absolute value of correlations between the dimensions within a cluster is available.

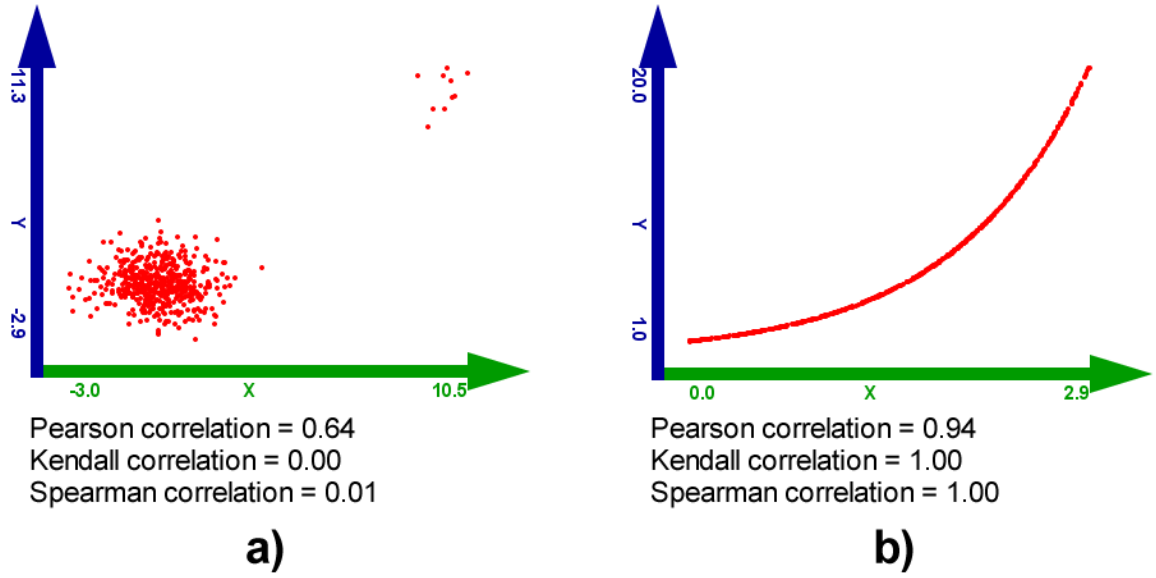


Figure 4.1: Two examples showing the differences between classic and robust correlation measures in the presence of outliers (a) and non linear functional dependencies (b).

The higher these minima are the more similar are the attributes in a cluster. Also for this clustering procedure robust correlation computations are recommended. Certainly the fact that there are groups of data items having the same correlation pattern is neglected. This yields to the result that the most dominant group determines the dimension grouping or that equally strong represented groups cancel out their correlations.

A discussion of the drawbacks of a clustering of dimensions soon leads again to the disadvantages of the automated group finding itself mentioned in the previous section. Additionally the choice of the measure indicating the similarity between dimensions poses another challenge. While the dissimilarity for the clustering of data items can be intuitively defined by a distance measure, the measure for the attribute relationship should be chosen according to the needs of the user. The definition of the similarity strongly depends on the aim of the investigation and on the properties of the data. Should only linear or also non-linear relationships be detected? Should significantly large groups in the data already indicate correlations? And finally can any correlation measure introduce a reasonable grouping of the dimensions? Besides the correlation also other similarity measures can be defined, which is discussed by Ankerst et al [12].

Thus the main drawback of this grouping approach is that an interactive visual inspection of the introduced similarities is missing, if only the statistical technique is used. Consequently a fully automated feature subset selection choosing one attribute per introduced dimension group suffers the previously mentioned problems. Additional uncertainty arises by the heuristic that selects the variables. The first question to answer is how many groups achieve the best match with the structure of the dimension relationship and consequently how many dimensions have

to be selected? Also a choice for a variable per cluster has to be made, which strongly depends on a measure of interest defined by the user. Examples would be the dimension holding the highest variance of its values or the dimension which could be considered as the center attribute of a cluster. However a feature subset detected by this heuristic needs further inspection of the relationship and manipulation of the intermediate as well as the final result by the user. Therefore a visual exploration assisted by numerical facts computed by statistical routines is a necessity.

Besides of this feature subset selection based on a dimension grouping, in the field of machine learning heuristics have been proposed to choose certain attributes of the data for unsupervised learning procedures like clustering. Those routines are in general involved with higher computational costs and the introduction of a measure that indicates the quality of a chosen dimension subset. This measure can be based on the solutions of a certain process like a clustering or an outlier detection. There it is examined if the subset of dimensions produces the same or even better results with respect to the given task as the use of all attributes. But subsets are selected for these specific tasks and can not be seen as the optimum subset for a general purpose data exploration. Furthermore the heuristics introduced to avoid the inspection of all possible 2^p feature subsets of a p dimensional dataset, calculate a result that is a local optimum, which - like a clustering solution - may be far worse than the best possible subset of attributes.

4.1.3 Dimension Reduction

Dimension reduction algorithms can not be used for the identification of attribute groups. They actually suffer from losing the reference to the original attributes of the data. PCA and projection pursuits create linear combinations of the dimensions, in which the data items are defined. For a data dimensionality of 10 or more it is hard to understand which effects a data attribute has on a principal component for example. Furthermore there are no parameters for simple methods like the PCA that steer its performance. If a projection of the data on two dimensions should be achieved, but the first two principal components only explain a low fraction of the variance in the data, the user must choose more principal components to capture the majority of the information. But nevertheless the PCA provides with the explained variance a hint for the quality of the dimension reduction.

More complicated is the user's task for SOMs and MDS operations. The fact that any linear and non-linear mapping of the data on a low dimensional subspace is possible has tremendous drawbacks. In general no statement about the quality of a given mapping can be made. While the PCA provides with the explained variance a guide to which extend the spread of the data is captured, the results of other routines have to be examined and explored to identify how stable and reliable they are. There is also no linear function that combines attributes to new dimensions. So in general there is no reference at all to the original data dimensions. Further-

more the SOM algorithm and its huge number of adaptations provide a vast variety of settings that can not be easily accommodated to a dataset to achieve a reasonable dimension reduction result. The same applies to implementations of an MDS algorithm. For example spring models that approximate the dissimilarity of data items in the p dimensional data space in a low dimensional subspace introduce a huge number of settings concerning the convergence of the model and the properties of the springs.

SOMs are of special interest for the visual analysis of multivariate data because its algorithm incorporates a mapping from p variate data space to two dimensions. Thus visualizations can be easily achieved and results can be presented. Nevertheless the two dimensional representation of the dataset does not depict the data items. The visualization is built on the model vectors to which data points can be assigned. Thus interactive exploration of the data may be difficult and not that intuitive because of the introduced abstraction. On the other hand SOMs also provide an implicit clustering of the dataset, because the model vectors try to represent the data and agglomerate, where the density of data items is high. But this fact is not conveyed because in the two dimensional mapping the grid positions are evenly spaced. But as the SOM is an unsupervised learning algorithm like the clustering it suffers the same disadvantages. Furthermore there is no objective function that can be optimised to compare mapping results, and even the consequences of the setting of the parameters on the mapping process are not obvious.

4.1.4 Outlier detection

For this discussion the outlier detection based on distributions, on distances to neighbouring objects and on the density is considered. Those principal techniques have in common that they provide parameters that the user can (possibly interactively) change to examine their behaviour on the underlying dataset and thus steer the number of detected outliers. Nevertheless those changes are difficult to interpret, if only a numerical output is provided. Certainly the user can examine the detected data items and compare them to items that are considered to be near the center of clusters or the whole dataset itself, but the possibilities of interactive visualization applications would enrich these exploration tasks significantly, because besides the visual validation of the results the impact of parameter changes can be investigated.

The distribution-based techniques provide both a statistical characterization of one dimensional outliers as well as the detection of high dimensional data items that do not fit to the trend of the majority of the data. A one dimensional outlier detection starts with the robust estimate of the center and the scatter of the sample. By the quantile of the assumed distribution of the dimension values the user steers the number of detected outlying values. The disadvantage of this approach is that the theoretic distribution used for the quantile calculation has to match approximately the distribution the sample comes from. The statistical routine itself also does not investigate, if there are gaps in the data, which would indicate the limits of groups or the de-

cision boundary for the classification of outliers and non-outliers. Furthermore a visualization of the one dimensional data by histograms enriched with interaction techniques would provide a more efficient outlier detection tool as those statistical approaches, because the human visual system is able to identify outliers easily in a space of a dimensionality up to 3.

In contrast to that the statistical outlier detection for high dimensional data has the advantage, that there is no visual competitor, that allows an easy identification of outlying data items by the user. The distribution-based approach computes the robust distances according to an estimate of the distribution model of the p dimensional data items and classifies data points as outliers if their robust distance is higher than a user defined quantile of appropriate distribution. If the data is assumed to be p dimensional normal distributed, a quantile of the chi-squared distribution with p degrees of freedom is used. This approach guarantees the detection of p variate outliers, instead of the classification of outlying objects in a low dimensional subspace as long as the data nearly corresponds to the multivariate normal distribution. By using the robust distance a one dimensional measure for the outlyingness is introduced. Hence a visualization can help to decide, whether the detected outliers are significantly different than the majority of the data, by showing gaps in the distance values, because the problem is again reduced to a one dimensional outlier detection application. Especially the interactive modification of the quantile values would introduce a tremendous improvement for the efficient outlier identification.

The drawback of this approach is that the data has to approximately show a p dimensional elliptic distribution. Other distributions do not provide such an elaborate theoretical background as the normal distribution does. Consequently it can be a problem to apply other distribution models on this approach. Furthermore it may be cumbersome to transform an arbitrary dataset so that it applies to this assumed distribution. Also groups in the data do not allow useful results. Thus the main disadvantage of this approach is that the data has to satisfy certain constraints.

Density-based and distance-based approaches do not assume distribution properties of the data. Therefore they rank the outlyingness of a data item according to the number and the proximity of its neighbouring objects. The main disadvantage of those algorithms is the tuning of the parameters, which can not be intuitively made. These settings also have to be adapted for each dataset, because other density properties and the different number of data items make a reuse of parameters not possible. Thus a visual feedback could certainly help to verify, if reasonable parameter values are used.

Furthermore the higher the dimensionality of the data the more doubtful the calculations of the k nearest neighbours are, because the magnitude of the distances becomes higher and differences are represented by least significant places of a floating point number. Thus the setting of the parameters becomes more difficult because, small changes in the values can lead to tremendous changes in the outlier detection result. In this case a preparation of the data by dimension reduction or feature subset selection has to be performed.

4.2 Interactive Visual Data Analysis

Information visualization applies the extraordinary human pattern recognition skills to explore data. Combined with the possibilities of interactive updates and modifications of the visualizations and the linking of different views, which allow insight to certain aspects of the multivariate data, the visual data analysis became a very powerful and important tool in the field of data mining. But as the user of such applications is not used to think in higher dimensional spaces, this concept has also shortcomings. These issues are discussed in this section for the finding of groups in the data and dimensions as well as for the identification of outliers.

4.2.1 Grouping of Data Items

The search for groups in multivariate datasets by information visualization applications demands the use of linked views that present different aspects of the data. As example one view could depict all (relevant) dimensions of the dataset, while the second view helps to identify low dimensional patterns like the correlation between dimensions. Thus the combination of a parallel coordinate technique illustrating all variables of the data and a scatterplot visualization for a more intuitive investigation of structures can be very efficient. Furthermore the possibility to select data items has to be possible in all used views. The selection performed in one view should be visually propagated. This is usually achieved by drawing selected data items in a certain colour.

To make use of the fact that a visualization technique provides an insight to a specific aspect of the data, also the selection possibilities have to contribute to this issue by allowing the user to work with these aspects. This can be explained for example by considering the parallel coordinates. Of course selections can be drawn on the axes so that only those data items are marked showing data values within the selection intervals of each attribute. But the parallel coordinates also show the correlation patterns between neighbouring variables very well. To capture this aspect by a selection type the angular brushing [53] has been introduced. With angular brushing those data items are selected that have a line segment featuring an angle within a specified angular range. Thus data points having the same correlation properties can be highlighted.

But these adaptations do not only affect selection techniques. Also visualization specific interaction capabilities can improve the visual group finding process. For this purpose parallel coordinates provide a reordering of displayed attributes, which allows the alignment of the variables so that similar dimensions are placed near to each other. This significantly improves the detection patterns spanning several dimensions. Furthermore flipping of axes can be realized so that the minimum is shown on top of the screen space. This operation visually inverts correlation patterns, which allows displaying positive and negative correlated dimension pairs

similarly. Thus linear functional dependencies between attributes are illustrated in only one way. As this example for the parallel coordinate view shows, each visualization technique has its own capabilities to facilitate the cluster detection.

The detection of a group in the data could be achieved by iteratively drawing selections according to visual patterns like gaps or peaks in the distribution of the dimension values. By creating several selections on the axes of the parallel coordinate view a multidimensional cluster can be defined. How stable the cluster is, can be verified by shifting or resizing single selections. If there are tremendous changes introduced by one of these operations, then a cluster can be considered as arbitrarily defined and thus not justified by the structure of the data. The use of angular brushes in contrast to that could select data items that have the same correlation patterns within any depicted pair of dimensions. This may lead to different clusters satisfying constraints that are not based on the dimension value ranges. Of course also combinations of these selection techniques can be applied. But as this variety of possibilities to determine a cluster in parallel coordinates grows exponentially with the number of dimensions, a multitude of solutions exist. To find those partitions that fit the real structure of the data is by far not trivial.

To introduce also an uncertainty aspect of the selection for data items, where it is not sure, whether they belong to the detected group, fuzzy selections [43] were proposed. This allows an extension of the selection concept, where not only the states *selected* and *not-selected* are possible [34]. To achieve this a degree of interest (DOI) is set for each data item having values in the interval $[0, 1]$, where 0 indicates not selected objects, 1 marks data points of special interest for the user and values between express the degree of uncertainty.

Nevertheless the interactive selection of data items has the main drawback, that only one or two dimensional patterns are used for selections. Thus a cluster having multivariate properties can not be detected by investigating low dimensional aspects of the data. The fact that there is no clue for the quality of the selected group, make the detection of multivariate clusters via interaction techniques nearly impossible. Certainly different aspects of the data expressed by the visualizations can be examined to assess the reasonableness of groups, but this assessment depends on the user's skills concerning visual data mining and his/her knowledge about the data. Furthermore the data analysis of several experts may create slightly different solutions. So this categorization by a visualization tool also suffers the same problems as the use of different cluster algorithms: Results may be similar but not exactly the same.

A further disadvantage is that the creation of a detected group can not be reproduced. Only a protocol, capturing all interaction processes, allows the traceability of forming the clusters. But such an approach can only record the actions taken by the user. There are no semantics explaining, why a selection has been drawn with a given interval and so forth.

From the viewpoint of the user a well implemented interface allows a pleasant work with selections and linked views. But if the dimensionality of the data is high, it will be cumbersome to work with high dimensional visualization techniques such as parallel coordinates. Also the fact that many attributes have to be considered for the selection process causes the user to invest time and concentration to keep track of the actions done so far. The distinction between important dimensions and not interesting variables becomes more crucial and can not be easily achieved visually. Again knowledge about the data or experience values have to be applied to choose the essential attributes.

Also the placing of similar dimensions near to each other in views like the parallel coordinates is important to improve the detection of structures and trends in the data. But as the number of possible orders grows exponentially with the dimensionality, it becomes cumbersome to rearrange the dimensions of a high dimensional dataset manually. A heuristic should be applied to accomplish an initial solution. But not only the placing of dimensions in views is concerned with this issue also the order of the usage in techniques such as worlds-within-worlds or dimensional stacking highly influences the pattern perception. Also pixel bar charts, which use attributes for splitting, ordering and colouring purposes, can reveal certain trends by assigning the variables to a certain operation. But it may be laborious to figure out the best dimension assignment.

4.2.2 Grouping of Dimensions and Feature Subset Selection

As discussed in section 4.1.2 the correlation is a significant measure to observe similarities between dimensions. Information visualization provides techniques that allow the interactive investigation of correlation patterns between attributes. For the exploration of these relationships it is not important, if the majority of the data items are correlated. Also subgroups showing the same functional dependency between two attributes can be detected and thus also a clustering of data points can be introduced based on these patterns as outlined in the previous section. But although those local correlations can be detected in views like scatterplots or parallel coordinates, where even tools like the angular brushes encourage exploring these aspects of the data, a numerical feedback validating a significant coherence between the selected objects on the dimensions of interest is missing.

To group the dimensions of a dataset, those investigations can be made for each pair of dimensions. But as the dimensionality grows this approach becomes inefficient. Even the presentation of all tuples of attributes in one view like a scatterplot matrix does not allow the fast identification of groups in the variables unless the correlation patterns are very clear, what is in general not the case.

Apart from the challenges of the grouping process also the selection of representative attributes per dimension cluster can not be realized easily with visualization tools. The easiest way is to use the variables showing the highest variance of their values. But this task should be automated and not be achieved visually by examining the value ranges of the dimensions.

4.2.3 Outlier detection

The detection of one-, two- or three-dimensional outliers can be achieved easily with the means of visualization. Histograms as well as 2D and 3D scatterplots allow the human visual system to identify data items that seem to have different behaviour than the majority of the data. Gaps between objects as well as significant displacements against the main extent of the point cloud are indications for outlying values.

But as the dimensionality of multivariate datasets does not allow a depiction of all attributes of the original data items in a scatterplot, the intuitive investigation of the three dimensional space can no longer be applied. Like the detection of high dimensional groups in the data the search for multivariate outliers needs a huge variety of visualization and brushing techniques. But in contrast to the task of finding clusters the search for correlation patterns and groups in dimension values is now focused on a very small subgroup of data. Thus it is easier to define the overall behaviour of the majority of the data and invert the selection to detect outliers, because besides the high dimensionality the fact that outliers are heterogeneous in general makes this task difficult.

New developments in the visualization of high dimensional data try to emphasize outlying values so that they are not overlooked caused by the clutter introduced by dominant patterns. But those methods often identify high dimensional outliers by examining extreme values of the variables separately. A robust estimate of the location and the scatter of the values of the dimensions is used to identify one dimensional outliers. But a simple example with data items defined in two attributes shows already that a one dimensional outlier does not identify outlying values in the data space. In figure 4.2 an example is shown by a two dimensional scatterplot and a parallel coordinate view. The scatterplot reveals that there are two groups of data, an elongated diagonal group holding the majority of data items and indicating a correlation between the two dimensions and a spherical point cloud deviating from the major trend. Furthermore solid lines indicate the medians of the attribute values. The dashed lines represent the decision boundaries for the one dimensional outlier detection, which uses 1.5 times the inter quartile range added respectively subtracted from the median. It is shown that this approach detects the outlying values of the major point cloud. But the small group that is shifted from the center of the data is not identified. Nevertheless both the objects marked as outliers as well as the not detected group should be considered as special data points, that may be created because of other reasons than the remainder of the data. This shows that the group of outlying values can be heterogeneous,

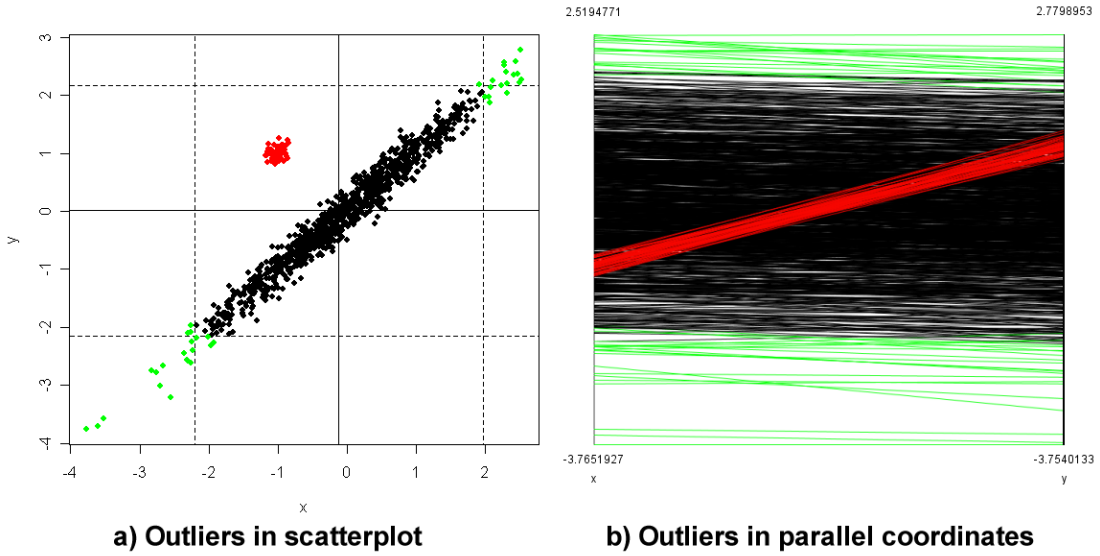


Figure 4.2: A two dimensional dataset demonstrating the failure of one dimensional outlier detection for the identification of high dimensional extreme values is visualized with a 2D scatterplot and a parallel coordinates view. The objects at the margin of the large elongated group are detected as outliers (green), while a small group shifted from the center of the data (red) is not identified.

because there are at least three subgroups showing different behaviour. The parallel coordinate plot shows the same data and stresses the attention to the different correlation behaviour of the groups. Consequently this example demonstrates the failure of the outlier detection per dimension and how two visualization techniques present aspects of the data in different ways.

The example indicates the importance of correlation patterns. But they are not the solution for multivariate outlier detection. The correlation does not consider gaps in the data. Thus if there are separated groups having the same behaviour between dimensions, but different values in the attributes, a selection based on similar correlation would highlight both groups as one. Consequently the selection approach to detect outliers can be successful for a certain type of outlying values, but it is not guaranteed to identify high dimensional extreme values that have no significant pattern in one attribute or a two dimensional subspace. Furthermore the detection of two dimensional features like correlation strongly depends on the ordering of the attributes in the visualization. While scatterplot matrices provide all pairs of dimensions in the data, parallel coordinates establish for each variable two neighbouring attributes. Thus the arrangement of the visualized dimensions plays a crucial part for finding groups and outliers. Depending on which attributes are set in relation to each other, different correlation patterns would be considered for those tasks.

4.3 Integration

In this section examples for the collaboration between the information visualization methods and statistical routines are discussed. The aim is to show possibilities how drawbacks of a technique of one of both fields can be redeemed by a procedure that was introduced by the other science. Of course the proposed combinations are not able to overcome all challenges posed by the analysis of multivariate patterns in the data or by the shortcomings of the used techniques, but improvements for the visual data mining process are achieved. But as the data has to satisfy certain constraints for the majority of the multivariate statistical procedures the usefulness of transformations as well as their integration into an information visualization application are outlined.

4.3.1 Data Preparation

As mentioned before distribution-based multivariate outlier detection assumes that the data nearly applies to a theoretical distribution, while other procedures such as clustering require the dimensions to have the same range of values. Clustering approaches are especially sensitive to the scale of values per dimension, because they mostly base their similarity calculations on distance measures like the Euclidean distance. This implies that if one attribute shows significantly higher value differences between the objects, these relations strongly dominate the results of the distance computations and thus the clustering solution.

This issue is demonstrated in figure 4.3, where a dataset containing four groups of data items defined in the attributes X and Y is used for a k means clustering. The left plot shows that the attribute X has a strong influence on the clustering, because of its high values in comparison to the other variable. Thus the groups are defined as sections on the dimension X. The right visualization illustrates that the group finding process on the attribute values transformed to the unit interval is successful.

But also simple linear scalings to the interval [0, 1] can fail, if there are extreme values in a given dimension. Thus these deviating items are mapped near to a limit of the value range, while the remainder of the data values are projected towards the other limit. The main information that remains in the transformed variable values can be seen as a binary decision, if a data item is an outlier in this dimension or not. This issue can have negative impact on multivariate procedures such as clustering or the principal component analysis. As alternative the robust z standardization could be applied to all attributes, which maps the actual data at approximately the same value range. Extreme one dimensional outliers still have major influence, but the essential information of this dimension is not compressed to a small interval of values.

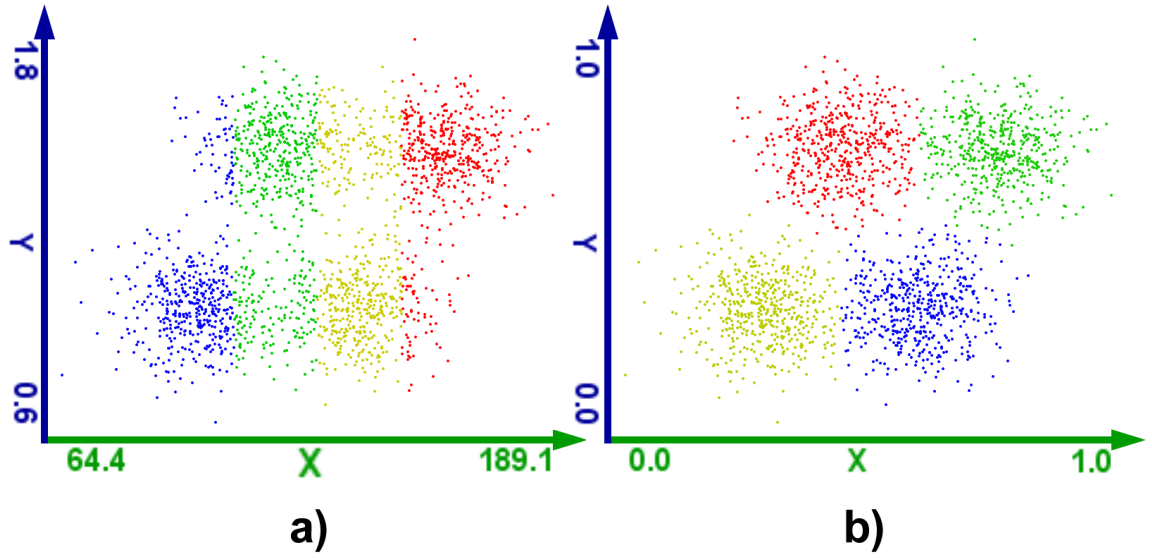


Figure 4.3: The impact of different value ranges of attributes on k means clustering algorithm is demonstrated based on a two dimensional dataset showing four groups.

An example for this issue is shown in figure 4.4. A dataset containing four normal distributed groups defined in two dimensions with the centers at the positions $(0, 5)$, $(4, 0)$, $(7, 15)$ and $(10, 500)$ is shown in the left scatterplot. After applying a mapping on the unit interval for each attribute the principal components, which are shown in figure 4.4 (b), were computed. The evaluated directions are not along the actual spread of the data, which is created by the extreme values of one group in the variable $V2$ and by the shifted positions of the groups along the dimension $V1$. By using the robust z standardization as transformation for the data, the PCA computes these important directions correctly, as depicted in figure 4.4 (c). The impact of the applied transformations that should compensate extreme values on the multivariate statistical methods has to be considered and tested. The wrong usage of these mappings can create misleading results.

4.3.2 Grouping of Data Items

For the detection of groups of data items the combination of a clustering procedure with the possibilities of linked visualization views and their selection techniques obviously can achieve benefits for the analysis of data. The clustering algorithm can partition the dataset automatically before the user tries to identify groups visually. This division of the data into clusters is a more efficient and comprehensible starting point for the visual analysis. The advantage of clustering routines is that they perform their calculations in the data space and thus really introduce high dimensional groups. The following interactions in the information visualization applications can aim to explore the clustering result by modifying the views and analysing features of the

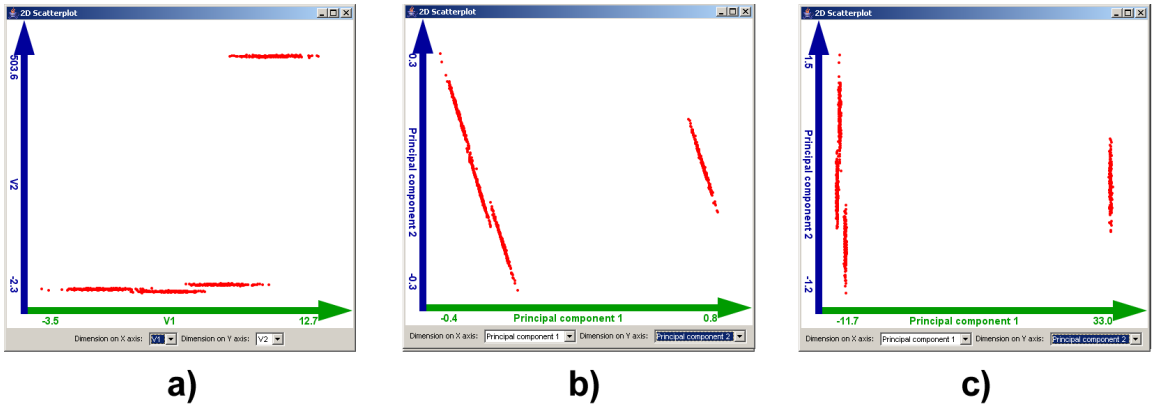


Figure 4.4: This example demonstrates the impact of transformations of variables on the PCA. The left scatterplot shows the original two dimensional data. The visualization in the center illustrates the first and second principal component computed from the variables scaled to the unit interval. The final scatterplot on the right depicts the first two principal components computed from the robust z standardized attributes.

clusters. But it is also possible to investigate the validity of the partitions by modifying cluster centers or by excluding dimensions and starting a reclustering. Especially if a certain cluster algorithm is taken into account the possibilities for interactions are manifold but also strongly depend on the used grouping approach. The following example discusses intuitive manipulations of results of the popular k means clustering algorithm. Thus they can not be applied in this way on a hierarchical clustering heuristic, which would alternatively provide a dendrogram structure that allows different interactive modifications.

The k means approach introduces a hard clustering, where each data item is assigned to one cluster. This allows the visualization of each group by assigning the same colour to its members. Additionally the cluster centers, which are the representatives for the clusters and thus used for interaction techniques, can be accentuated. By examining the cluster result in a view, that depicts all dimensions of the dataset, such as the parallel coordinates, those dimensions can be identified, which have major influence on the clustering procedure. These attributes show a rather clear separation of the clusters. The reason for the disproportional impact on the clustering can be caused by wrong transformations, that yield to different value ranges for the dimensions, or that there is a group of dimensions explaining the same information and thus bias the cluster process. But it can also indicate that those attributes explain the groups in the data very well. To find the exact explanation for this issue the semantics of the dimensions have to be examined.

To manipulate the cluster result a variety of possibilities can be provided. In the first place the cluster centers can be modified. They can be moved to a new position or fixed, so that a follow-up clustering can not modify their positions. Clusters can also be fixed, so that further

procedures are not able to modify the membership of the cluster, or deleted, which causes the assignment of its members to the partition with the nearest center. Furthermore clusters can be merged to one group, which involves the calculation of a new cluster center. But also subdivisions of existing clusters can be initiated by either applying a clustering algorithm on its members, or by splitting them along a dimension, which introduces two new clusters having the same cluster center coordinates in all attributes except for the splitting variable. The members of the split cluster are assigned to one of the new created groups, by identifying the nearest cluster center.

After those modifications have been performed, a new clustering can be started, that is based on these settings, which are represented by the number of the cluster centers and their positions. These techniques can be used to create partitions of higher quality with respect to the energy function of the k means approach. If a modification results in a worse clustering, an undo function can repair this mistake. Besides this issue the user can also verify how stable the introduced clusters are. If small changes yield to significantly different partitions, this may indicate that the used clustering algorithm does not apply well on the underlying dataset or that there are no significant high dimensional groups. Furthermore the mentioned interaction techniques allow the accommodation of the clustering result to the user's visual impression of the groups. The clusters can be modified so that they cover visual groups in the dataset by repositioning the centers and assigning the data items to the group with the nearest center. But also a hierarchical group structure can be introduced by manually starting subclustering procedures for already created clusters.

The interactive collaboration between clusterings and the user's modifications also allows a more efficient exploration of the relationships between the attributes. Conclusions of the behaviour of the clustering procedure yield to a better understanding. An example for this issue is the snapping back of repositioned cluster centers. This can be caused by a large field of attraction of a locally optimal partitioning solution or by the fact that the moving operations were only fulfilled based on low dimensional features, that are not significant for a clustering in data space.

To investigate the quality of a clustering and the shape of the introduced groups, an intuitive visualization like the scatterplot is needed. Thus a projection of the multivariate data on two or three dimensions has to be accomplished by dimension reduction techniques or feature subset selection heuristics. Although the illustration of the data in this visualization can not capture the whole high dimensional information, it is still a hint, if the clustering introduces meaningful partitions. Furthermore high dimensional groups that separate from other data items by gaps can be identified, and a verification, if a single cluster covers these deviating objects is easy to perform. On the other hand this approach is also a hint for the quality of a dimension reduction technique, because the user can see, whether the most important information is cap-

tured by the mapping, if the introduced cluster borders are sharp. Thus if members of different clusters overlay in the visualization, aspects of the data that were considered for the clustering, were not incorporated in the dimension reduction result.

Based on an intuitive low dimensional representation of the data, the outlined interaction operations can be executed. In the case of the manipulation of cluster center positions, the alterations of their position in the visualization have to be projected back into the data space. This allows the inspection of high dimensional changes in the partitions and facilitates the comprehension of multivariate cluster results. Furthermore the interactive actions are nearly performed in data space and are no longer based on low dimensional patterns.

Certainly the use of dimension reduction introduces an intermediate layer that has to be well understood and must produce stable and high quality results. Thus the PCA as the simplest dimension reduction technique is recommended to allow the comprehension of introduced modification. But even more important is the fact that the directions of the principal components of a dataset are always the same, yielding to the same mapping, while other algorithms like SOM and MDS can create different projections if a rerun is initiated. But the use of the PCA involves also the fact that the variance in the data must be captured well in the first three principal components. This issue or the alternative - namely the consideration of only a few attributes - for either the clustering or the dimension reduction is a tremendous limitation.

But also a different combination of clustering routines and information visualization is possible. Partitioning can be based on a subset of data items that has been selected in advance. This allows the user to examine the high dimensional behaviour of data points that fulfil a certain constraint like similar correlation patterns. It is also possible to introduce initial groups in the data by the means of interaction techniques and start a clustering based on these settings. This can be seen as a validation of the introduced partitioning as well as an interactive definition of a starting condition for the clustering, which - depending on the used cluster algorithm - can be crucial for the quality of the cluster solution. If a fuzzy clustering is considered, also the smooth selection concept [43] could be applied.

In this sense a cluster can also be interpreted as a selection. Thus manually drawn selections and computed groups in the data could be used alternatively for the information drill down process to identify data items of special interest. Consequently both techniques can build up on the result of previous steps, which would represent an efficient collaboration between statistical methods and information visualization techniques.

The major drawback of these combinations is the non-existing traceability of the determined clusters, because either the clustering result has been manually modified or the interactions that took place to initiate a clustering routine based on predefined partitioning can not be easily translated into a set of parameters for the used algorithm. Thus it is a necessity to capture the interactions and the initiated clusterings by a protocol that allows tracing back how partitions have been introduced and where possible mistakes have been made.

Besides the integration of the clustering in the visual data mining process, it can also introduce an abstraction of the data. Hence a clustering can be used to eliminate noise in the high dimensional data space, that originates by fluctuations in measurements or by the nature of the data itself. The determination of a large number of clusters, each representing only hundreds of data items, can eliminate cluttering in visualizations and reveal the main structures in the dataset.

Concerning the detection of significant patterns in the visual exploration a fuzzy clustering approach would allow even more possibilities to enhance a visualization. Because of the fact that each cluster holds a membership value for each data item, this information can be used for the colour and opacity of drawn objects. That approach can reduce clutter in the visualization and enhance the perception of the distribution of the clusters. The location as well as the main behaviour of the members of a cluster is stressed, because only data items near the cluster center are drawn with full opacity. This is demonstrated by the example depicted in figure 4.5, where the dataset UVW [1] is visualized by a parallel coordinates view using different colour settings for the 149769 data items. In the first visualization each data point is drawn with the colour red and full opacity. The second illustration is based on the result of a fuzzy k means clustering, where 6 clusters were created. Therefore the maximum membership value of a data item is mapped to the opacity value of a data item. Its colour is a linear combination of the cluster colours weighted by the membership values. This results in the accentuation of the main patterns around the cluster centers, showing significant correlation properties between and variance issues on the dimensions.

4.3.3 Relationship between Dimensions and Dimension Grouping

Statistical approaches allow a meaningful grouping of dimensions by measuring the relationships between attributes. But a definition of similarity between dimensions focuses only on one aspect to indicate how closely variables are related to each other. It strongly depends on the needs of the user or the given task that implies a certain similarity measure. Furthermore there exists the trade-off between local and global similarity. It is clear that visualizations can give a more detailed insight into the relationships between two attributes. Interaction techniques also allow an exploration of patterns that indicate similarities. But because of the high dimensionality of the data it is not possible to examine each pair of dimensions visually. These aspects

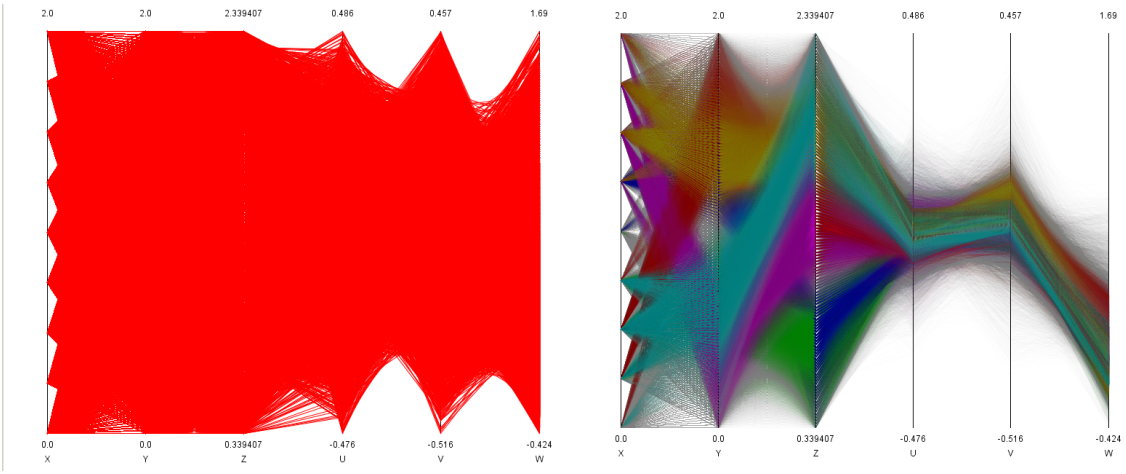


Figure 4.5: The left parallel coordinates plot shows the six dimensional UVW dataset [1] containing 149769 data items, while the right illustration reveals patterns by incorporating the information of a fuzzy k means clustering using 6 clusters.

justify the combination of the statistical approach and the interactive exploration by information visualization techniques.

Once a computational similarity measure for dimensions is defined respectively selected by the user, a clustering can introduce groups of dimensions. But these groups can only be seen as an initial partitioning that has to be examined. Too many uncertainties accompany this approach, so that the interactive modification of the clustering result is crucial. Thus a fast examination of visualizations of the dimension pairs should be possible. The scatterplot matrix is an efficient technique that allows this. But as the patterns of interest have to be detected for each scatterplot, which is time consuming aside from the case that there are significant correlations in the data, what can not be assumed, statistical approaches can ease this process.

Firstly it is possible to integrate the dimension clustering information into the scatterplot matrix visualization. Scatterplots depicting attributes of the same cluster could be coloured according to their cluster membership. Thus a categorization between visualizations showing inter and intra cluster relationships can be easily made. Furthermore the similarity measure for the dimension pairs can be shown per scatterplot as well as integrated into to the colouring properties of the visualized data items. As a further indicator for the similarity between dimensions a smooth regression technique such as LOWESS [26] could be applied. This approach calculates linear regressions for partitions of the data and accumulates them to a smooth curve. Thus the introduced curve identifies linear correlations by straight lines and deviations from this pattern by significant curvatures. The effectiveness of this indication is demonstrated in figure 4.6, where a scatterplot matrix uses a LOWESS curve and a linear regression to emphasize the functional coherence between the illustrated variables. LOWESS has the additional

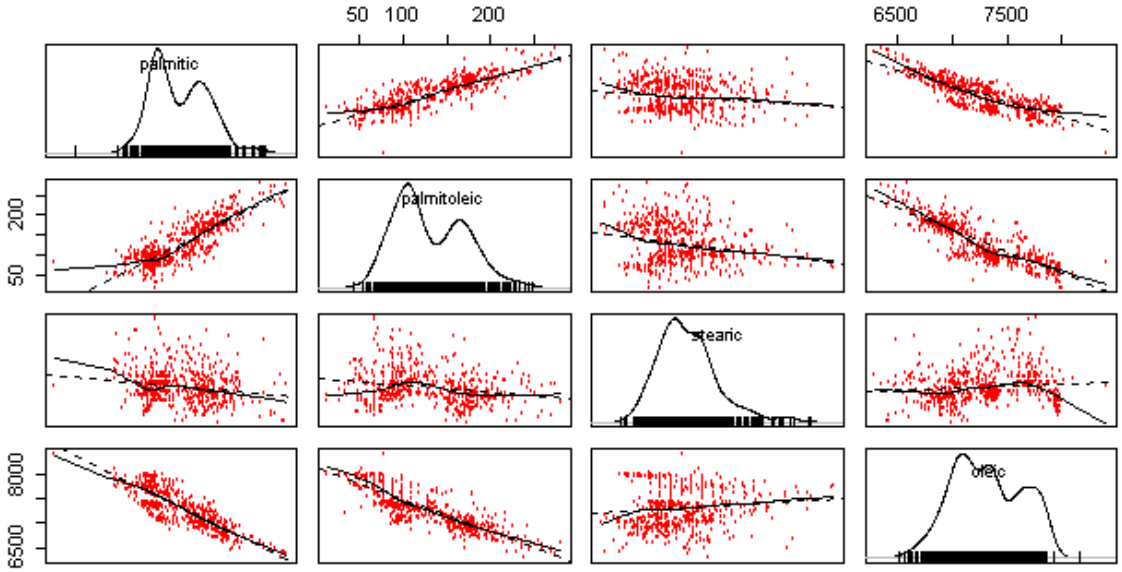


Figure 4.6: A section of a scatterplot matrix illustrating the olives dataset. To enhance the detection of correlation patterns a LOWESS (solid curve) and a linear regression (dotted line) is visualized. This view was created with R [5]

advantage that it can be easily applied on subsets of the data and thus can also serve for the identification of local similarity patterns.

As interaction technique the user could select dimension pairs by clicking on a scatterplot. This would provide a simple mean for the interactive modification of the dimension clustering. For example selected dimension pairs can be assigned to a new or an already existing cluster. For a better visualization of the dimension group structure an alternative linked view showing a dendrogram should be used. Interactive cluster hierarchy visualizations such as InterRing [122] could also be applied to manipulate hierarchical cluster structures. But this visualization can not replace the discussed concept of the scatterplot matrix, which allows the examination of dimension similarities.

This concept can be enhanced by visualizing a scatterplot matrix using only the attributes of a selected cluster. Additionally the mean dimension can serve as cluster center, and in the main diagonal of the scatterplot matrix each dimension is plotted against this center of the variable group. This would allow an analysis of the deviation and the dissimilarity of attributes to their cluster center. Additionally a categorization of dimensions that are at the boundary of the group and thus maybe part of a different cluster can be made.

If the user wants to examine the similarity between two dimensions in further detail, selection techniques can be combined with numerical feedback of computed similarity measures based on a scatterplot visualization showing the attributes in question. Especially for the

exploration of local relationships between two attributes it is a necessity to interactively highlight the data items that indicate this pattern. Additionally the similarity measure in use can be recomputed based on the currently selected subset of data points. This allows the numerical verification of a visually detected feature of the dimension pair. The use of correlation measures can even create a more accurate tool. Because of the fact that values of classic and robust correlation measures diverge in the presence of outliers, both calculations can be applied to interactively identify data items that contradict the local or global correlation pattern. Linked with the changes of selections the correlation measures have to be recomputed and finally show nearly identical values, if no outlying object is highlighted. Certainly this procedure can not be applied to a large number of dimension pairs, but it may be useful to decide, whether a dimension at an attribute group boundary can be assigned to the cluster in question.

4.3.4 Outlier detection

As multivariate outlier detection is difficult to accomplish by the means of information visualization, because of the possibly heterogeneous behaviour of the outlying items that should be identified, a statistical routine could establish an initial solution for this task. Afterwards visualizations combined with interaction techniques provide efficient means to examine the computed result and to steer the provided parameters that decide how many data items are classified as outliers. It is important that immediate visual feedback is given for the parameter tuning. Otherwise no advantage in comparison to a static post visualization of the outlier detection result is achieved. The possibilities of interactive linked views that provide insight into different aspects of the multivariate data also allows a manipulation of the results of the algorithm, which is independent from the parameter settings. For example data items that are misclassified by the statistical functionality could be manually selected and assigned to the correct group. Furthermore a details-on-demand approach could help to investigate, why these data items were categorized in this way, by showing the computed degree of outlyingness or the facts on which the automatic classification was based. This would also provide hints for the improvement of the parameter settings or possible future executions of the statistical algorithm.

If the current data mining task is not primarily concerned with the identification of outlying objects, also an on-the-fly integration of outlier detection information in the interactive visual exploration can be accomplished. In the case that the user visually detects data items that seem to deviate from the main behaviour of a group of objects, it is crucial that facts that can be computed by the multivariate outlier detection algorithm can be shown immediately. The reason for this is that the user's decisions are mostly based on low dimensional features, while the statistical routines consider the dimensionality of the data space.

Certainly also a dimension reduction technique can be applied to map the data items to a two or three dimensional space, so that they can be visualized by scatterplots. This approach al-

lows the fast visual detection of outlying objects, but therefore the dimension reduction process must capture the most important information in the data, as outlined in section 4.3.2, where the task of clustering was discussed. Especially because of the uncertainty that a dimension reduction approach captures the important information for the outlier detection task, an incorporation of the information of the outlier identification algorithm into the dimension reduction should be considered. In the scope of this work this was accomplished by using the robust covariance matrix for the robust distance calculation, on which the outlier detection is based, and for the principal component analysis, which is applied for the dimension reduction.

In contrast to the finding of data items with similar behaviour, outliers are in general a heterogeneous group of objects. Thus detected outliers in a view showing a low dimensional projection of the data items need not be located in a specific area, but should be positioned at the border of data agglomerations. If this is not the case several issues have to be considered. The dimension reduction technique may not capture the necessary amount of information to visualize the actual relations between the data items or the outlier detection algorithm may not be suitable for the given data respectively its parameters are not correctly set. Especially for such tasks it is important to interactively manipulate the parameters of the statistical algorithms, which provide an immediate feedback, to efficiently determine the reason for such discrepancies.

As in the field of statistics post visualizations have been developed to validate and inspect the results of the outlier detection algorithms, it is crucial to adapt those views in visual data mining applications and enhance them by adding interaction techniques. The interactive manipulation of the parameters of the algorithms is the first step that has to be accomplished. Furthermore the user can be allowed to define fuzzy decision boundaries to incorporate uncertainties in the classification process. Depending on which statistical routine is used to detect outliers, this additional functionality can only influence the interface and the final storage of the classification result or it also has to be considered in the computational back end of the application. But also the drawing of selections and manual modifications of the outlier detection result could be used to automatically evaluate more accurate parameter settings, which means that the user's interactions have to be translated into properties of the used statistical method. If this handshake between the user's input and the capabilities of the computational routines can be achieved, the quality of outlier detections will be tremendously increased.

Chapter 5

Library for statistical Functionality for Visualization

For the implementation of a library containing basic statistical functionality for information visualization applications routines have to be determined that can assist the visual data mining process and moreover enable the combination of user interaction and computational tasks. Because there is a huge range of statistical methods the examination of open source as well as commercial programs and the research in recent papers documenting the collaboration between visualization and statistics helped to state five categories of statistical functionalities that apply to this criterion.

Spotfire [10] [7], one of the leading commercial programs in the field of information visualization, offers hierarchical clustering methods and the k means algorithm to partition a dataset. Furthermore Self-organizing Maps and Principal Component Analysis can be applied to reduce the dimensionality of the data. Statistical measures like the median, the arithmetic mean, the variance as well as quartiles and so called outside values, describing whether a data value is a one dimensional outlier, can be calculated for selected data items. Boxplots and QQ-Plots for the comparison between the distribution of a sample and a theoretic distribution are provided. Additionally decision trees [80] and methods for analysis of variance (ANOVA) [98] enrich the functionality of Spotfire.

The statistical functionality of the commercial application Miner 3D [4] also concentrates on the dimension reduction via PCA and on clustering using the k means algorithm. Additionally a set of statistical moments can be evaluated to characterize the behaviour of selected data items.

Likewise the open source project GGobi [112] [2] provides moments calculation for objects that are currently selected. As dimension reduction procedures the PCA and MDS can be used. To reduce the number of data items to speed up further calculations or to allow clearer

visualizations a subsampling algorithm can be applied. But GGobi is also popular because of its possibility to apply a set of transformations on the data items. The user interface even allows the definition of multiple transformation operations, which improves the data preparation significantly.

The majority of the publications concerned with the combination of statistical routines with interactive visualizations concentrates on the topics clustering and dimension reduction, where especially the self-organizing maps were discussed. As outlined in section 2.3 the proposed applications mainly concentrated on the use of statistical routines as starting point of a visual data exploration.

In this section the chosen functionality that is incorporated in the statistics library is listed and its usefulness is discussed. Afterwards the differences between the availability of statistical routines for arbitrary applications and the implemented statistics library for information visualization are explained by focussing on interaction possibilities that are necessary to effectively combine both fields. Finally first approaches for semi-automatic sense making are explained, where the information visualization uses statistical methods to filter concrete facts from the data.

5.1 Components of the Library

Based on the outlined research concerning useful statistical routines for information visualization techniques the functionality that is provided by the statistics library can be divided into five categories. The first one is concerned with operations that are performed on each dimension of the data separately. This concerns the preparation of the data for multidimensional operations by applying transformations as well as analysing the properties of their distributions by calculating statistical moments. Transformations are included in the library because data preparation is a crucial step for procedures like clustering or outlier detection. This is also the reason why GGobi provides a very efficient user interface for applying such operations on data attributes. In comparison to that statistical moments are standard computations, which are supported by the main software products for visual data mining.

For the investigation of patterns between dimensions the calculation of correlation measures and the covariance matrix is realized in the statistics library. While correlation coefficients can give a hint for the similarity and the coherence between attributes, the covariance matrix is a crucial component of a variety of statistical applications. The probably most popular routine using this functionality is the principal component analysis.

As group finding is one of the most important data mining tasks, the integration of clustering procedures into the statistics library is a must. The clustering can be seen as the reduction of data items by calculating cluster centers representing large numbers of data points. To de-

crease the number of attributes of the data a dimension reduction can be applied, which captures the main information of the data space in low dimensional subspace.

As the majority of statistical routines requires data showing a certain distribution pattern, hypothesis testing has been implemented. Thus it can be inspected, if dimension values apply to a given theoretic distribution. Additionally for each incorporated theoretic distribution random values, distribution and density values as well as quantiles can be calculated. This functionality is necessary to use those theoretic distributions to create interactive statistical analysis by means of visualization.

Finally the multiple linear regression as a simple model that tries to explain the functional relationship between p independent variables and one dependent attribute was implemented. The reason for this is that regression is one of the basic approaches to predict the behaviour of a variable. Thus it can be used to simplify and explain the coherence between a subset of dimensions and another data attribute.

5.1.1 Transformations and Moments

Transformations can be used to map the values of a dimension into a given range or to change the shape of the distribution of the attribute. For multivariate methods such as clustering the range of dimension values is of high importance. If a clustering is based on a dimension x with values between 0 and 1 and on dimension y with values between 0 and 1000 the clustering concentrates on dimension y, because the high values of this dimension have a stronger influence on the distance calculations that measure the dissimilarity between data items. To run a clustering, where each dimension is treated equally, both dimensions have to be mapped on the same domain. To achieve this a linear scaling to the $[0, 1]$ interval is sufficient. Alternatively a classic z standardization $((x - \mu)/\sigma$ can be used, where μ represents the arithmetic mean and σ represents the standard deviation.

But these two linear transformations do not consider outlying values. Already one extreme value can make the result of those transformations unusable for further operations, because the extreme value is mapped on one end of the interval and the ”actual” data values are projected to the opposite interval limit. Thus the main information stored in the values of the attribute is lost and a multivariate routine only observes the categorical decision telling, if a value is an outlier in the given dimension. To avoid these effects a robust z standardization can be applied, where μ represents the median and σ represents the median of absolute deviations (MAD). Alternatively the outlying values can be extracted by a one dimensional outlier detection. Afterwards the ”actual” data values can be scaled by the non robust transformations described above.

Besides the linear mappings for changing the domain of the dimension values logarithmic and squareroot transformations are used to alter the distribution of the values. Statistical routines often assume a certain theoretical distribution like the normal distribution. The non linear transformations can help to convert the distribution of the given values so that it is similar to an assumed theoretical distribution.

Moments are parameters that characterize a sample or the values of a dimension. They can be divided into estimators of location, estimators of scatter, percentiles and higher moments. Parameters of location are the arithmetic mean, the median and the α-trimmed mean. Because of being influenced by even one outlying value, the arithmetic mean is often replaced by the median, which is the most robust estimate of the center of a population. The disadvantage of the median is that it is only based on at most two observations. Thus it is no efficient parameter estimation that indicates the real location of the sample center for a rising number of data items in contrast to the arithmetic mean. To deal with the trade-off between robustness and efficiency the α-trimmed mean can be applied. By setting the parameter α the robustness is steered because the $\frac{\alpha}{2}$ lowest and the $\frac{\alpha}{2}$ highest values are discarded. The remaining $1 - \alpha$ values are considered for the computation of the arithmetic mean. This approach allows the calculation of the center by using a certain percentage of the data to maintain efficiency to a certain degree and rejects outlying values.

Similar observations can be made for the moments of scatter. The variance and the standard deviation are classic estimates that are based on all given values. Thus they have a high efficiency, but are not robust. They are also the most popular measures to characterize the spread of the values of a sample around its center. The MAD representing the median of the absolute deviations of the values from their median is the most robust estimate for the scatter. The α-trimmed standard deviation allows the user to weigh the importance of robustness and efficiency.

The α-percentile or the α-quantile is defined as the value that is higher or the same as the α fraction of the sample values. Important quantiles are the so called first, second and third quartile (ie. the 0.25-, 0.5- and 0.75-quantile). The second quartile is equal to the median. The difference between the third and the first quartile is also known as the inter quartile range (IQR), which can also be used as a robust measure of scatter. Assuming normal distribution the standard deviation is equal to $\frac{IQR}{1.349}$. The three quartiles are also used for drawing a boxplot [113], which is a popular illustration for the distribution of a set of values.

Higher moments are the skewness and the kurtosis. They describe certain shapes or properties of distributions. The skewness indicates a right-skewed distribution by values higher than 0, meaning that the majority of values is higher than the center of the distribution. A negative skewness characterizes a left-skewed distribution. The kurtosis measures the weights of the tails of a distribution. Values higher than zero indicate that the given distribution has

more values in the tails as the normal distribution. Negative values imply fewer values in the tails and a sharper peak than the normal distribution.

5.1.2 Correlations and Covariances

In data analysis the detection of correlations between dimensions is of high importance, because they indicate the linear or monotone functional coherence between the values of two attributes. For the calculation of a correlation coefficient a classic method and two robust methods are implemented in the statistics library. The classic correlation also known as the Pearson correlation can be influenced by outliers and measures only the linear coherence between two variables. The robust correlation coefficients according to Spearman and Kendall reduce the influence of outlying values significantly [8]. They also detect non-linear correlation patterns that may be created by logarithmic or exponential dependencies.

The calculation of correlations can also be used to detect similarities between dimensions. The value of correlation coefficients is within the interval $[-1, 1]$, where -1 indicates a complete negative and 1 a complete positive correlation. A negative correlation means that the higher the value of a data item in the first dimension is, the lower is its value in the second dimension, while positive correlations imply the inverse. A perfect correlation is given if the drawing of the data points in a scatterplot results in a straight line. A correlation of 0 implies two uncorrelated dimensions [95].

Thus the correlation can be used to group dimensions. A group should contain attributes which are significantly correlated to each other. This criterion is fulfilled if the absolute correlation coefficients are near 1. To find those groups automatically a hierarchical clustering approach can be used. Therefore each dimension is set as initial cluster. Afterwards the two most similar clusters are merged iteratively until only one cluster holding all dimensions remains. The most similar clusters are defined by those two clusters that have the highest absolute correlation coefficient to each other. After the merge the correlation of the new cluster to the remaining ones has to be renewed. This is accomplished by keeping the lowest absolute correlation coefficient of a dimension that is part of the new cluster. That approach assures that the clusters separate from each other and that no chaining effect appears. The hierarchical clustering introduces a dendrogram structure that can be used to adjust the number of dimension groups. Ideally the number of groups is chosen in a way, that the minimum correlation between two attributes in a group exceeds a certain level.

But the correlation can also be applied to rank two dimensional scatterplot visualizations, so that plots that show interesting patterns for the user are automatically detected. This approach gains importance, if high dimensional datasets are analysed, where the number of possible scatterplot visualization does not allow a manual search for interesting combinations of attributes.

The Rank-by-Feature framework [107], which is also discussed in 2.3.1, demonstrates the effectiveness of this procedure. While this framework uses the Pearson correlation besides other non robust measures, the application of Spearman and Kendall correlation coefficients would also detect views that contain linear patterns, which are masked by outliers, as well as other monotonic functional coherences.

A different measurement for the relation between dimensions is provided by the covariance matrix that holds the variances of the dimensions in the main diagonal and the covariances between the attributes in the off-diagonal entries. Thus it is a symmetric matrix, that describes a p dimensional dataset as a hyperellipsoid approximating the shape of the data in the p dimensional space. This interpretation makes the covariance matrix important for multivariate methods such as dimension reduction or for examining the relation between dimension groups. In the library the Principal Component Analysis (PCA) and the calculation of the Mahalanobis distance build up on the computation of the covariance matrix. The PCA uses the covariance matrix to find the directions with the highest variance in the data. An elaborate discussion of the use of the PCA can be found in section 5.1.3. The Mahalanobis distance considers the distribution of the data for its distance values by projecting the data items to a space where each dimension has variance 1 and all covariances are 0. This is achieved by the inverse covariance matrix.

The classic estimation of the covariance matrix is based on all data items and thus can be influenced by multidimensional outliers that distort the hyperellipsoidal shape described by the covariance matrix. Hence a robust estimate of the covariance matrix according to the fast estimation of the Minimum Covariance Determinant (MCD) algorithm [101] was implemented. Because the calculation of the covariance matrix is based on a subset of data items representing the majority of the data, groups of outlying data points do not affect the variance and covariance estimations. This robust covariance matrix can be used for a robust PCA or the calculation of the robust distance. Thus the robust PCA calculates independent from outlying values the direction with the maximum variances and hence can be used to project the data items on the principal components, which visualize the outlying values separated from the intrinsic data.

The use of a robust covariance matrix for the Mahalanobis distance yields in the calculation of the so called robust distance, where p dimensional outliers have high distance values. This allows the outlier detection in the multidimensional space. Assumption for the correctness of this approach is that the data has an approximately p dimensional elliptic distribution. To assure this condition, transformations could be applied on the dimensions.

5.1.3 Clustering and Dimension Reduction

Cluster operations try to partition the dataset into groups. This is used to detect trends, meaning that clusters characterize a big amount of data items with the same properties. Thus the cluster centers can be viewed as representatives for data points, so that their values and characteristics can be interpreted more easily.

Besides its usefulness concerning interpretation and simplification of complex datasets a clustering can also serve as filter for noise in the data. To fulfil this functionality a large amount (eg. $N/500$) of cluster centers is defined. These centers represent only a view hundreds of data items after the clustering procedure. But they can be used as abstraction of the data. This abstraction was not introduced for partitioning reasons but for the elimination of tiny disturbances, that may be introduced by errors of measurement or that lies in the nature of the data itself. That approach can also be seen as a reduction of the data points for visualization issues or to fasten future computations. A prominent example, where cluster information is used to replace the illustration of each data item, is the hierarchical parallel coordinates [41] approach.

The most popular clustering algorithm is the k means clustering, that is also provided by the statistics library, because of its simple concept and its status as THE standard cluster approach in data mining applications. k means clustering is an optimization procedure that minimizes the sum of distances of the data items to their cluster center, by starting with randomly chosen initial centers. Thus different initializations reach other local minima of this energy function. Consequently it is recommended to start several clusterings and choose the best result measured by the lowest energy function value as final solution. But this approach can not be applied on large datasets because of too high computational efforts to allow an interactive cooperation between the clustering and visualization. For this reason the statistics library provides the possibility to find a better initial center setting by running several k means clusterings on a small subset of the data. The centers of the best solution are taken as initial centers for the clustering on the whole dataset.

For the calculation of the new updated cluster centers traditionally the arithmetic mean of the data items of each cluster is used. Alternatively the median per dimension can be applied to evaluate the new centers. This has the advantage that outlying data items can not attract the centers. But because the componentwise median is no convex combination of the data items of a cluster, it is not guaranteed, that the evaluated centers lie in the convex hull of their data points. This is shown by the example described in table 5.1. The data items span a triangle in three dimensional space that is not passing through the origin of the coordinate system. Nevertheless the multidimensional median computed from those data points is located at this origin.

Furthermore the distance calculations can be made by using the Euclidean distance or the Manhattan distance. Another option is the setting of weights for the used dimensions to steer their influence on the clustering. This allows the user to differentiate between very important,

Dimensions	x	y	z
Item 1	1	0	0
Item 2	0	1	0
Item 3	0	0	1
Median	0	0	0

Table 5.1: An example showing that a median per dimension has not to lie within the convex hull of its data items.

helpful and not necessary attributes for the clustering. The weights can be set arbitrarily between 0 and 1, where 1 indicates the maximum possible influence.

Additionally to the hard clustering with k means the statistics library provides the fuzzy k means clustering, where a data item is assigned to each cluster with a certain percentage. The usage of a fuzzy clustering has the advantage that grey area - realms where data items can not be assigned to one cluster exclusively - are detected. Thus an adequat visualization can help to assess, if the number of clusters was chosen correctly or if the data can be grouped easily. A high number of and/or large grey areas can indicate that a too small number of clusters was set or that the clustered data appears to be a connected point cloud, where a partitioning is introduced highhandedly. A further advantage of the fuzzyness is that the user is not tempted to take the assignment of data items as given fact, especially if a data item is situated far away from its cluster center. The memberships of the data items thus also indicate the certainty of the affiliation of data points to their clusters. For a hard clustering this fact can be expressed by the distances of the data items to their cluster centers, but there is no indication to which cluster the data item can be assigned alternatively unless the distances to all centers for each data item are kept.

Also for the fuzzy clustering a precalculation of the cluster centers is possible by performing a hard k means clustering on a small subset. It is also possible to set weights for the dimensions and to decide, whether the distance calculation should be accomplished with the Euclidean or the Manhattan distance. Because of the more complex calculation of the cluster centers no option is provided, where the user can change the center calculation scheme.

While the clustering calculates a small number of representatives for tens or hundreds of thousands data items, the dimension reduction tries to depict the information represented in the p dimensional dataset with a few attributes. For the visualization the number of attributes is commonly bounded by two or three. The statistics library therefore provides the Principal Component Analysis (PCA), which computes directions where the highest variance in the point cloud representing the dataset is detected. These directions that are linear combination of the original dimensions are called principal components. The first principal component represents the direction with the highest variance, while the orthogonal second principal component is

the channel with second highest variance. This fact makes it possible that the majority of the information of the data is represented by a few principal components, while the rest only describes details. Thus the principal component analysis can be used to project the data into a low dimensional space, where the main information is represented. The number of dimensions of this space can be interactively chosen by plotting the explained variance of the first m principal components. Usually this graph shows a sharp drop of the additionally explained variance by adding the i-th principal component. Accordingly $i-1$ principal components are recommended to use for the dimension reduction.

Because the calculation of the principal components is based on the covariance matrix of the data, the statistics library provides besides the classic PCA also a robust version. For the latter a robust estimate of the covariance matrix is used. This allows the calculation of directions that are not influenced by outlying values. A projection using the robust principal components can help to identify groups of data items that highly distinguish from the majority of the data.

5.1.4 Distributions and statistical Tests

Theoretic distributions are used to test if a set of data values applies to a given distribution. The normal distribution as well as the uniform and the exponential distribution are of high importance for this task, because they are distributions that describe shapes that are common for samples. To check if the values of a dimension come from a given theoretic distribution, the statistics library provides test routines. The user can apply a significance level and thus decide how conservative the test result should be. The null hypothesis states that the given data values are from the theoretic distribution, for which they are tested. If the calculated p-value is higher than the user defined significance level, that has a standard setting of 0.05, the null hypothesis is kept.

These tests are important, to decide, whether a statistical routine that assumes the data to have a certain distribution can be applied. As described in 5.1.2 the calculation of the robust distance assumes an elliptical p dimensional distribution. Thus the tests could be used to investigate, whether the values of each dimension apply to a normal distribution. But also as a ranking criterion to automatically find attributes of special interest, tests can be applied that indicate to which extend the attributes apply to a given theoretic distribution. The significance values could be used as a ranking criterion, which may be the first step for a feature subset selection approach.

Furthermore the statistics library provides a test to examine, if the values of two samples come from the same distribution. This test is useful to analyze pairs of dimensions. If two data attributes have the same distribution, this might indicate correlations or dependencies that can be validated by calculating a correlation coefficient or fitting a regression model.

Theoretic distributions can also help to visualize, whether a sample or the values of a dimension are distributed accordingly. Therefore a two dimensional scatterplot is used, where the quantiles of the theoretic distribution are plotted on the x axis and the ascendingly ordered data values of a dimension are mapped on the y axis. Now a straight line is fitted through data points representing the first and the third quartile. If the data points nearly lie on a straight line, the data values come from a set having the shape of the given theoretic distribution.

Besides the theoretic distributions used for the tests of samples, the chi-squared distribution is provided. The quantiles of this distribution can be used as decision boundary for a multidimensional outlier detection. Because the robust distances of a p dimensional dataset are assumed to come from a chi-squared distribution with p degrees of freedom. Consequently the 0.975 quantile of this chi-squared distribution identifies a limit, which would detect 2.5 % of the data part of the p dimensional elliptic distribution as outlying data points. By setting the percentage defining how many outliers are expected, an outlier detection application can be interactively steered.

5.1.5 Linear Regression

The multiple linear regression is a procedure, that tries to predict the value of an attribute depending on the values of q independent variables. The function used for the prediction is linear in the independent variables. The advantages of this approach are that the dependencies of the predicted variable are stated by the created function and the simple model allows a fast and intuitive interpretation of the influence of the independent variables. Because of the restriction to allow only linear terms the regression can be computed very fast also for a large amount of data. For visualization issues the linearity introduces a q dimensional plane that can be easily incorporated in scatterplot or parallel coordinate views. The main disadvantage of the linear regression is that the model can not capture functions with higher order terms, which can yield to wrong results, if there are coherences between the variables that are of high complexity. The second issue is that already one outlying value can strongly influence the model and thus produces a significant deviation from the correct result, which would be achieved without the outlier.

5.2 Hooks of Interaction

To integrate statistical routines in interactive information visualization the statistics library needs to allow interactive recalculations and fast updates of the results of statistical methods. To realize this close cooperation additional parameters have to be passed to the routines or even new methods have to be provided. These additional extensions of the interface are the so called hooks of interaction.

In information visualization applications the use of selections is a basic functionality for highlighting certain data items to interactively explore the dataset. Those selections can be modified, deleted or combined with other selections by using logical operations like *AND*, *OR* respectively *NOT*. Because the calculations of a statistical routine can be based on selected data items, additionally to the parameters of the routine itself the subset of the data points to use must be indicated. Therefore methods of the statistics library have a degree of interest (DOI) parameter. This parameter holds a flag for each data item, indicating if it is selected by the user. If a statistical routine allows the specification of a smooth selection, where graduations between *selected* and *not selected* are possible, the DOI parameter holds a value between 0 and 1 for each data item, where 1 indicates a full selection.

Further important aspects in the field of information visualization are focus and context concepts, where details of the data can be examined while the connection to the overview of the data is kept. Therefore the statistics library can be used to provide on the fly statistics for the data, on which the user focuses. Furthermore overall estimates would allow a comparison between the data items of interest and the main behaviour in the data and thus gives a hint of the placement of those objects in the dataset. Parameters for location and spread are especially adequate for this approach. But also correlation or covariance patterns could be considered. Applications that realize this statistical feedback concept can use the corresponding calculations by passing a DOI parameter defining the subset of data items, on which the user focuses.

Besides the general interaction parameters new possibilities for the communication with statistical routines can be introduced. Methods like the clustering can be adapted in a way that the visualization becomes an interactive user interface steering the parameters of the clustering algorithm. For example the representation of cluster centers in information visualization views can be used to apply merge, division or deletion operations on the clusters. Also a subclustering algorithm can be started for partitioning the selected cluster. The cluster centers could be moved and used as initial centers for a new clustering. Each of these operations has its own method in the library, which translates the applied modification into new parameter settings for a clustering algorithm, which is executed afterwards. Because the interactive collaboration between statistical functionality and visual representations is not well established yet, such operations have to be introduced for each algorithm separately. The statistics library is prepared to intro-

duce an intermediate layer for each provided routine that allows the translation of interaction operations into settings for an algorithm.

5.3 Concepts for semi-automatic Sense Making

In this section useful visualization applications based on statistical routines are introduced that allow both fields to contribute to an efficient data mining workflow. The outlined tools have been implemented and a case study, where their work is demonstrated, is discussed in section 6.

5.3.1 Transformations

As the aim of transformations is to apply a function on the data so that their value range applies to a certain range or their structure is similar to a theoretic distribution, a simple visualization in a scatterplot creates a visual control system. As illustrated in figure 5.1 the quantiles of a given theoretic distribution and the ascending ordered transformed dimension values are mapped on the two axis of a two dimensional scatterplot. Additionally a line is drawn that passes through data items that represent the first and the third quartiles of both samples. If the data items are depicted nearby the line, the transformation was successful and the data values apply to the given theoretic distribution. If there are deviations, different transformations have to be applied.

For this application the user can specify per dimension the squareroot, logarithmic, standardization and unit interval transformations. As theoretic distributions the uniform and the normal distribution are supported. This tool should be considered before a multidimensional operation like clustering or dimension reduction is applied, because it allows the preparation of the data to the needs of statistical routines.

5.3.2 Outlier Detection

As multivariate outlier detection the MCD algorithm for the robust covariance matrix estimation is applied. Based on this outlier resistant description of the shape of the data, the robust distance is calculated to provide a degree of outlyingness for each data item. The result of this approach is depicted by scatterplots illustrating the first and the second principal components computed from the robust covariance matrix estimation. To data items that are detected as outliers a colour is assigned. Alternatively those data points can be highlighted, that are used to calculate the robust covariance matrix. This approach provides a visual feedback of the quality of the multivariate outlier detection. If outliers are mainly detected at the border of the depicted point cloud, then it can be assumed that the algorithm was successful. An alternative indication for this issue is, if the data items, on which the MCD covariance matrix is based, agglomerate

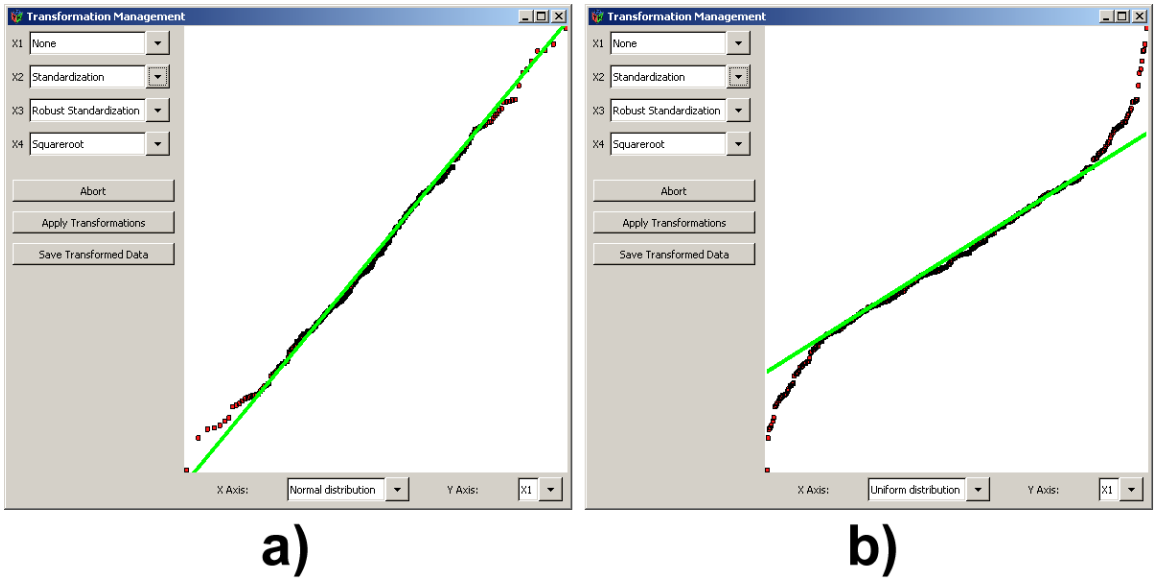


Figure 5.1: Two screenshots of the transformation tool showing a normal distributed attribute compared to the theoretic normal distribution (a) and to the theoretic uniform distribution (b). No transformation has yet been applied to the shown variable.

near the center of the illustrated data. Certainly this observation can only be made, if the used principal components explain the majority of variance in the data.

To illustrate the calculated outlyingness an additional scattplot showing the Mahalanobis distances versus the robust distances has been realized. This visualization is shown in figure 5.2 and incorporates the decision boundaries that classify data points into outliers and non-outliers by a horizontal line for the robust measures that are mapped on the y axis and by a vertical line for the Mahalanobis distances. These lines divide the visualization space into 4 areas. The lower left area contains only those data items showing low robust and low Mahalanobis distances. Thus they are identified as the actual data. The lower right area contains data points with high Mahalanobis distances. Consequently using the classic distance measure they would have been classified as outliers, but the robust distance reveals, that those objects are non-outliers. In general this area is empty as the robust distance that is not influenced by outliers shows higher distance values. The upper left area contains masked outliers, because they are only detected by the robust distance, while the upper right quadrant shows data items with high deviations from the robust as well as from the classic center of data. Additionally the identity line is drawn. If the data items are illustrated nearly along this line, then the data comes from multivariate normal distribution.

The user can now steer the number of the p dimensional data items that are detected as outliers, by a slider holding the quantile of the chi-squared distribution with p degrees of freedom. If the data is multivariate normal distributed, the robust as well as the Mahalanobis

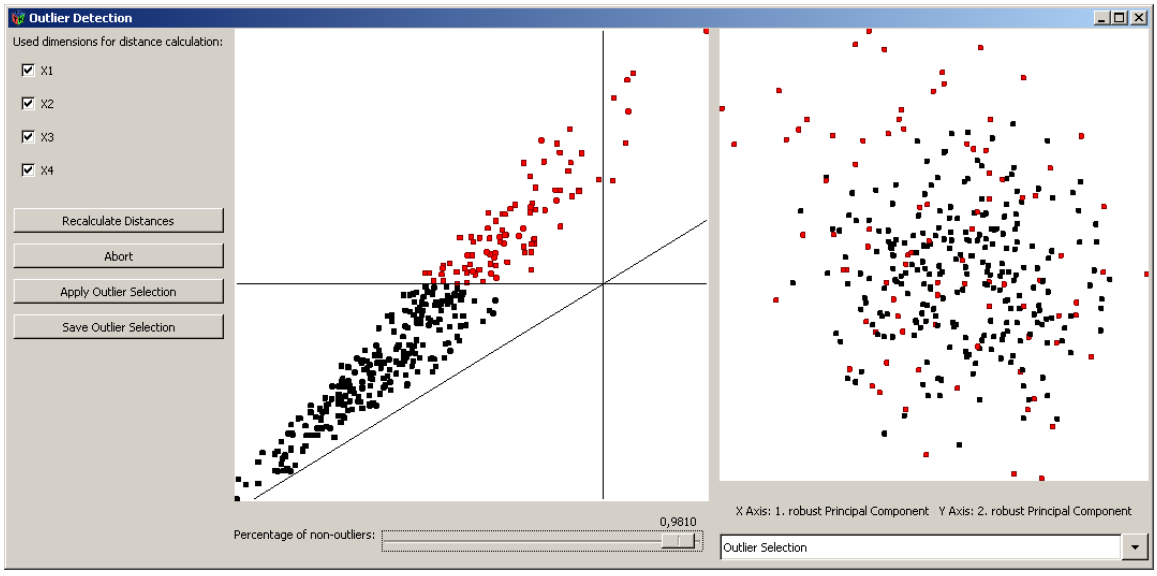


Figure 5.2: A screenshot of the interactive outlier detection tool showing the selected attributes and buttons for the invocation of storing and rerun operations as well as the outlier decision view in the center and the highlighted outliers shown in a scatterplot on the right presenting the first two robust principal components of the dataset.

distances come from this chi-squared distribution. Thus if a quantile of 97.5 % is set, 2.5 % of data would be marked as outliers. Because of the fact that data does not apply to this multivariate normal distribution, deviations from this theoretic values occur. Consequently the user is encouraged to interactively specify the correct chi-squared quantile in the user interface. The visualizations give immediate feedback and help to find the best setting.

5.3.3 Interactive Dimension Reduction

The approach for the interactive dimension reduction that is proposed in this work is strongly related to Visual Hierarchical Dimension Reduction (VHDR) [121] and can thus also be seen as an interactive feature subset selection procedure. Nevertheless differences between those two applications exist and will also be discussed shortly in this section.

A hierarchical clustering algorithm for grouping the attributes of the data is applied after starting the interactive dimension reduction tool. As similarity measure the absolute correlation value between pairs of dimensions is used. Thus the maximum similarity value is 1, while dissimilar variables are decorrelated and show values near 0. To compare clusters the minimum absolute correlation between pairs of dimensions in the cluster are considered. Thus the hierarchical clustering applies the complete link metric. In comparison to that VHDR uses a different similarity measure that counts the number of data items that have similar values in the compared attributes.

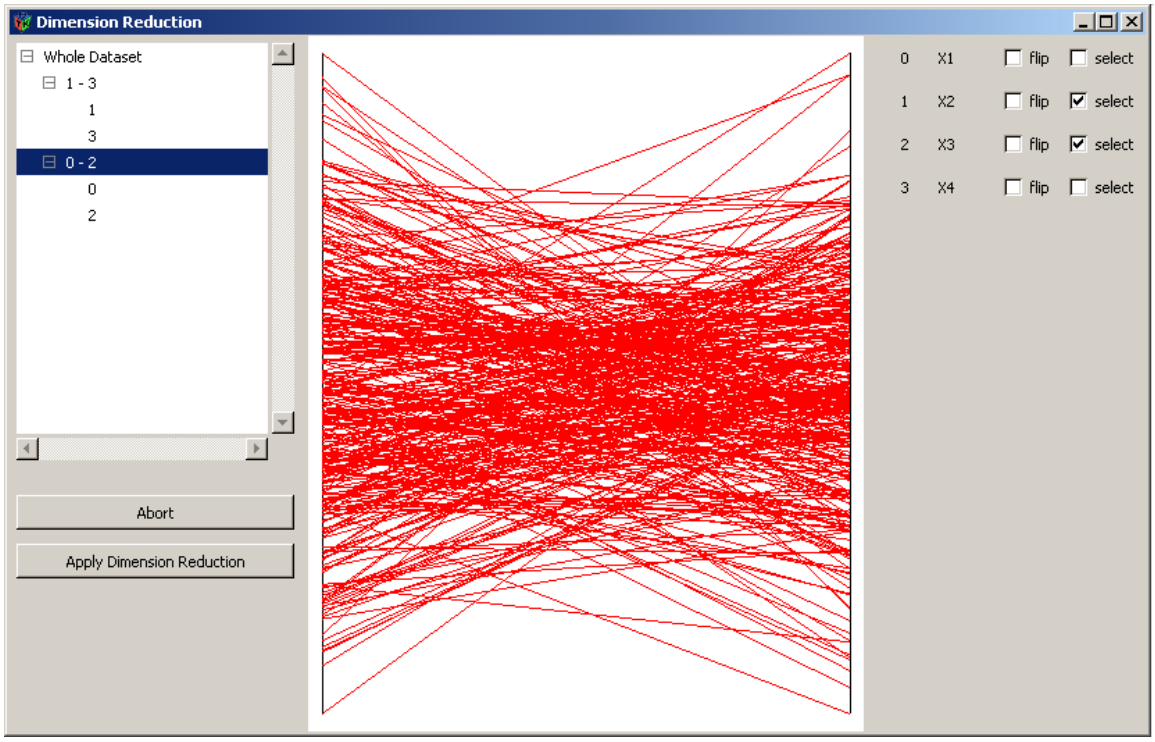


Figure 5.3: A screenshot from the interactive dimension reduction tool showing the illustration of the dendrogram structure in the upper left corner and a parallel coordinates visualization of the selected dimension group in the center of the window.

While VHDR allows an interactive exploration and manipulation of the established dimension hierarchy with the InterRing [122] visualization tool, in this work a simple dendrogram structure is visualized. It allows collapsing and expanding operations for the dimension clusters. Additionally if a cluster is selected, the variables of this group are visualized in a parallel coordinate plot as depicted in figure 5.3. This rather simple exploration approach should help the user to select dimensions for further processing steps. No limitations are given for the user, so that any subset of dimensions, independent from the introduced attribute grouping, can be selected. VHDR restricts the user to choose clusters and proposes representative dimensions for the chosen clusters. Certainly this constraint is softened by the fact that the user is able to manipulate the clustering result and the selection of the representative dimensions.

Additionally the interactive dimension reduction allows the user to pick the principal components of a cluster, which is not supported by VHDR, which uses the dimension subset only for visualization issues. Although the selection of a principal component implies a more complex interpretation of the dimension reduction, it may be helpful for statistical approaches like clustering.

5.3.4 Interactive Clustering

The very popular *k* means clustering algorithm is used to enhance a group finding heuristic with interaction possibilities in an information visualization view. The clustering result is visualized by a scatterplot depicting the data items projected on the first two principal components. Furthermore a scatterplot and a parallel coordinates view can be opened to explore the original data. The data items are coloured according to the partitions introduced by the *k* means procedure. Additionally the cluster centers are accentuated in the scatterplots. This is shown in figure 5.4.

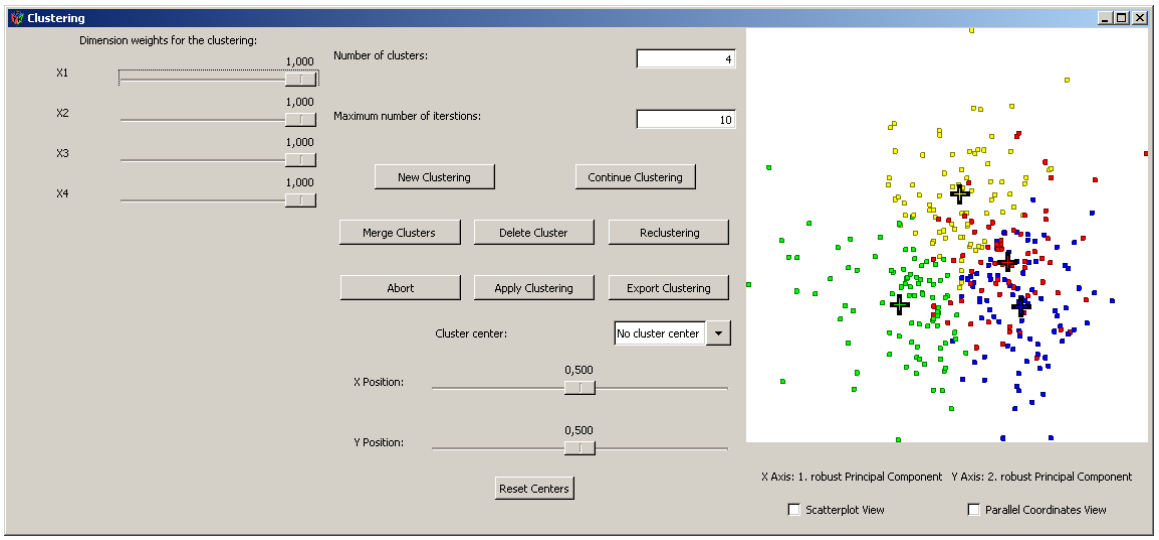


Figure 5.4: A screenshot of the interactive clustering interface showing sections for the dimension weights, for the *k* means properties and the possibilities for interactive clustering as well as for the visualization of the cluster result. The latter is accomplished by a scatterplot presenting the first and second principal component of the clustered data.

To overcome the drawback of *k* means to specify the number of used clusters in advance, the achieved clustering result can be examined and clusters can be subclustered or splitted into two regions, if they cover several groups. Because the clustering in general calculates a local optimum of its energy function, cluster centers can be repositioned and a reclustering can be initiated to test, if a better result can be achieved by a different initial center setting.

These interaction techniques can help the user to explore the importance of certain dimensions for the clustering and the stability of cluster solutions. If a dimension is found that has major influence on the grouping heuristic, the weight of this attribute can be decreased, so that the distance calculations do no longer depend to this extreme extend on this variable. The concept of dimension weights certainly can also be used to set the degree of interest per variable.

5.3.5 Group Fingerprints

As final application a visual analysis of the created groups (outliers and clusters) can be made. Therefore a simple bar diagram has been implemented. For each dimension the relative difference between the mean vector of the group and the center of the whole dataset is shown. This tool provides a numerical as well as a visual identification of a cluster or an outlier group by summarizing its main characteristics. An example of this visualization is shown in figure 5.5, where the center of the group shows significantly higher values in the variables $X1$ and $X2$ in comparison to the mean of the whole dataset. In the attribute $X3$ members of the group seem to have low values compared to the remainder of the data, while the dimension $X4$ can not be used to characterize the group because of the small difference between the two compared centers.

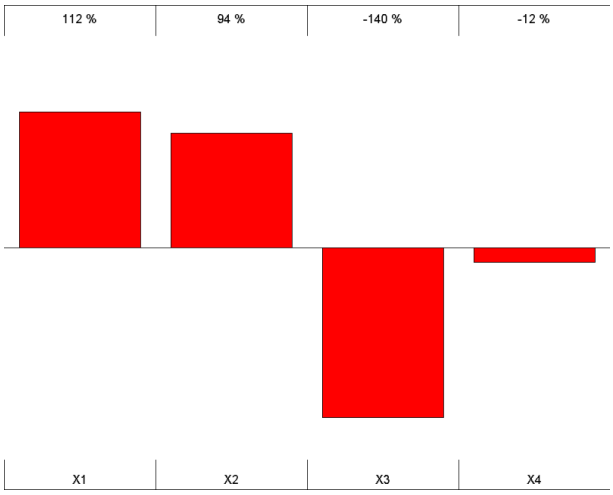


Figure 5.5: An example of the group fingerprint visualization for a cluster in a four dimensional dataset.

Chapter 6

Proof of Concept Cases

This section discusses two proof of concept cases that demonstrate the effectiveness of the interactive combination of statistical routines and information visualization techniques. For this purpose the concepts for semi-automatic sense making discussed in section 5.3 are realized. Based on the computational algorithms of the statistics library an application was implemented that applies basic visualization techniques like 2D scatterplots and parallel coordinate views. To make a fast drawing of the graphics possible, OpenGL [105] was applied . It was not in the scope of this work to explore possible improvements of information visualization views by enhancing the presentation of the data items by the results of statistical routines. Thus no efforts to reduce cluttering or to enhance the perception of outliers were made. Instead fundamental visualization approaches were used as validation tool and most important as interface for the user to modify and adapt the results of the statistical algorithms. To achieve this also a graphical user interface based on the GTK+ [3] library was created.

Furthermore these examples show that it is essential to provide a variable workflow that can be easily adapted to the user's needs. As the realized application covers with tools for transformation, dimension reduction, outlier detection, clustering and visual group analysis five fundamental tasks for the visual data mining process, the sequence in which they are applied is not fixed. One essential reason for this is that transformations should be applied before each multivariate statistical routine like clustering or outlier detection. But also for the realization of a information drill down process an outlier detection can be executed before a group finding process, to avoid distortions by outlying values, and certainly also after the partitions have been created to accomplish further analysis of single clusters. As this list of meaningful orders of data mining tasks could be carried on, the demonstration application allows executing each step at an arbitrary position in the workflow, whereby it can be based on the results of its predecessors.

6.1 Letter Recognition

In the first proof of concept case the letter image recognition data [39] containing 20000 observations and 16 numeric variables is used, which describe the properties of letters given as black-and-white images. To make the demonstration of an interactive clustering application possible, only the letters A, B, C, D, E and F are considered which reduces the number of data items to 4640. For the analysis all numeric attributes were used except the horizontal and vertical position of the bounding box of the letters that do not contain information to discriminate the letter types. The remaining attributes of the dataset, which were considered for this example data mining process, are stated and explained in table 6.1. As the dataset is based on the discretization of the letters by a pixel raster, these measurements show integer values and are not continuous.

Attribute ID	Abreviation	Explanation
0.	*width*	The width of the bounding box of the letter
1.	*high*	The height of the bounding box of the letter
2.	*onpix*	The number of pixels set to on.
3.	*x.bar*	The mean of the x coordinates of the on pixels.
4.	*y.bar*	The mean of the y coordinates of the on pixels.
5.	*x2bar*	The variance of the x coordinates of the on pixels.
6.	*y2bar*	The variance of the y coordinates of the on pixels.
7.	*xybar*	The correlation between x and y coordinates of the on pixels.
8.	*x2ybr*	The mean of $x^2 * y$.
9.	*xy2br*	The mean of $x * y^2$.
10.	*x.ege*	The mean of the vertical edge pixels.
11.	*xegvy*	The correlation between *x.ege* and y
12.	*y.ege*	The mean of the horizontal edge pixels.
13.	*yegvx*	The correlation between *y.ege* and x

Table 6.1: The attributes of the letter image recognition data used for the data mining process.

For the analysis of the dataset in a first step the interactive dimension reduction is applied to exclude attributes, whose information is well represented by others, and to investigate the relationships between the variables. Afterwards an interactive clustering process is initiated, where a k means approach partitions the data. The user can modify the introduced division of the data items by repositioning the cluster centers and by reassigning the objects to the cluster with the nearest center. Finally a visual analysis of the created groups is performed.

6.1.1 Interactive Dimension Reduction

For the dimension reduction process the values of the attributes were mapped to the unit interval. This is in general not necessary for the hierarchical clustering, which is based on the correlation information, but if dimension groups are summarized by principal components, equal value ranges of the variables are crucial. The correlation measures do not depend on different scales of the attributes, because the robust estimates take only the ranks of the data values in account, and the classic Pearson correlation normalizes the covariance calculations by the standard deviations of the variables.

The interactive dimension reduction tool starts with a hierarchical clustering of the dimensions. As similarity measure the Pearson correlation is considered. The resulting dendrogram structure is depicted in figure 6.1, where the IDs of the attributes according to table 6.1 are shown. The user can collapse and expand each node of the dendrogram to navigate through the tree structure. By selecting a dimension group a parallel coordinate view is shown, which illustrates the corresponding attributes in the order they are listed in the node name. For the selection of a single attribute no visualization is defined, because this tool aims to analyse the relationships between variables.

The use of parallel coordinates allows an efficient visual detection of similar variables. Hence it is crucial to place dimensions, which are highly correlated, near to each other. This is accomplished by using the ordering introduced by the clustering procedure. Negative correlations patterns, which are apparent in parallel coordinates by a large number of line crossings, disturb this efficient pattern recognition by causing additional cluttering. Thus it is essential, that the user can flip axis, so that a positive correlation is visually established. Consequently highly correlated dimensions show parallel lines for the majority of the data items and can be easily recognized. Also an easier pattern recognition is accomplished by the dimension ordering, which can be seen in figure 6.2, where the dataset is shown by using the occurrence of the attributes in the dataset as ordering, and by using the dimension ordering created by the clustering. Additionally the attributes *x.bar* and *xybar* have been flipped, which is obvious, if the minima and maxima of the dimension values are examined.

For the interactive detection of attributes that can be omitted for further tasks, a top-down and a bottom-up approach can be considered. The latter starts by scrutinizing each dimension pair in the dendrogram to decide, whether both variables represent the same information. If this is the case the user can exclude for example the attribute showing the smaller value range from further observations and propagates the result of this examination up to the next hierarchy. If both dimensions show different patterns, no exclusion can be applied. Certainly this approach is the most accurate procedure, but it is unsuitable for datasets with more than 20 dimensions. In contrast to that the top-down approach recommends the selection of a level in the dendrogram structure, which also means that a certain number of attribute groups is considered. Now

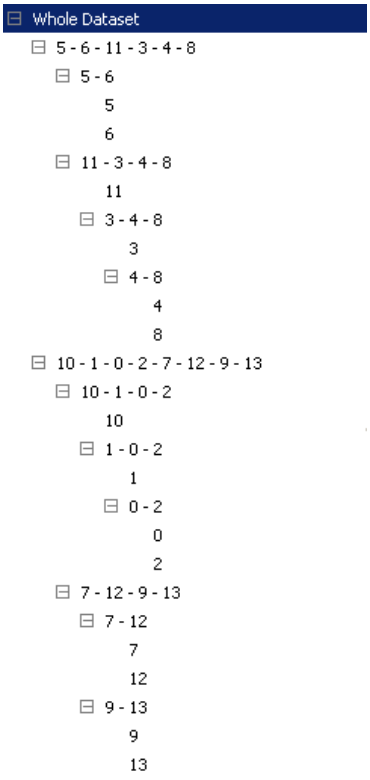


Figure 6.1: The dendrogram structure created by the hierarchical clustering of the attributes of the letter image recognition data based on the Pearson correlation.

each group is investigated according to similarities. As the clustering introduced a dimension ordering, the user can expect that the most similar attributes are placed near to each other, which allows an efficient detection of variable groups representing the same patterns.

For this proof of concept case the top-down approach is chosen, for which four dimension clusters are considered, which are illustrated in figure 6.3.

The first group holds only the dimensions *x*2*bar* and *y*2*bar*. As they do not show a visual pattern, which indicates high correlation, both have to be considered. This conclusion is underpinned by the correlation value of 0.47 and the comparatively low explained variance of 73.7 % achieved by the first principal component calculated from these dimensions.

In the second group the attributes *y.bar* and *x*2*ybr* show the highest correlation value of 0.78. The first principal component computed from this dimension pair can explain 89.6 % of their variance. Consequently one of these dimensions can be excluded. As the attribute *y.bar* shows a higher value range it is chosen to represent both variables. Additionally the numerical information and the visual pattern between *x.bar* and *y.bar* indicate a strong dependency between them. Consequently also the variable *x.bar* can be represented by *y.bar*. In contrast to

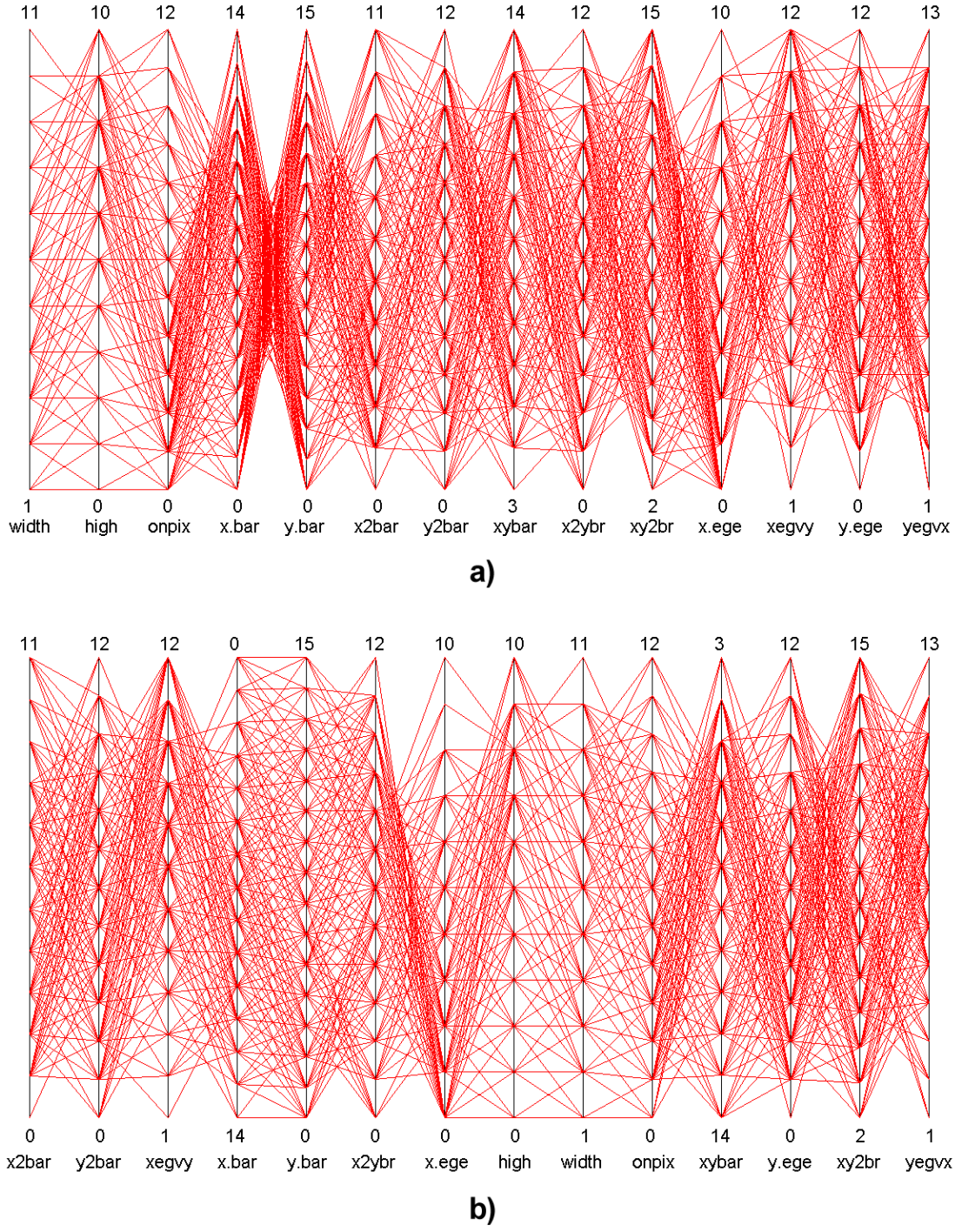


Figure 6.2: A comparison between the illustration of the letter image recognition data according to the attribute order in the dataset (a) and according to the dimension ordering introduced by the clustering (b).

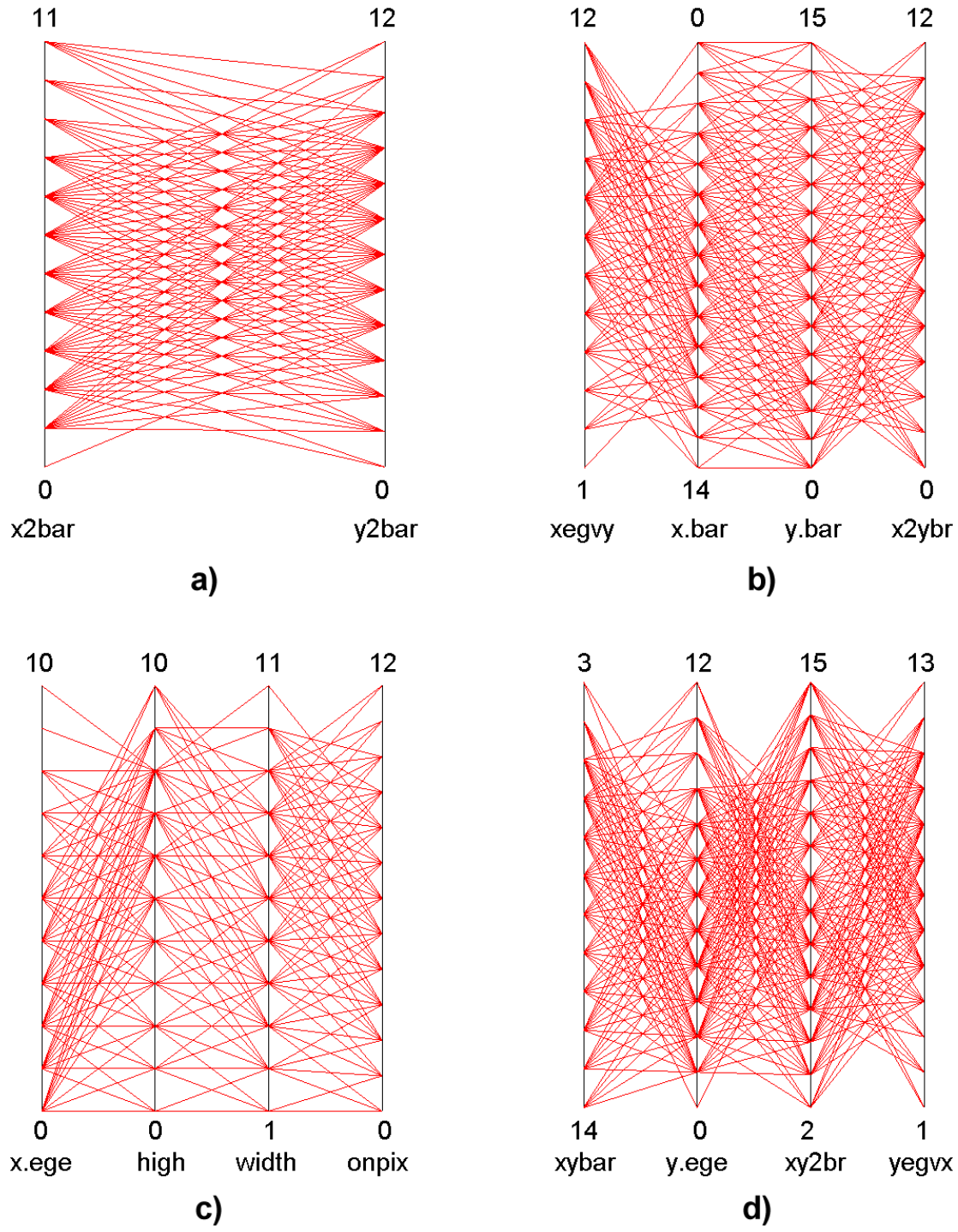


Figure 6.3: The four dimension groups that are examined for this proof of concept case are depicted in parallel coordinates.

that *xegvy* seems to have the highest dissimilarity patterns to the other attributes, so that it has to be kept for further tasks.

In the third group an analogous situation is given. While the attribute *width* can be considered as representative for *high* and *onpix*, the variable *x.ege* captures different patterns. In contrast to that the fourth group shows the highest dissimilarity values, so that no attribute can be excluded.

As a final result of this dimension reduction for the four attributes *high*, *x.bar*, *x2ybr*4 and *x.ege* representatives could be detected. The remaining 10 dimensions were kept to avoid a considerable information reduction.

6.1.2 Interactive Clustering

For the clustering it is a necessity to provide dimensions with the same value range to avoid, that single attributes have stronger influence on the group finding process. As no principal component analysis was applied in the interactive dimension reduction, all used dimensions values are already mapped to the unit interval.

The interactive clustering application starts with an initial k means clustering algorithm, which provides a starting point for the group finding process. The result of this clustering step can be investigated in the parallel coordinates and a 2D scatterplot illustrating the objects projected on the first two principal components is available. Both views show the cluster centers and the data items coloured according to their cluster memberships. The result of the initial clustering can be seen in figure 6.4.

The scatterplot shows that the data reveals three major groups. Two of them are covered by the red and the green cluster. The major group is divided by the remaining clusters. That this grouping, which is shown by the projections on the principal components, is that significant in the data itself, can be doubted, because the first principal component explains 25 % of the variance in the data and the second principal component describes 22 % of the overal scatter.

The parallel coordinates visually recommend that for example the center of the blue cluster should have a higher value in the dimension *x.bar* and lower values for the attributes *y.bar* and *y2bar*. Consequently manipulations of the clustering result can aim to adapt the positions of the cluster centers to the perceived cluster agglomerations in the parallel coordinates and to introduce a better division of the shown data cloud in the scatterplot visualization. (As this application does not pay attention to the plotting order and cluttering of data items in the parallel coordinates, some patterns can be misleading. Therefore a plot of single clusters against all data items could avoid drawing wrong conclusions concerning the optimum coordinates of the cluster centers.)

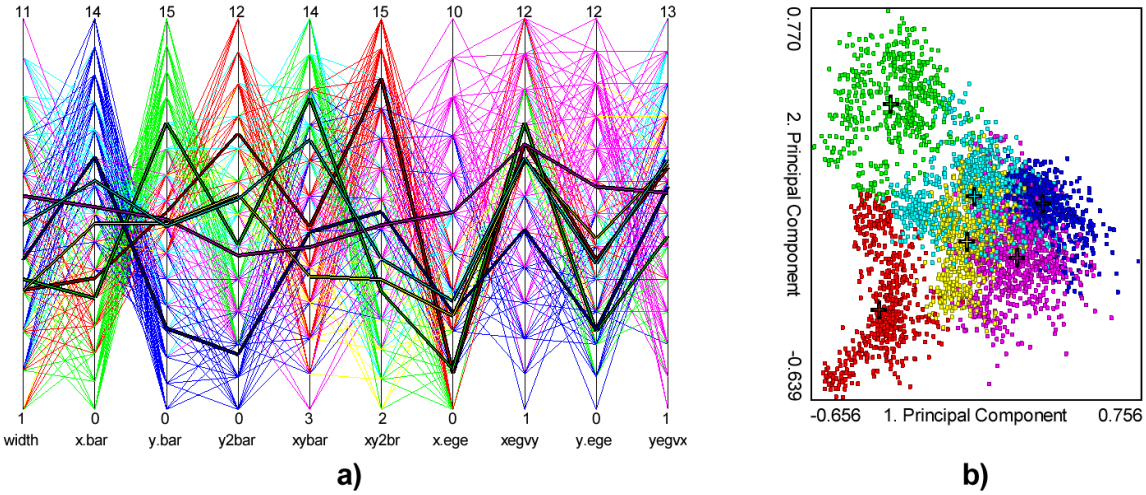


Figure 6.4: Initial k means clustering result visualized in parallel coordinates (a) and a 2D scatterplots showing the first and second principal components (b).

To achieve this, the user can now interactively modify the clustering result by repositioning the centers in the scatterplot view. As this visualization represents the cluster centers as two dimensional projections from the 10 dimensional dataspace, the repositioning actions are mapped back in the coordinate system of the dataset and thus represent multivariate manipulations of the clustering result. Because the scatterplot is linked with the parallel coordinates, the impact of a repositioning on the single attributes can be observed, which allows an investigation of the functional relationships between the principal components and the original data variables. After the center positions have been adapted, the data items can be assigned to the cluster with the nearest center. The results of this operations are illustrated in figure 6.5.

Here it can be observed, that the reassignment process creates a 2D Voronoi diagram like cluster shape in the scatterplot view. In the scatterplot the centers are now positioned at visual agglomerations within the data cloud (red, green, yellow and cyan cluster) or at extreme positions at the border of it (magenta and blue cluster). The latter positions were introduced to create a clearer division of the main group of the data. In comparison to the initial clustering also the matching of the centers with their data items in the parallel coordinates view could be improved. This can be observed for the blue cluster. Certainly because of the manipulation of the centers in the projected PCA space, it is not possible to set individual dimension values for the center, which would have been essential for the magenta cluster in the dimensions *x.ege* or *y.ege*. Thus also the parallel coordinates view should provide interactive manipulation possibilities.

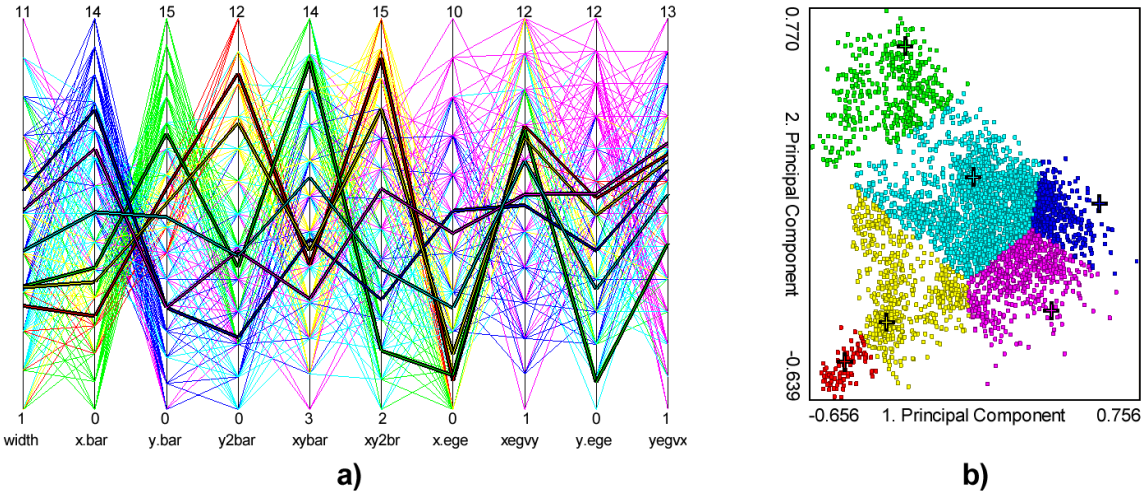


Figure 6.5: The repositioned cluster centers and reassigned data items visualized in parallel coordinates (a) and a 2D scatterplots showing the first and second principal components (b).

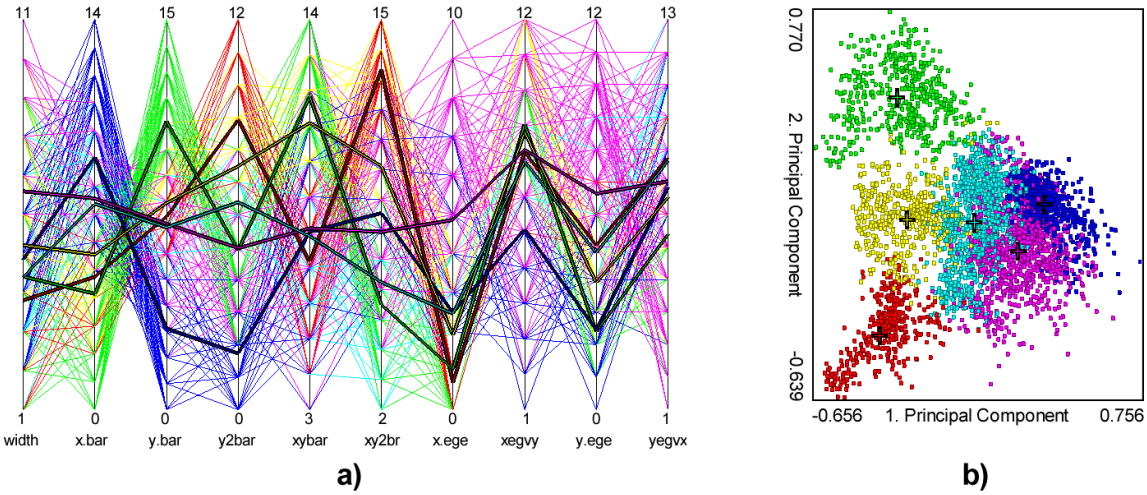


Figure 6.6: The result of the reclustering based on the repositioned cluster centers visualized in parallel coordinates (a) and a 2D scatterplots showing the first and second principal components (b).

Afterwards a reclustering that started with the repositioned cluster centers was initiated. Figure 6.6 shows that this cluster result reveals in the scatterplot, that there are actually four groups in the data. The first three are now covered by the red, green and yellow cluster. The remaining three clusters separate the biggest agglomeration more clearly.

As this dataset also provides the information for each data item, which letter is described, a comparison can be made, if the clustering achieves correct results. An analysis shows that the result of the reclustering is superior to that of the initial clustering. The classification rate for four letters could be significantly improved. For example the green cluster represents 76.6 % of

the objects of letter F after the reclustering, while in the starting solution only 71.4 % were represented. The red and the yellow cluster show a deterioration. Consequently further interactive steps to improve the clustering respectively to adapt the k means results to the structure of the data can be made.

6.1.3 Visual Group Analysis

As a final application for the first proof of concept case a visual group analysis is presented, which creates for each group a visualization that represents a unique "cluster fingerprint". Thus it enables the user to capture the main characteristics of a cluster immediately. Furthermore the differences between the groups and a comparison to the whole dataset can be examined efficiently.

This is accomplished by comparing the cluster center to the mean vector of the whole dataset. The relative differences for each dimension are drawn as bars. The cluster fingerprints for the final clustering result are shown in figure 6.7.

The bar diagrams show with high bars significant differences to the majority of the data. For example the red cluster (figure 6.7 (a)) tremendously deviates from the main behaviour in the dimensions $y2bar$, $xy2br$ and $x.ege$, while the magenta cluster (figure 6.7 (e)) has significant differences in the attributes $x.ege$ and $y.ege$. In contrast to that the cyan cluster (figure 6.7 (f)) achieves the best match of the average behaviour of the dataset.

In the presence of correlated variables also these functional dependencies could be detected, if different groups show similar patterns in those attributes. But as these related dimensions have been excluded in the interactive dimension reduction process, this issue can not be shown in this example.

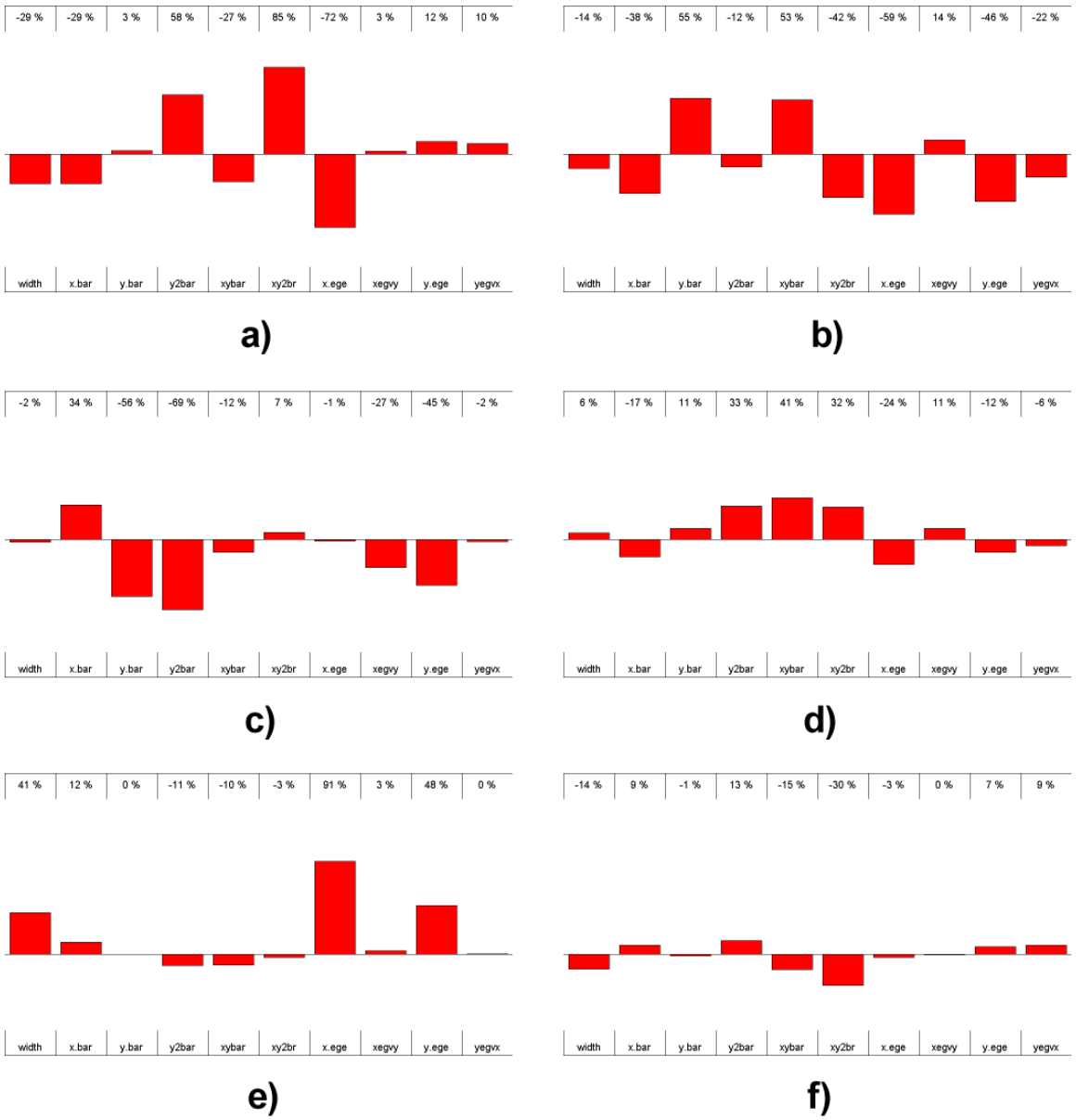


Figure 6.7: Cluster fingerprint visualization: (a) red cluster, (b) green cluster, (c) blue cluster, (d) yellow cluster, (e) magenta cluster, (f) cyan cluster

6.2 Average Wind Speed

The second proof of concept case is based on the average wind speed data [52], which describes the daily arithmetic mean of the wind speeds in knots measured at 12 meteorological stations in the Republic of Ireland. The dataset contains measurements from the years 1961 to 1978 and thus 6574 data items. In table 6.2 the locations of the meteorological stations are stated.

Abreviation	Average Wind Speed measured in knots at ...
RPT	Roche's Point
VAL	Valentia
ROS	Rosslare
KIL	Kilkenney
SHA	Shannon
BIR	Birr
DUB	Dublin
CLA	Claremorris
MUL	Mullingar
CLO	Clones
BEL	Belmullet
MAL	Mull Head

Table 6.2: The attributes of the average wind speed data used for the data mining process.

For the analysis of this dataset an interactive outlier detection is performed to identify extreme data objects, which possibly represent calm or windy days as well as data items that represent atypical wind patterns between the different measuring stations. As this group of outliers can be heterogeneous, a clustering is performed on those items, to investigate, if they show major patterns. Also on the remaining data items a clustering is applied to identify common relationships between the stations. But to accomplish these tasks the attributes must fulfil certain constraints, as the outlier detection requires an elliptic multivariate distribution. Thus this proof of concept case starts with a data transformation.

6.2.1 Data Transformation

As this example application uses a distribution based outlier detection, it is crucial to provide also tools that allow non linear mappings of dimension values so that their distribution can be adapted. For this purpose squareroot and logarithmic transformations are made available. But for the immediate validation, whether the dimension values apply to a given theoretic distribution, also a visualization technique is provided. This is achieved by a scatterplot visualizing the quantiles of the distribution of interest versus the ascending sorted (transformed) dimension values. Additionally a line is drawn through data points representing the first and the third quar-

tile of both value sets. If the drawn data points are arranged along the line, the distributions match, and the transformation can be applied to map the original dimension values to a set of values that show the theoretic distribution that a statistical routine expects.

In the context of this proof of concept example the dimension values should be mapped on a normal distribution, so that the multivariate dataset approximately shows an elliptic distribution pattern. As all dimensions seem to have a similar empirical distribution function, the squareroot transformation showed the best results. Representative for all attributes the possible mappings and the corresponding visualizations for the variable *RPT* is shown in figure 6.8. While the original data values as well as the logarithmic transformed values show significant deviations from the green line, these options could not be used. In contrast to this the squareroot transformation maps most of the data items on the guiding line. Exceptions appear in the tails of the distribution.

6.2.2 Interactive Outlier Detection

After the data preparation the interactive outlier detection tool can be started. This application calculates the robust covariance matrix according to the Fast-MCD algorithm. Consequently a subset of at least 50 % of the data items is searched, which allows the computation of the covariance matrix with the lowest determinant. Afterwards based on this estimate the robust distances for all objects are evaluated. Furthermore the classic covariance matrix is also calculated to provide the Mahalanobis distance for each data item. Both distance measures are visualized in a scatterplot, which is shown in figure 6.9. Additionally the squareroot of the quantile of the chi-squared distribution is drawn, which acts as decision boundary for the outlier classification. The quantile value can be interactively steered by the user. Thus the outlier detection can be interactively modified.

In this example the limit for non-outliers is set to a robust distance of 5.736, which corresponds to the squareroot of the 0.9999 quantile of the chi-squared distribution with 12 degrees of freedom. While the Mahalanobis distance would have classified at about 50 data items as outlying, the robust distance detects 1087 outliers. The reason for this tremendous amount may be that the distribution does not fully satisfy the needs of the outlier detection algorithm.

As further analysis tools the data items are mapped on the first two robust principal components, which are calculated from the previously mentioned robust covariance matrix estimate. These projected data points are shown in a scatterplot, for which two highlighting modes exist. The first shows the detected outliers and is linked with the interactive outlier detection view. The second marks those data items that were used for the calculation of the robust covariance matrix.

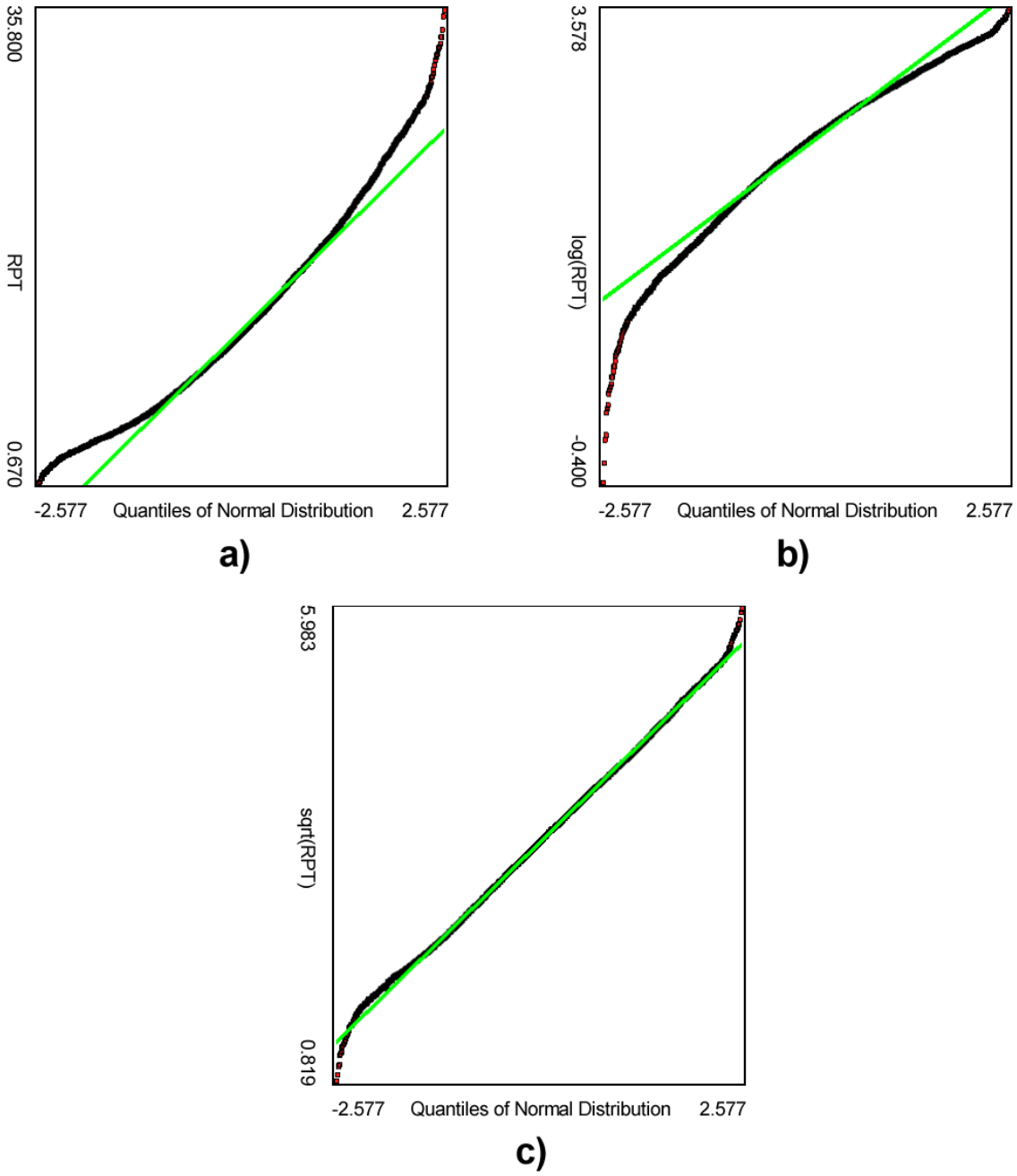


Figure 6.8: Scatterplots showing the quantiles of the normal distribution versus the ascending ordered values of the variable *RPT* non transformed (a), log transformed (b), squareroot transformed (c).

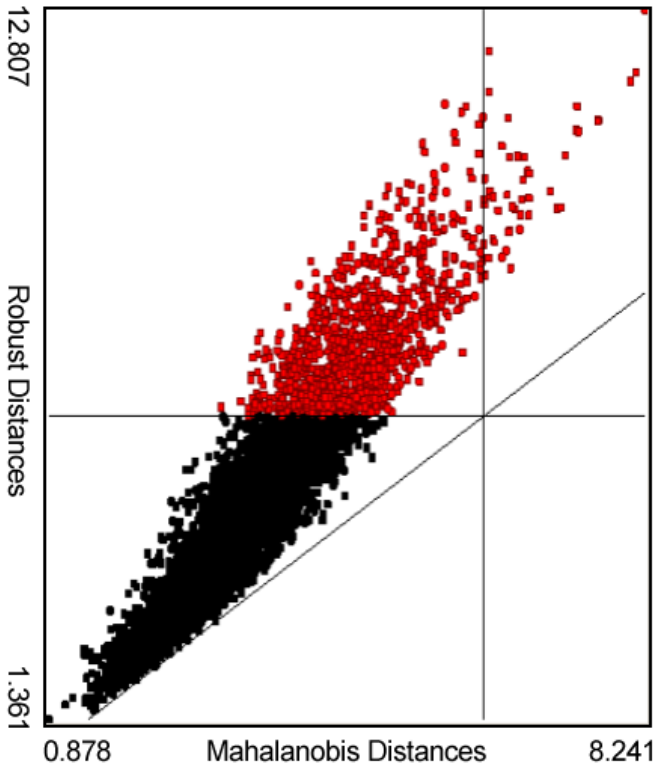


Figure 6.9: Outlier detection view plotting the Mahalanobis distances versus the robust distances. The chi-squared quantile value used as decision boundary for the outlier detection is represented by vertical and horizontal lines.

Both visualization modes for this example are shown in figure 6.11. Here it is obvious that mainly data items on left side of the data cloud are detected as outliers. The reason for this observation is that the objects, on which the computations are based, are located in the right part of the data cloud. An ideal situation would be, if these data points would be located near the center of the dataset, and those objects that are positioned at the boundary of the dataset, would be identified as outlying.

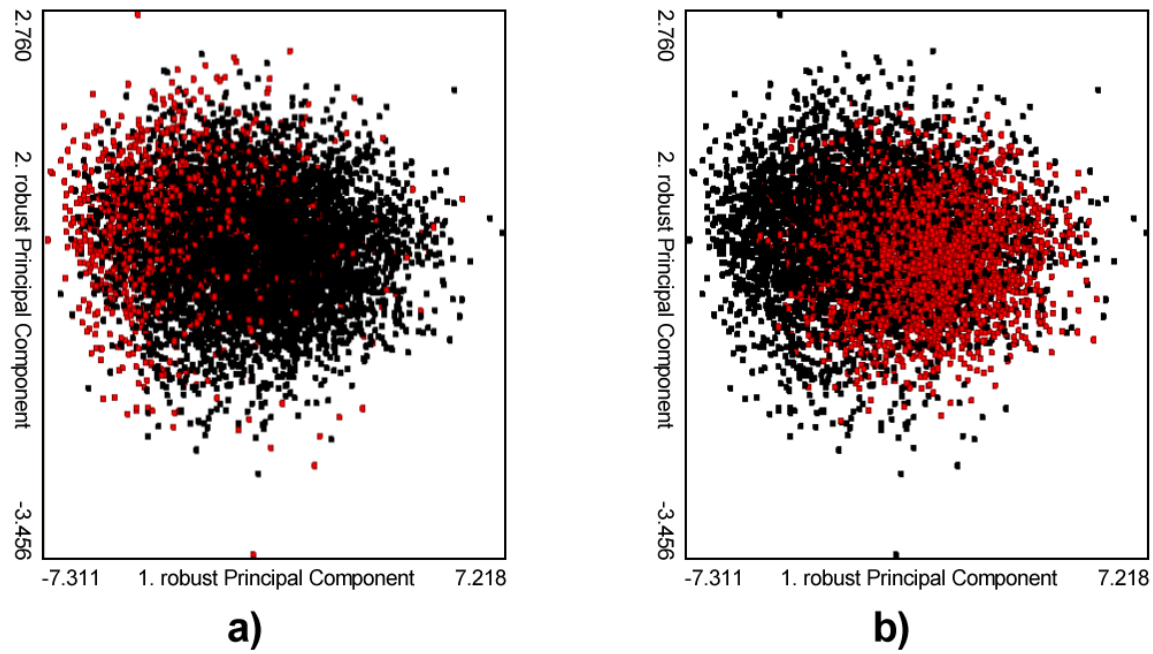


Figure 6.10: Scatterplots showing the data items mapped on the first and the second robust principal component. In (a) the detected outliers, in (b) the used data items for the calculation of the robust covariance matrix are highlighted.

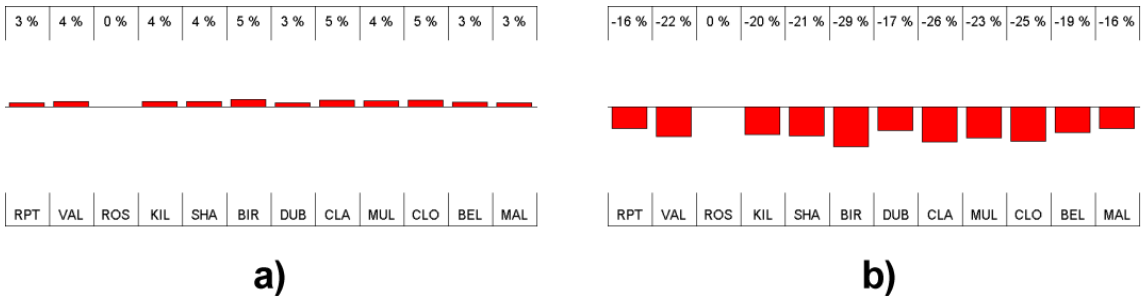


Figure 6.11: Visual group fingerprints for the actual data (a) and the detected outliers (b).

Finally a visual analysis of the actual data and the outliers in comparison to the overall data has been accomplished. This evaluation shows that the detected outliers have significantly lower wind speeds as the whole data set. But as this group can be heterogeneous a clustering is applied in the following section.

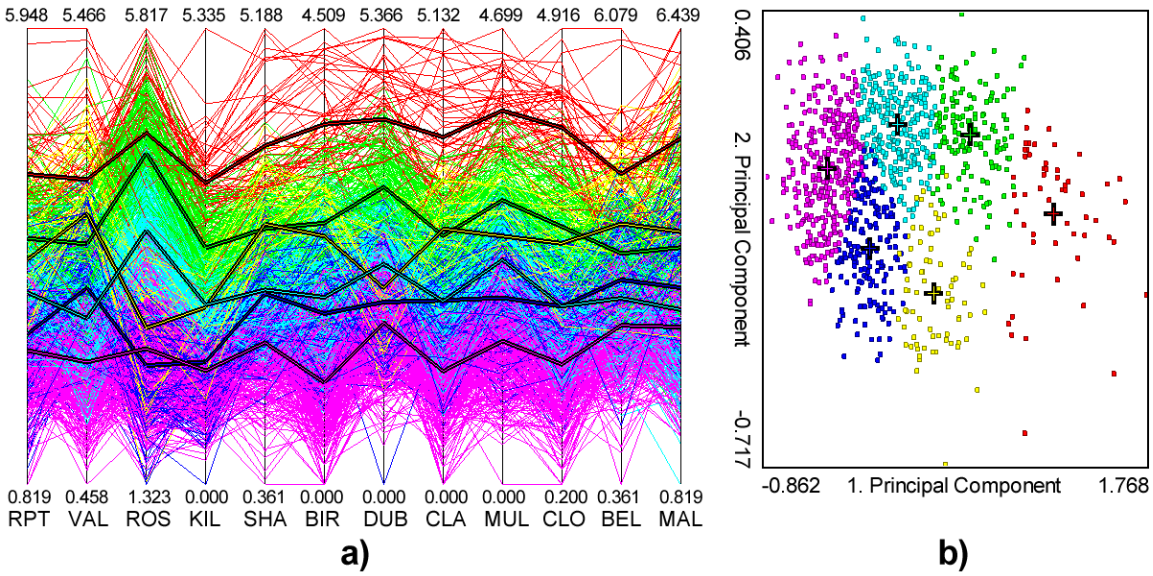


Figure 6.12: The final clustering result partitioning the detected outliers shown in parallel coordinates (a) and in a 2D scatterplot visualizing the first and second principal components (b).

6.2.3 Clustering of Detected Outliers

For the clustering of the identified outliers the visualizations do not reveal any significant groups in the data. Consequently the interactive clustering approach can be used to reposition the cluster centers in order to find better partitions by starting reclustering operations. This was performed several times. The final result is shown in figure 6.12.

The parallel coordinates show that visualizations of the cluster centers cross each other several times. The single exception is the red cluster center, which covers the highest values in all attributes. It is interesting, that the clusters can be ordered according their dominant rank of the dimension values of their centers. This implies the group sequence red, green, yellow, cyan, blue and magenta. Crossings between cluster centers can only be observed for neighbouring groups. For example the yellow cluster center only crosses with the green and the cyan centers. Remarkable dimension pairs could thus be identified as those neighbouring attributes, which show a high number of crossings. Examples are the pairs VAL, ROS and DUB, CLA. It is also obvious that the main trends in this group of outliers do not show a uniform behaviour in all dimensions. Fluctuations in the attribute values are common, especially the variables VAL, DUB and MUL seem to be outstanding, because on these locations most of the groups have their maximum wind speeds.

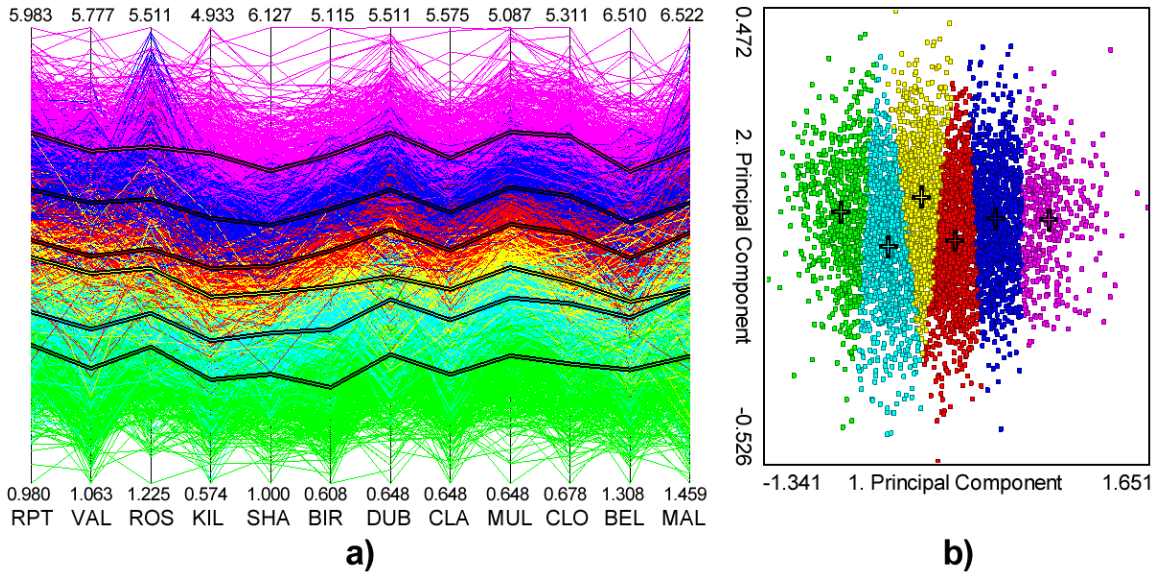


Figure 6.13: The final clustering result partitioning the real data shown in parallel coordinates (a) and in a 2D scatterplot visualizing the first and second principal components (b).

6.2.4 Interactive Clustering of the Actual Data

Analogous to the group of outliers the remaining data does not show agglomerations in the projection of the first two principal components. But in this case the first principal component explains 78 % of the variance in the data, while the second only covers 6.5 % of the information. Hence the clustering introduces nearly horizontal group boundaries in the scatterplot shown in figure 6.13 (b). The parallel coordinates (figure 6.13 (a)) show that the data items have a uniform behaviour in all attributes and that the clusters simply divide the data items according to the magnitude of the average wind speed per attribute. Consequently the number of clusters that is chosen as well as the partitions that are introduced are highhanded, because there is no significant group structure in the data. Nevertheless the cluster centers show the main trends in the data, which seem to be the same for each cluster: Peaks in the center patterns can be observed at the stations RPT, DUB, MUL and CLO, while SHA, CLA and BEL show low center positions. An interesting question that could not be answered with the implemented tool is, whether the visualization of the same value range on each attribute can even out these patterns, so that the cluster centers appear as flat lines.

Concluding it can be said, that the detected outliers can be seen as those days, on which different wind conditions could be observed at the meteorological stations. From the analysis of the outlier fingerprints in figure 6.11 it can be seen, that these days in average are the calm days with low wind speeds, which is valid for all stations except for the measurements taken at *ROS*. There no difference between the average values of the identified outliers and the actual data can be seen.

Chapter 7

Implementation

This section provides a short documentation of the classes of the statistics library. Therefore general issues of the implementation are discussed. Afterwards the functionalities are addressed by explaining computations that are possible as well as giving example calls of functions and runtime estimates.

7.1 General Comments

To allow the fast processing of millions of data items the statistics library was implemented in C++. The main issues that were considered were to provide methods for interactive changes of the parameters for the statistical routines and to allow the work with datasets containing millions of data items. To fulfil the latter demand, fast loops using pointer operations were implemented to run through all values of large `float` arrays as fast as possible. Especially for robust routines a sorting process is very often used. To avoid the overhead of many sorting operations a fast algorithm for finding the k smallest element [30] of an array was implemented. Empirical tests show the advantage of this algorithm in comparison to a sorting routine.

The interfaces were created in a way that each routine returns a `bool` value indicating `true`, if the calculations finished successfully. This is a must to provide a rather simple concept for handling exceptions and thus a safe usage of the library. The first parameters that are passed to the routines are those, into which the result values are written. Thus single variables and `const float` pointers for the access of read only `float` arrays are passed by reference. For methods that provide copies of a set of values an already created array, into which the values are written, is needed for the function call. The next parameters of the methods are data specific parameters like values of the needed dimensions, the covariance matrix or the mean vector on which the further calculations are based. The final set of parameters concern the properties of the statistical routine. An example is the number of clusters or the maximum number of itera-

tions for the *k* means clustering algorithm. A further feature is to provide as much functionality via static methods as possible. So unnecessary object creation and deletion is avoided and the written code to use those methods becomes shorter and easier to read.

Algorithms like the principal component analysis or the calculation of correlation measures are based on routines provided by the numerical recipes in C [95]. Those routines were developed to provide fast and numerical stable computations of often needed procedures such as the calculation of the eigen values of a matrix. The numerical recipes routines in use are mentioned for each functionality.

Routines like the calculation of classic moments and the transformation of data values are also implemented on data structures provided by the Information Visualization Library (IVL) [91]. These data structures allow the management and the fast processing of large data chunks.

Furthermore this section provides an estimate for the running times of the algorithms of the statistics library. Those running times were timed on an ASUS laptop with Intel Pentium Mobile Technology and a clock frequency of 1.73 GHz and a main memory of 1 GB.

To emphasize that a certain functionality is provided by the statistics library, the prefix SL is used for all class names. Classes that operate on the data structures of the Information Visualization Library are indicated by the prefix SLIVL.

7.2 Utils and Matrix Operations

The class `SLUtils` supports four basic operations that are invoked by calling static methods. First a rank calculation for a given `float` array is implemented, returning the positions of the data values in ascending order. Equal values are ignored and thus have different ranks. Second the `getKSmallestValue` routine returns the *k* smallest element of a given `float` array. For this functionality the algorithms `RANDOMIZED-SELECT, RANDOMIZED-PARTITION` and `PARTITION` [30] were integrated in the library to make a faster alternative to the `qsort` routine available. The `qsort` method performs a quick sort algorithm that is provided by the standard C++ libraries. This alternative is used, if not the whole sorted list of values is needed. Because especially robust statistical routines are based on identifying the data value of a specified rank, this method is often called and helpful for the further integration of the robustness in visual data mining applications. Therefore the running time of the routine was tested to ensure a good performance. The table 7.1 shows the run time comparison between the `qsort` algorithm with the following output of the *k* smallest element and the `getKSmallestValue` routine. Each method was tested five times on three datasets with different number of data items. For the comparison the median as well as the mean of the running times in milliseconds were taken.

Number of data items	618.600	1.546.500	3.093.000
Median `kSmallest`	32	47	79
Mean `kSmallest`	128	1581	91
Median `qsort`	2703	7922	15250
Mean `qsort`	3281	12684	17734

Table 7.1: Runtimes in milliseconds of two different approaches to retrieve the k smallest element.

The third functionality of the class `SLUtils` is the count of unique values, which is important to test, whether there are dominant categories in the given array. The fourth operation arises from the third and returns the unique values themselves.

Assuming the presence of a given `float` array `pValues` holding `iEntryNum` data values and the rank `k` for which the data value should be retrieved, the routines of `SLUtils` can be called as follows:

```
int* pRanks = new int[iEntryNum];
bool bIsCorrect = SLUtils::getRanks(pRanks, pValues, iEntryNum);

float fKSmallest = 0.0f;
bIsCorrect = SLUtils::getKSmallestValue(fKSmallest, pValues, 0,
iEntryNum-1, k);

int iUniqueValNum = 0;
bIsCorrect = SLUtils::countUniqueValues(iUniqueValNum, pValues, iEntryNum);

float* pUniqueValues = new float[iUniqueValNum];
bIsCorrect = SLUtils::getUniqueValues(pUniqueValues, iUniqueValNum,
pValues, iEntryNum);
```

Another helpful class is `SLMatrixOperations` which realizes the calculations of the determinant and the inverse of a matrix. To achieve this, the routines `ludcmp` and `lubksb` of the numerical recipes in C are in use. For the given two dimensional `float` array `pMatrix` and `pInvMatrix` holding `iDimNum` × `iDimNum` values the matrix operations can be invoked as follows.

```
float fDeterminant = 0.0f;
bool bIsCorrect = SLMatrixOperations::getDeterminant(fDeterminant, pMatrix,
iDimNum);

bIsCorrect = SLMatrixOperations::invertMatrix(pInvMatrix, pMatrix,
iDimNum);
```

7.3 Distance Measures

The class `SLDistances` provides static methods for the calculation of the Manhattan, Euclidean, the Mahalanobis and the robust distance. Furthermore the squared Mahalanobis, squared robust and the squared Euclidean distance can be computed. A dimension weighting option is available for the Manhattan, the Euclidean and the squared Euclidean distance. If no position is given, to which the distances should be calculated, the mean vector of the given data items is used as reference point.

The routine for the calculation of the robust distance has as additional parameter the degree of robustness ranging from 0 to 1. This property is used for the robust estimate of the covariance matrix, where 1 indicates the maximum possible robustness and 0 the classic covariance matrix estimate. Because the Mahalanobis and the robust distance are based on the inverse covariance matrix, the matrix inversion routine of the class `SLMatrixOperations` is called.

Given a dataset represented by a `SLDataMatrix` object `pData` with `iEntryNum` data items, a possible function call for the calculation of Manhatten distances could be as the following. Other distance calculations are invoked similarly.

```
float* pDistances = new float[iEntryNum];
bool bIsOK = SLDistances::getManhattanDistances(pDistances, pData);
```

7.4 Moments

The class `SLMoments` provides methods for the computation of the classic moments arithmetic mean, variance, standard deviation, skewness, kurtosis and mean of absolute deviation. Therefore the routine `moment` of the numerical recipes in C has been adapted. The robust moments median, median of absolute deviations (MAD), quantiles and inter quartile range are also made available. These computations call sorting routines like `qsort` or the `getKSmallestValue` method of `SLUtils`. Thus they introduce a rearrangement of the passed array holding the data values. Besides the classic and robust measures the library also provides the calculation of the trimmed mean and the trimmed standard deviation with an additional parameter steering the trade-off between robustness and efficiency.

To perform the calculation of a moment a `float` variable has to be created that is passed by reference so that the result can be written into it. As final parameters the array `pValues` holding the data values and `iEntryNum` indicating the number of array entries are specified. To calculate all classic and all robust moments the following function calls can be applied.

```
bool bIsCorrect = SLMoments::getClassicMoments(fMean, fAbsDev, fStandDev,
fVariance, fSkewness, fKurtosis, pValues, iEntryNum);

bIsCorrect = SLMoments::getRobustMoments(fFirstQuartile, fMedian,
fThirdQuartile, fIQR, fMAD, pValues, iEntryNum);
```

Furthermore a one dimensional outlier detection can be invoked. For this method a nearly normal distributed sample is assumed. The user can specify the percentage of outliers that is expected. The outlier detection returns a `bool` array setting a flag for each data item indicating, whether it is detected as an outlying value.

The class `SLIVLMoments` provides methods for the calculation of the classic moments by using the data structure `IVLAttributedScalarContents` of the Information Visualization Library. This implementation also allows specifying subsets of data items for which these estimates are computed.

7.5 Correlation Operations

The class `SLCorrelation` provides methods for the calculation of the classic Pearson correlation as well as for the robust Spearman and Kendall correlation. For this functionality routines from the numerical recipes in C are applied. For the classic correlation coefficient the method `pearsn` is used. To compute the two robust correlation measures the routines `spear` and `kendl1` are called respectively. These implementations additionally provide probability values, which indicate with values near zero a significant correlation. A high probability value confirms the null hypothesis that states that two dimensions are not correlated.

A runtime analysis of the correlation calculations was only possible for the Pearson and the Kendall correlation, because the Spearman correlation considers all possible pairs of data points and thus has quadratic computation effort in the number of given data items. The table 7.2 thus shows the computation times for the Pearson and the Spearman correlation on three datasets with different numbers of data items.

Number of data items	618.600	1.546.500	3.093.000
Pearson	16	78	78
Spearman	375	813	1609

Table 7.2: Runtimes of two different correlation measures in milliseconds.

For the computation of the correlation measures the functions `getPearsonCorrelation`, `getKendallCorrelation` and `getSpearmanCorrelation` are provided. For these methods two

`float` parameters have to specified, into which the correlation coefficient and the probability indicating the significance of the correlation pattern are written. Furthermore two `float` arrays are passed holding the data values of the two dimensions that should be analyzed. Finally the number of data items is indicated by the `int` variable `iEntryNum`. Consequently a function call for the calculation of the classic correlation can be stated as follows.

```
bool bIsCorrect = SLCorrelation::getPearsonCorrelation(fCorrCoeff,
fProbability, pDimValues1, pDimValues2, iEntryNum);
```

Besides the possibility of the calculation of the correlation between two dimensions the computation of a correlation matrix is provided to summarize the coherences between a set of dimensions. This implementation writes the correlation values above the main diagonal and the probability values below the main diagonal of the matrix into the given two dimensional array. In the function call for the correlation matrix calculation a parameter can be passed indicating which correlation measure should be used.

Furthermore a hierarchical clustering of a given correlation matrix can be performed. The clustering creates a `SLDendrogram` object, which allows the access to each dimension group, which arose during the hierarchical grouping process. A dimension group is represented by an `SLDendrogramNode`. These data structures can also be used for a hierarchical clustering on data items. But therefore modifications concerning the performance and the data management have to be accomplished.

The class `SLIVLCorrelation` provides the calculation of the Pearson correlation by using the data structure `IVLAttributedScalarContents` of the Information Visualization Library. Again it is also possible to compute the classic correlation between a pair of dimensions as well as the correlation matrix between a set of dimensions. To realize the correlation computation the routine `pearsn` had to be adapted to perform all calculations on the given IVL data structure. These methods also allow the computation of the correlation measure on a subset of data items, by specifying a start index and the number of objects that should be considered.

7.6 Transformations

The class `SLTransformations` makes popular transformations available via static methods. For each transformation a method overwriting the given values with the transformed ones and a method writing the results in a separate array exist. The latter differs in the function name by the tag `Copy`. The following Transformations are provided:

- Absolute transformation: $f(x) = |x|$
- Squareroot transformation: $f(x) = \sqrt{x}$
- Logarithm naturalis transformation: $f(x) = ln(x)$
- Logarithm to the base of 10 transformation: $f(x) = log10(x)$
- z standardization with given mean $\bar{x}$ and given standard deviation σ: $f(x) = \frac{x-\bar{x}}{\sigma}$
  Instead of mean and standard deviation also robust measures can be passed, which results in a robust z standardization.
- Classic z standardization, where the mean $\bar{x}$ and the standard deviation σ are calculated from the given data values: $f(x) = \frac{x-\bar{x}}{\sigma}$
- Robust z standardization, where the median $\tilde{x}$ and the MAD are calculated from the given data values: $f(x) = \frac{x-\tilde{x}}{MAD}$
- α robust z standardization, where the alpha trimmed mean $m(\alpha)$ and the alpha trimmed standard deviation are calculated from the given data values: $f(x) = \frac{x-m(\alpha)}{s(\alpha)}$
- Inverse z standardization with given mean $\bar{x}$ and given standard deviation σ: $f(x) = x * \sigma + \bar{x}$
- Linear scale to unit interval with given minimum and maximum: $f(x) = \frac{x-min}{max-min}$
- Linear scale to unit interval, where the minimum and maximum are calculated from the given data values: $f(x) = \frac{x-min}{max-min}$
- Linear scale to arbitrary interval $[min, max]$. Therefore the data values must lie in the unit interval: $f(x) = \frac{x-min}{max-min}$
- Scale zero preserving: A linear scale to the $[-1, 1]$ interval, where zero values are mapped to zero. Other values keep their sign but are linearly scaled so that the maximum absolute value is mapped to 1.

Besides the linear and non linear transformations for mapping data items to a certain value range respectively for manipulating their distribution this implementation also covers inverse projections such as the inverse z standardization. These functions allow mapping data items or computed facts such as cluster centers back to the original value range of the dimension.

An example for the invocation of a transformation is given by a squareroot transformation on the `iEntryNum` data values of the `float` array `pValues`. Therefore the transformed values are first copied into a new created array. The second call of the transformation overwrites the original data values.

```
float* pSqrtTrafo = new float[iEntryNum];
bool bIsCorrect = SLTransformations::sqrtTransformationCopy(pSqrtTrafo,
pValues, iEntryNum);
bIsCorrect = SLTransformations::sqrtTransformation(pValues, iEntryNum);
```

Additionally a subset of these transformations is also implemented as iterators based on the data structure `IVLAttributedScalarContents` of the IVL. The implemented transformations and their iterator classes are:

- Absolute transformation: `SLIVLAbsoluteIterator`
- Squareroot transformation: `SLIVLSquareRootIterator`
- Logarithm naturalis transformation: `SLIVLLogIterator`
- Logarithm to the base of 10 transformation: `SLIVLLog10Iterator`
- z standardization: `SLIVLZStandardizationIterator`
- Inverse z standardization: `SLIVLZDestandarizationIterator`
- Linear scale to unit interval: `SLIVLUnitIntervalIterator`
- Linear scale to arbitrary interval: `SLIVLIntervalIterator`
- Linear zero preserving scale: `SLIVLZeroPreservinScaleIterator`

7.7 Covariance Matrices

The class `SLCovarianceCalculator` provides static methods for the computation of the classic covariance matrix. Those methods support the use of all data items or of subsets of the data points. The necessary mean vector that is subtracted from the data items can be passed as parameter. If it is not given, the mean vector of the used data items is calculated. For the robust estimate of the covariance matrix an object of this class has to be instantiated. Thereby the following parameters for the Minimum Covariance Determinant (MCD) algorithm can be set:

- `iStartSubsetNumber`: The number of initial subsets that should be created for the search of the global optimum.
- `iStartImproveSteps`: The number of improvement iterations that should be performed on the initial subsets.
- `iBestSubsetNumber`: The number of subsets that should be improved until convergence is reached.

The standard constructor sets the recommended values of 500 initial subsets, on which 2 improvement iterations are performed, whereupon the 10 best solutions were chosen for the improvement iterations until convergence is reached.

The function call for the calculation of the robust covariance matrix takes a parameter steering the degree of robustness of the estimate. A value of 0 indicates that the classic covariance matrix is returned, while a value of 1 makes the most robust estimate with a break down point of near 50 % possible, meaning that nearly half of the data can deviate from the main behaviour of the data points. After running the MCD algorithm the robust covariance matrix, the used robust mean vector and the chosen subset of data items that were considered for the covariance estimate can be retrieved.

A comparison of the runtimes of covariance matrix calculations is given in table 7.3. There both, the classic and robust, algorithms are applied on datasets containing different numbers of dimensions and data items. For the robust covariance calculation the previously mentioned standard settings and the maximum robustness factor were used. Thus to reduce the running time, lower numbers of samples could be set.

Number of data items	618.600	1.546.500	3.093.000
Classic covariance matrix (5 dimensions)	93	204	422
Classic covariance matrix (10 dimensions)	281	687	1406
Classic covariance matrix (15 dimensions)	594	1453	2922
Robust covariance matrix (5 dimensions)	25344	148562	209250
Robust covariance matrix (10 dimensions)	110329	188391	339921
Robust covariance matrix (15 dimensions)	154578	323219	439203

Table 7.3: Runtimes in milliseconds of the classic and robust covariance matrix calculation for different dataset sizes.

To invoke the calculation of the classic covariance the data has to be provided as a `SLDataMatrix` object. As parameter also a two dimensional `float` array has to be created. This array represents the matrix, into which the covariances and variances are written. Thus its dimensionality must match the number of attributes in the data. A possible function call with the two dimensional `float` array `pCovMatrix` and the `SLDataMatrix` object `pData` looks like this:

```
bool bIsCorrect =
SLCovarianceCalculator::calculateClassicCovarianceMatrix(pCovMatrix,
pData);
```

For the robust covariance calculation a `SLCovarianceCalculator` object has to be instantiated. Afterwards the object representing the data is passed to the invocation method of MCD algorithm. Finally the computed information can be retrieved by get methods.

```
SLCovarianceCalculator* pCovCalc = new SLCovarianceCalculator();
bool bIsCorrect = pCovCalc->calculateRobustCovarianceMatrix(pData);

const float** pMCDCovMatrix = NULL;
bIsCorrect = pCovCalc->getMCDCovarianceMatrix(pMCDCovMatrix);
```

7.8 Principal Component Analysis

The class `SLPrincipalComponentAnalysis` allows the calculation of the principal component analysis (PCA). To do that, an object of this class has to be instantiated, for what two options exist. The first possibility allows passing a previously computed covariance matrix and a mean vector to the object. The way those parameters were computed (robust or classic) decide how the PCA is influenced by outlying values. The alternative to that is to specify the data points on which the PCA should be based and a `float` value between 0 and 1 indicating the degree of robustness that should be used for the covariance estimation, which is now called by the `SLPrincipalComponentAnalysis` object. The specification of the robustness can be omitted, if the classic covariance matrix should be considered for the PCA.

For the evaluation of the eigen values and eigen vectors of the covariance matrix the routines `tqli` and `tred2` of the numerical recipes in C were used. This task is performed during the instantiation. Afterwards the principal components and the explained variances of arbitrary subsets of principal components can be retrieved. Furthermore it is possible to map data items on specified principal components.

For a given `SLDataMatrix` object `pData` representing the data the following code achieves a mapping of the data items on the first principal component.

```
SLPrincipalComponentAnalysis* pPCA = new
SLPrincipalComponentAnalysis(pData);

float* pMapping = new float[iEntryNum];
bool bIsCorrect = pPCA->mapOnPrincipalComponent(pMapping, pData, 0);
```

The runtime for the calculation of the principal components of a given covariance matrix with dimensionality 25 amounts 157 milliseconds. For a matrix holding 50×50 entries 547 milliseconds of computation time were needed.

7.9 Clustering

The *k* means clustering functionality is provided by the class `SLKMeansClustering`. To perform the clustering an object of this class has to be instantiated. The constructor sets the standard values 10 respectively 0 for the maximum iteration number and for the minimum update distance of the cluster centers. Set methods provide the possibility to change these standard settings. Additionally dimension weights can be specified. If no weights are in use, all attributes are treated with equal importance. Two different calls to invoke the *k* means clustering exist. The first specifies the initial cluster centers by an `SLClusterCenters` object. This object represents the *k* cluster centers in each dimension, which is used for the clustering. For the second option only the number of clusters has to be passed to the routine. Additionally this method allows to set a flag indicating if the cluster centers should be randomly chosen or found by the best 5 *k* means algorithm on a small subset of 300 data items. Both clustering invocations need the data items that should be clustered represented as a `SLDataMatrix` object. Furthermore a flag could be set, indicating, whether the means or the medians per dimension of the data items per cluster should be used to set the new cluster centers. The same is valid for the distance measure, where the user can choose between the Euclidean and the Manhattan distance. If those flags are not specified by the function call, the standard settings are used, meaning that the Euclidean distance and the means of the data items are calculated during the clustering. After the cluster process get functions allow the retrieval of the calculated value of the objective function of the *k* means clustering result as well as the cluster ids and the distances to the nearest cluster center per data item. Additionally the cluster centers and the number of items per cluster can be queried.

The following code snippet performs a *k* means clustering on a `SLDataMatrix` object `pData` and generates `iClusterNum` groups. Finally the cluster ids are requested and the number of ids is written into an `int` variable.

```
SLKMeansClustering* pKMeans = new SLKMeansClustering();
bool bIsCorrect = pKMeans->performKMeans(pData, iClusterNum);

const int* pClusterIDs = NULL;
int iIDNum;
bIsCorrect = pKMeans->getClusterIDs(pClusterIDs, iIDNum);
```

In table 7.4 the runtimes are stated for the *k* means clustering performed on datasets containing different numbers of data items and dimensions. Therefore the cluster centers were calculated by using the arithmetic mean of the objects assigned to a cluster. In table 7.5 the corresponding computation times are shown for the *k* means based on the median calculations for the cluster centers.

Number of data items	618.600		1.546.500		3.093.000	
Number of dimensions	5	10	5	10	5	10
5 clusters	2250	4125	5734	10969	11250	20500
10 clusters	3734	6797	9579	17719	19109	34516

Table 7.4: Runtimes in milliseconds for *k* means on different dataset sizes whereby 5 as well as 10 clusters were generated. The arithmetic mean was used for the cluster center calculation.

Number of data items	618.600		1.546.500		3.093.000	
Number of dimensions	5	10	5	10	5	10
5 clusters	4375	13125	8641	15167	17906	29734
10 clusters	7485	10890	14344	23047	25062	43547

Table 7.5: Runtimes in milliseconds for *k* means on different dataset sizes whereby 5 as well as 10 clusters were generated. The median was used for the cluster center calculation.

The similar procedure can be applied to the class `SLFuzzyKMeans`, which implements the fuzzy *k* means clustering. Before the clustering as additional parameters to those introduced by the *k* means algorithm the fuzzification exponent can be set. The invocation of the fuzzy clustering can be accomplished as explained above. Merely the get methods provide different information per data item. Instead of the cluster ids and the distances to the nearest cluster, the memberships per data item to a given cluster are provided. The runtimes of this fuzzy clustering implementation are summarized in table 7.6.

Number of data items	618.600		1.546.500		3.093.000	
Number of dimensions	5	10	5	10	5	10
5 clusters	10922	13516	27375	33500	53937	60937
10 clusters	28046	33219	69797	83016	139360	166234

Table 7.6: Runtimes in milliseconds for the fuzzy k means clustering on different dataset sizes whereby 5 as well as 10 clusters were generated.

7.10 Regression

The class `SLRegression` provides a static method for the calculation of the least squares regression. For this function the independent dimensions of the data items have to be specified by a `SLDataMatrix` object. The dependent values of the data points are passed as a `float` array. For the calculation of the regression parameters the matrix inversion implemented by the `SLMatrixOperations` is used. The parameters are written into a `float` array, that has an additional entry for the constant regression estimate besides the estimates for each independent variable. Thus a possible invocation of the linear regression with the `SLDataMatrix` object `pXData` and the `float` array `pYValues` can be accomplished as follows.

```
float* pRegCoeff = new float[iDimNum+1];
bool bIsCorrect =
SLRegression::getLeastSquaresRegressionParameters(pRegCoeff, pXData,
pYValues);
```

In table 7.7 the runtimes for the least squares linear regression implementation are given for different numbers of independent attributes and data items.

Number of data items	618.600	1.546.500	3.093.000
Linear regression (1 independent variables)	63	156	281
Linear regression (2 independent variables)	125	343	594
Linear regression (5 independent variables)	453	1016	3797
Linear regression (10 independent variables)	17844	6359	5985

Table 7.7: Runtimes in milliseconds of the linear regression applied on different dataset sizes.

7.11 Theoretic Distributions

The five provided theoretic distributions are implemented in the classes

- `SLUniformDistribution`,
- `SLNormalDistribution`,
- `SLLogNormalDistribution`,
- `SLChiSquaredDistribution` and
- `SLExponentialDistribution`.

All classes provide static methods to retrieve the density values (pdf), the distribution values (cdf), the quantiles and accordingly distributed random values. Each method allows the setting of the parameters of the distribution. If standard parameters like the zero mean and the standard deviation of 1 for the normal distribution are in use, the settings can be omitted. Thus the function calls are shorter and look like these:

```
SLNormalDistribution::getDensityValues(pPdf, pXValues, iEntryNum);
SLNormalDistribution::getDistributionValues(pCdf, pXValues, iEntryNum);
SLNormalDistribution::getQuantileValues(pQuantileValues, pQuantiles,
iEntryNum);
SLNormalDistribution::getRandomValues(pRandomValues, iEntryNum);
```

Thereby the first parameters represent arrays into which the results of the routines are written. The `float` arrays `pXValues` and `pQuantiles` hold the data values respectively the quantiles, which are needed for the calculations. The `int` variable `iEntryNum` specifies how many entries the arrays contain.

For the creation of the random values that are distributed according to the given distribution the method `ran2` of the numerical recipes in C is used. This routine is not the fastest random number generator provided by this library but it ensures a sequence of non repeating random numbers with a length of more than 2×10^{18}.

Besides the random number generation further routines of the numerical recipes in C had to be integrated into the library to provide these functionalities. Because there is no analytic solution for the integral of the probability density function (pdf) of the normal and the log normal distribution the values of the cumulative distribution function (cdf) have to be integrated numerically. Hence the routine `qromb` implementing the Romberg's numeric integration scheme is in use. This routine calls the helping methods `trapzd` and `polint`. For the computation

of the pdf of the chi-squared distribution the function `gammln` returning the logarithm of the gamma function is integrated. The cdf of the chi-squared distribution requires the evaluation of the incomplete gamma function, which was realized by adding the routine `gammq` to the implementation.

7.12 Statistical Tests

The class `SLKolmogorovSmirnovTest` provides functionality to test, if a set of `float` values comes from a normal, log normal, exponential or uniform distribution. Additionally a test if two samples come from the same distribution is possible. To invoke this functionality, an object of this class has to be instantiated and the needed test can be applied by calling one of the five test functions. Afterwards the significance level (p-value) and the Kolmogorov-Smirnov statistic can be retrieved by get methods. An important parameter concerning the tests is the significance, which is the decision limit for the significance level deciding, whether the null hypothesis is rejected or not. This parameter can be set in the constructor or by a set method. The standard value for the significance is 0.05. For the calculation of the Kolmogorov-Smirnov test the routine `ksone` of the numerical recipes in C was adapted for the test between a sample and a theoretic distribution, while testing for the same distribution of two samples is based on the routine `kstwo`. To evaluate the p-value for the Kolmogorov-Smirnov statistic the function `probks` is in use. The results of the tests show that `probks` provides a more conservative estimate of the p-value than the popular software package R, meaning that the nullhypothesis is rejected more likely than by using R.

To apply a hypothesis test for uniform distribution on `iEntryNum` values represented in the `float` array `pValues` the following code snippet can be used. Thereby `true` is written into the `bool` variable `bNullHypothesis`, if the values come from a uniform distribution.

```
SLKolmogorovSmirnovTest* pKSTest = new SLKolmogorovSmirnovTest();
bool bNullHypothesis;
bool isCorrect = pKSTest->isFromUniformDistribution(bNullHypothesis,
pValues, iEntryNum);
```

A comparison between the runtimes of tests for uniform distribution on a single array of values and of tests for the same distribution based on two samples is presented in table 7.8. For these tests different numbers of data items were considered.

Number of data items	618.600	1.546.500	3.093.000
Test for uniform distribution	1516	5485	10687
Test for the same distribution	7079	20672	41266

Table 7.8: Runtimes in milliseconds of the hypothesis tests calculation for different numbers of data items.

Chapter 8

Summary

The exploration of high dimensional datasets is a tremendously growing working field. With the capabilities of today's computers to handle data containing millions of data points and thousands of dimensions it is essential to apply efficient methods to extract the information the user is searching for. Statistical routines and techniques of information visualization are useful to achieve this goal. But as one method on its own has several shortcomings combinations between the different capabilities of these sciences could be developed to improve the exploration of multivariate data, the so called data mining process.

8.1 Introduction

Information visualization techniques create graphics and animations that stress certain structures and aspects of high dimensional data. The user, who examines the data, applies his or her pattern recognition skills as well as the experience and knowledge about the data to draw the correct conclusions. This is an efficient approach to detect data items of special interest, examine the main trends in the data or investigate functional dependencies between variables.

In contrast to that statistical routines use the possibilities of computers, which execute millions of operations within milliseconds. This allows the fast calculation of facts and numerical summaries. Also models that can predict variable values or introduce a simplification of the data can be fitted. Thus coherences within attributes as well as significant patterns of the data items can be revealed and analysed.

Because of the usage of different systems that gather the information of interest, a combination of those sciences would introduce a verification of the results of the applied methods. Consequently an error detection approach for the data mining process could be established that decreases the probability that wrong conclusions are implied. But the intelligent application of the strengths of the disparate techniques also makes a more efficient data exploration possible.

To achieve this collaboration, in this work the implementation of a library containing statistical routines adapted for the use in information visualization applications is presented. Because of the vast number of routines developed in the field of statistics for data analysis and exploration, the basic functionality, that every visual data mining tool should provide, had to be determined. Furthermore aspects like robustness, which decreases the occurrence of distorted results caused by outliers, and fuzzyness, which allows soft decision boundaries to describe uncertainties, are considered. A further demand on the library is that its routines must be able to process large datasets efficiently.

Furthermore a sample application was developed to demonstrate possible combinations of visualization and statistics. In the focus of this tool are tasks like outlier detection, dimension reduction and clustering, where computational approaches are combined with visual verifications and user interactions that can manipulate the results of the statistical routine. Special attention was paid on an interactive workflow, where the user can determine the order of the steps of the data mining procedure.

8.2 Related Work

The information visualization techniques to illustrate multivariate datasets are manifold. They can be roughly classified into the four categories geometric projection techniques, icon-based and pixel-based approaches and finally hierarchical visualizations [55]. This work applies graphic representations of the first type, which maps the variables of the data on the screen space. The most popular approaches of this category are scatterplots and scatterplot matrices [27] as well as parallel coordinates [60]. For scatterplots two attributes are mapped on the x and on the y axis of the visualization space and data items are represented by points in the coordinate system that is spanned. To allow an illustration of all dimensions of a dataset a scatterplot matrix was introduced, which shows all possible tuples of variables by scatterplot visualizations. The parallel coordinates achieve the representation of all attributes by mapping them on axes, which are drawn as equidistant vertical lines. The data items are illustrated by poly lines which connect the projected dimension values.

Also the field of statistics provides a multitude of analysis procedures for data exploration. In the scope of this work only tasks are addressed that are of special importance for a visual data mining application. Thus the multivariate outlier detection, dimension reduction and clustering techniques are considered.

As outliers strongly influence statistical routines and cause wrong results, an efficient detection of these objects is crucial. For this purpose a variety of heuristics has been developed. The most popular approaches are distance based, density based and distribution based methods. Routines of the first type consider the distance of each data item to its k nearest neighbour.

If this distance exceeds a user specified limit, the object is identified as outlier [74]. Density based techniques refer to a volume parameter and a minimum number of data points that has to be located within this volume to define dense regions. Data items that could not be assigned to such an area are marked as outlying [21]. The multivariate outlier detection application that is applied in this work uses a distribution based approach, which assumes that the data applies to a multivariate elliptic distribution. This demand is necessary, because for each data item the robust distance is calculated, which is based on the robust estimate of the covariance matrix [101], that describes the shape of the data cloud. If the data objects correspond to the distribution constraint their distance measures show a chi-squared distribution. Consequently a chi-squared distribution quantile can be considered to determine a decision boundary that differentiates between outliers and actual data points.

Clustering approaches group similar data items to introduce partitions of the data. The main two methodologies for this task are hierarchical and partitional techniques. A hierarchical clustering based on a merging operations initiates each data item as cluster. Afterwards the two most similar clusters are merged to a new cluster. This procedure is iteratively performed until only one cluster representing the whole dataset remains. This nested group structure can be represented by the tree-like dendrogram. In contrast to that partitional approaches assign data items to clusters according to an update rule that optimizes a global energy function. The most popular algorithm of this type is the k means clustering [50], where k indicates the user defined number of partitions that are created. While these routines assign each data item to exactly one cluster, fuzzy clustering approaches exist, which calculate for each data item membership values that indicate to which degree it is associated with each cluster. For this purpose the fuzzy k means algorithm [17] was considered.

To reduce the dimensionality of a dataset three main techniques were introduced. The self-organizing maps (SOM) [76] are based on an unsupervised machine learning approach, that tries to iteratively fit reference vectors in data space to the structure of the data items. These vectors are connected in a two dimensional lattice which represents the low dimensional projection of the data. Multi dimensional scaling (MDS) techniques [77] try to achieve a projection of the multivariate data that maintains the distance relationships between pairs of data items. The simplest but nevertheless popular dimension reduction technique is the principal component analysis (PCA) [61], which evaluates the directions of the major variances in the data cloud. These directions are called the principal components, on which a mapping of the data items can be performed. As the first principal components describe the majority of the variance in the data, the main information of the data space is captured by the spanned subspace.

Feature subset selection approaches have the same aim as dimension reduction techniques. But a low dimensional representation of the data is achieved by choosing only the most informative data attributes. As this concept was developed for supervised machine learning

routines, it is difficult to apply it for the data exploration, because no measure for the quality of an attribute can be intuitively introduced.

The integration of computational routines in information visualization applications gained importance in the last 10 years. For this mainly clustering and the creation of low dimensional data representations were applied. The reasons why data partitioning has been favoured are that group finding algorithms provide a fast categorization of the data and significantly improve the detection and interpretation of the main trends. The focus on the reduction of variables simply rises from the fact that humans are used to think in three dimensional spaces, while multivariate datasets represent their main information in a higher number of attributes. To overcome this discrepancy, projection methods as well as feature subset selection approaches were applied.

But while simple visualizations of statistical results only serve to explore and present them, an interactive combination of statistics and visual techniques is rarely realized. An example for a successful interactive collaboration of both fields is the Visual Hierarchical Dimension Reduction (VHDR) [121] system, which applies a hierarchical clustering on the attributes of the data. The introduced dimension groups can be investigated and modified by using InterRing [122] a radial visualization tool for hierarchical data. Finally representative dimensions per selected cluster can be chosen. This approach integrates the user's knowledge and experience into the feature subset selection task for which a starting point is created by a statistical routine.

An interactive visual feature subset selection and clustering tool is presented by Guo [46]. By calculating a measure for the "goodness of clustering" for each pair of variables a colour coded matrix visualization is established, where bright fields represent attribute combinations that show significant cluster structures. As an ordering heuristic is applied the user can identify light regions in the visualization which represent groups of variables that contain groups of data items. These dimensions can be selected and used for a hierarchical clustering approach that detects groups of arbitrary shapes, because of the integration of graph and density based clustering concepts. The additional parameters that were introduced by these enhancements as well as the number of detected groups in the data can be steered by interactive visualizations.

A further example describing the power of the combination of visualizations and computational routines is the HD-Eye approach [55], which adapts the OptiGrid clustering [54], so that the user is involved in the group finding process. A density estimation of the clustering procedure is used to decide, whether the data space can be subdivided by separators such as hyper planes that are positioned in sparse regions. To achieve this, a set of projections is suggested from the system, for which an icon-based visualization indicates, if a mapping is helpful to decide, where a separator can be introduced. The user can select the projection that shows the most significant gaps between groups of data. A histogram-like view, showing the agglomerations of data items by high bars is used to define a separator. This approach is iteratively

applied, until no subspaces can be divided anymore. Consequently this approach is an attempt to incorporate the capabilities of the human visual system into a clustering routine to achieve better results.

8.3 Integration of Statistical Functionality in Visualization

As the sciences visualization and statistics rely on different systems that analyse the data, their weaknesses and strengths are mostly dissimilar. Interactive visual applications provide graphics that can be modified by the user to achieve an efficient information drill down process, where firstly an overview is given. Afterwards zooming and filtering techniques allow the concentration on patterns or data items of special interest. Finally details-on-demand operations show numerical summaries or the data values of the selected subset themselves. Consequently mainly the user's extraordinary pattern recognition skills and knowledge about the data guides the exploration process [108].

Contrary to that statistical routines are algorithms and calculations of formulas that use computers to cope with the enormous computational effort for large datasets. This implies that the applied procedures have to be well chosen for the data that should be analysed. Consequently if a dataset contains clusters of arbitrary shapes, a k means clustering may produce low-quality results, because it only creates spherical groups. As this example shows, a general purpose method may fail on a given dataset and the detection of this failure is difficult to accomplish. Furthermore the presence of outliers can also significantly distort results of statistical routines.

Therefore the following discussions propose possible combinations of statistical methods and information visualization techniques for clustering, outlier detection and dimension reduction that may compensate the drawbacks of individual approaches.

8.3.1 Grouping of Data Items

The most popular statistical routine in data mining applications is clustering, that introduces partitions of the dataset. The detected groups can be seen as a simplification of the data that allows an easier interpretation of the main patterns. But clustering results are also used to create clearer visualizations and a reduction of data items for time consuming calculations.

Consequently a clustering algorithm can introduce a meaningful division of the data into groups. But a visual verification of these partitions is crucial, because the introduced clusters may not be appropriate for the structures in the data. As general purpose clustering algorithms suffer that the number of clusters has to be set and/or the created clusters show a specific shape, their results could be manipulated to achieve a better fit of the real groups in the data.

A visualization system that captures both the high dimensionality of the data as well as local features has to be applied to provide a user interface for the exploration and manipulation of clustering results. In the scope of this work the use of parallel coordinates and scatterplots is suggested. While the latter makes the intuitive investigation of two dimensional features possible, a parallel coordinates view illustrates all dimensions of a dataset. Furthermore dimension reduction techniques are applied to map the data items in a two dimensional space, which is visualized by a scatterplot. This allows a validation of the quality of the introduced partitions and represents a user interface for multivariate modifications of the clustering result.

As operations that adapt the introduced partitions clusters can be split, merged or deleted. Furthermore a cluster can be selected for a subclustering procedure, where only the data items of the chosen partition are considered for a clustering. But also cluster centers can be repositioned and objects can be reassigned to the cluster with the nearest center. After those interactions took place a reclustering based on the adapted clustering result can be initiated to improve the solution. Thus an interactive information exchange between a computational routine and the user's interaction is established, which is a significantly improved system in comparison to information visualization applications that only allow the exploration of clustering results. Because now the user is not restricted to the initiation of interactions, that are based on the perceived (mostly lower dimensional) features, also a routine that considers patterns in data space can be interactively applied.

8.3.2 Dimension Reduction and Feature Subset Selection

Based on a user defined similarity measure between attributes, a clustering procedure can be initiated to introduce groups of similar variables. For this purpose a hierarchical clustering approach is adequate, because it allows the interactive modification of the group number. The established hierarchy of dimension relationships can be used as starting point for an interactive feature subset selection application that can also be combined with dimension reduction techniques. Thus a visualization of the dendrogram structure allows an interactive exploration of the clustering result. Dimensions that are represented by a selected node can be illustrated by parallel coordinates and serve as decision guidance for the feature selection. Additionally if for a group no representative dimension can be chosen, a dimension reduction approach is available. Consequently the main information represented by the cluster of dimensions is captured by a small number of artificial attributes, which can be selected.

Because a clustering approach, that is not adapted to a specific kind of data, can produce arbitrarily bad fits to the structure of the dimension coherences, it is crucial that the user examines the achieved grouping. To accomplish this, the most characteristic attributes of the clusters as well as those variables that can also be assigned to different partitions have to be determined. Dimensions of the first category may be those candidates that are chosen by the user to represent

other dimensions for further operations. Other attributes may be of minor importance from the clustering point of view. Nevertheless the user can incorporate her/his knowledge in the process and can also choose those attributes, if they represent crucial information.

Consequently this approach combines a statistical procedure to create an initial solution for the subset selection problem by introducing groups of dimensions for which a representative attribute can be chosen. But also the input of the user is required to choose the correct subset by visually validating the quality of the clustering. If the dimension clusters are visually heterogeneous, a new clustering can be tested to achieve a better result or a dimension reduction approach can be applied.

8.3.3 Multivariate Outlier Detection

In contrast to the detection of clusters, where similar data items are grouped, the identification of outliers searches for objects that deviate from the main behaviour of the data. Consequently this subset of data points may be heterogeneous. As browsing and selection techniques of information visualization applications only highlight data items showing similar properties, this technique is not adequate to detect high dimensional outliers. Consequently a statistical routine could be used again as an initial solution for the task. These methods provide parameters that can be modified to steer the number of identified outlying objects. Thus it is crucial to have a visual feedback that allows the interactive determination of the optimal parameter settings. Different linked visualizations, which are also able to apply dimension reduction techniques, could be used again to help the user accomplishing this task. A projection of the data items on a low dimensional subspace to realize a scatterplot illustration is especially helpful, because this approach allows the verification, whether the detected objects are at the border of the data cloud or deviate from the main groups in the dataset. Thus a validation of the statistical outlier detection is achieved and data items that are wrongly marked can also be manually deselected, which enhances the quality of the outlier detection.

The application of multivariate outlier exclusion is crucial for non-robust statistical routines that calculate misleading results in the presence of outlying objects. In contrast to that visualizations of large datasets mainly stress the major patterns in the data. Thus it is essential to detect possibly outlying data items to accentuate their representations.

8.4 Proof of Concept Cases

In two proof of concept cases the benefits of the interactive collaboration of statistical routines and visualizations have been demonstrated. Therefore five tools have been realized in a sample application, which address different tasks of the data mining process. These tools are a

visual transformation application, interactive outlier detection, interactive dimension reduction, interactive clustering and the visual group analysis. In this section the functionality of these applications and their benefits for the user are outlined.

The visual transformation tool provides a set of linear mappings that standardize or project data values to a certain value range. Additionally non linear functions like the squareroot and logarithmic transformation allow the adaptation of the distribution of the dimension values, which is crucial for applications like the distribution based outlier detection. For an interactive visual verification, if the data items were mapped to a theoretic distribution, a scatterplot shows the quantiles of either the normal or the uniform distribution versus the ordered transformed data items. If the plotted objects are positioned near to a line which is laid through the first and third quartile of these distributions, the data items match the given distribution. Thus an immediate and easy validation of the usefulness of a transformation can be achieved. In figure 6.8 this concept is demonstrated, where finally the squareroot transformation created the mapping of the data to a normal distributed sample.

The interactive outlier detection calculates the robust distances and Mahalanobis distances for all data items and plots these values in a scatterplot (figure 6.9). The user is able to steer the number of detected outliers by changing the decision boundary for this classification task. Additionally a linked scatterplot view of the data items projected on the first two robust principal components allows the interactive validation, whether the identified data items are located far away from the center of the data cloud (figure 6.11). Consequently the user has the possibility to detect items that deviate from the average behaviour of the majority of the data interactively and can verify the computational results immediately.

The interactive dimension reduction tool allows a visual and a computational examination of the attribute relationships. By using a hierarchical clustering based on the correlation matrix created from the dataset a dimension grouping and also a variable ordering is introduced. The latter allows a clearer illustration of the dataset in parallel coordinates in comparison to the order of the dimensions according to their occurrence in the data. This issue is illustrated in figure 6.2. Additional flipping operations allow the user to reduce cluttering caused by negative correlation patterns.

The hierarchical clustering also creates a dendrogram structure (figure 6.1), which serves as an interface for the interactive exploration of the dimension groups. If a node is selected the attributes of the cluster are shown in a parallel coordinates plot (figure 6.3). For the methodical exploration of these groups it is recommended to choose a level of the dendrogram and thus also a number of clusters. Afterwards each cluster is scrutinized to detect attributes representing the same patterns. To accomplish this, the patterns in the parallel coordinates as well as the correlation measures and the explained variance of the first principal component computed from selected attributes are applied. If similar variables are found a representative dimension for them can be chosen.

To find groups in multivariate datasets clustering is a crucial task. For the proof of concept cases an interactive *k* means clustering approach was realized. After initial partitions have been established the user can reposition cluster centers within a 2D scatterplot, which illustrates the data items mapped on the first and the second principal component. The modifications are projected back into the data space and thus are high dimensional interactions. Additionally a linked parallel coordinate view also shows the clustering result and the modifications of the cluster centers. The tool allows to reassign the data items to the cluster with the nearest repositioned cluster center and a reclustering based on the adaptations introduced by the user. In the figures 6.4, 6.5, and 6.6 the intermediate results of an interactive clustering workflow are illustrated.

The tool showed that the input of the user can efficiently improve the cluster result with respect to the objective function of the clustering or to a user defined quality measure. Thus the partitions can be interactively fitted to the structure of the data, which is crucial to overcome disadvantages of *k* means because of its simple concept. Also functional relationships between the principal components and the original data dimension can be explored by the interaction possibilities for the cluster centers.

Finally a visual group analysis tool is provided, where the mean vector of the groups are compared with the center of the dataset. In this visualization for each dimension the relative deviation between these two location parameters are expressed by bar diagrams. This provides a fast visual and also numerical summary of the main characteristics of groups. It also facilitates the comparison between clusters and consequently helps to comprehend, why data items where assigned to the same group. An example of this analysis is given in figure 6.7.

As it is necessary to apply each of these tools in any possible order, each of these described applications can be started based on the results of the previous step in the workflow. Consequently transformations of the data can be applied before each clustering, outlier detection or dimension reduction as well as visual analysis can be accomplished after any classification process. Thus this flexible workflow is a further valorisation of this sample application.

8.5 Library for Statistical Functionality for Visualization

For the determination of statistical functionality that is of high importance for information visualization applications leading software packages like SpotFire [7], Miner3D [4] or GGobi [2] were examined. Also publications of recent years, that discuss the integration of computational approaches in the visual data mining process were analysed. This research showed that the majority of the applied algorithms are concerned with clustering and dimension reduction. But also the use of transformations to prepare the data for further procedures was demonstrated especially in GGobi. Besides of these main tasks also standard calculations like statistical moments and correlation measures were common.

In the scope of this work also a stronger integration of robust methods should be obtained. This is shown by implementing robust estimators for the location and the spread of a set of data values as well as by providing the calculation of robust correlation measures. But the main application, which demonstrates the capabilities of robustness, is the statistical outlier detection, which introduces a measure of outlyingness for each data item.

Furthermore the concept of fuzzyness is considered, because decisions made in the real world, from where the data comes from, are rarely reduced to yes/no answers. The fuzzy k means clustering was implemented, to show that it is not possible to assign each data item strictly to one cluster. Thus a degree of uncertainty is introduced to differentiate between objects that are near a cluster center and those data points that are located at cluster boundaries.

After introducing the main categories of functionality that is made available by the library, the remainder of this section gives an overview of the provided routines.

8.5.1 Transformations and Moments

Transformations can be seen as mappings of the data values to a certain interval or as modifications of the distribution of a set of values. The former application is useful to prepare a dataset for clustering, so that each dimension has the same range of values, which avoids that one attribute has a stronger influence on the distance calculations in the group finding process. The latter is of importance for statistical routines such as the distribution-based high dimensional outlier detection, which can only be applied on data from a multivariate elliptical distribution. For those distribution mappings the statistics library provides linear, logarithmic, exponential and squareroot transformations.

As transformations are applied on single dimensions separately also statistical moments are in general calculated for attributes of the dataset. Classic as well as robust estimates for the location (arithmetic mean, median, α - trimmed mean) and the spread (standard deviation, median of absolute deviations, inter quartile range) are provided.

8.5.2 Correlations and Covariances

To analyse the coherence between two variables three correlation measures are implemented. The classic Pearson correlation, which is biased if outliers are present, and the robust Spearman and Kendall correlations can be calculated. The two robust estimates do not only detect linear relationships but also exponential and logarithmic dependencies between dimensions.

A rough estimate for the shape of the multidimensional data cloud is given by the covariance matrix, which is a symmetric matrix holding the variances of the attributes in the main diagonal and the covariances between the dimensions in the off diagonal entries. As the covariance matrix describes the data as a hyperellipsoid it can be applied to integrate the shape of the data distribution into the distance calculation. This is achieved by the Mahalanobis distance. If a robust calculation scheme for the covariance matrix like the minimum covariance determinant estimator [101] is applied, this concept can be used to calculate robust distances that assign high values to data items that strongly deviate from the majority of data items.

8.5.3 Clustering and Dimension Reduction

For the division of datasets into partitions the popular clustering procedures k means and fuzzy k means were implemented. While the first algorithm introduces a hard cluster structure, where each data item is assigned to exactly one group, the fuzzy approach calculates memberships that indicate to which degree an object belongs to a given cluster. Thereby the sum of the memberships for a data item always accounts 1. Additionally a hierarchical clustering approach based on the correlation matrix is realized to introduce groups of dimensions. This routine can be used as basis for an interactive feature subset selection application.

As dimension reduction routine the principal component analysis (PCA) is provided. It is based on the covariance matrix calculation. Thus also a robust PCA can be accomplished by using the MCD estimate as covariance matrix.

8.5.4 Distributions and Statistical Tests

As theoretical distributions the normal, log normal, exponential, uniform and chi-squared distribution are realized. For each of these distributions values of the probability density function (pdf) and of the cumulative distribution function as well as quantiles and random numbers are available. Additionally a Kolmogorov-Smirnov test can be performed to validate, whether a set of values comes from these theoretical distributions. Furthermore the invocation of tests, whether two attributes show the same distribution, is possible.

8.5.5 Regression

A least squares linear regression that predicts the values of a dependent variable based on a set of independent attributes is implemented. This model fitting approach is convenient for the identification of functional dependencies between dimensions of a dataset.

8.6 Implementation

The implementation of the statistics library is aimed to operate on large data sets holding millions of data items. Therefore special attention was paid to process large arrays of data values as fast as possible. To achieve this goal an implementation in the language C++ was chosen, which provides efficient pointer operations. The work with C++ also demands a concept for a failure safe usage of the library. To accomplish this issue all routines return a bool variable indicating false, if the functionality could not be executed correctly. Furthermore the parameters that are passed to the methods apply to a scheme so that function calls of different procedures have a similar structure and thus are intuitive to use. The first category of parameters concerns variables into which the results are written. Afterwards data specific information like dimension values or mean vectors have to be set. The final class of parameters is concerned with the properties of the applied algorithm itself. Examples are the number of clusters for a k means clustering, or the robustness factor for an MCD covariance matrix estimator.

To ease the integration of statistical routines into information visualization applications the definition of an adequate interface is crucial. Thus so called hooks of interaction have been realized. These special function calls enable the immediate recalculation of statistical facts like correlations and moments for subsets of the data items. This is important for applications, where numerical summaries of selected data items are requested. Those summaries have to be updated, if the selection changes or if details-on-demand actions are initiated. Besides these standard adaptations also task specific interface extensions had to be included. Based on the statistical routine and its visualization several interaction techniques can be specified. Consequently the implications of the user actions have to be translated into parameter settings for the computational algorithm in the statistical library to adapt its result. This was accomplished on the basis of the k means clustering. A typical visualization of a result of this partitioning procedure involves the representation of the data items in colours according to their cluster membership and the emphasis of the cluster centers. The latter can be used to manipulate the clustering result by repositioning its centers or selecting clusters to initiate merge and splitting operations. Those complex adaptations have to be covered by the hooks of interactions and reformulated into ordinary function calls to allow a reclustering based on the user's input.

As functionalities such as the principal component analysis and the robust distance calculation require matrix inversion and determinant evaluation, implementations for these operations were integrated from the numerical recipes in C [95]. Also correlation computations, the realization of theoretic distributions and the Kolmogorov-Smirnov test build up on the fast and stable routines of this repository of basic numerical procedures. Because of the integration of robust methods, which require the evaluation of a value, having a specified position in a sample, an efficient routine that returns the k smallest data value of an array was realized [30]. Empirical tests prove that this method is faster than the application of a quick sort procedure.

Chapter 9

Conclusions and Future work

The collaboration between statistical routines and information visualization is a growing working area. So far the majority of publications concentrates on the visual presentation of statistical results and its exploration. But as this work shows much more benefits can be achieved, if both fields are interactively connected.

The field of information visualization provides a multitude of interaction techniques that should also be used to influence statistical routines for grouping data items or dimensions as well as for finding outliers. Applications that try to combine statistics and visualization mostly lack the capabilities to convey the findings of visual data exploration to the data mining routines. Thus interactions have to be semantically adapted to the users needs and the possible interfaces of a statistical procedure. But the modifications by interacting with an information visualization application that appears as user interface for a data mining algorithm has also be translated into a reasonable parameter set for this algorithm. Also the creation of a protocol that can record all interaction steps and maybe also includes semantic information is an important task that has to be accomplished to reproduce results of a visual data exploration process.

Besides this outlook in the future work of combining visualization and statistics also fundamental steps of procedure are not yet fully integrated into visual data mining applications. The majority of statistical routines assumes that the data satisfies certain constraints with respect to its distribution or value range. Nevertheless this fact is not pointed out explicitly by publications that suggest the integration of statistical algorithms into a visual data exploration workflow. This work tried to pay attention to the task of data transformation, which is crucial for multidimensional operations like clustering, dimension reduction or multivariate outlier detection. Although this observation is a basic principle for using those algorithms, it should be brought to the user's mind by integrating corresponding options in an interface steering a statistical procedure, as well as a preparation of the data should be explained in further detail in the corresponding publications, because it has such a tremendous impact on the achieved results.

Concerning the statistical routines that are in use for visual data mining applications it is evident, that the concept of robustness has not yet been considered in large scale. Certainly robust algorithms introduce additional computational efforts. Nevertheless for the major classic algorithms a robust version should be provided to compare the results. Deviations in the solutions may lead to the discovery of interesting facts concerning the data, relativise the blind trust in the outcome of a statistical analysis and consequently encourage the user to examine and question the findings.

Also the process of a steady information exchange between the user's interaction and numerical summaries that validate or disprove the theses or the reasonableness of the actions made during the visual data exploration is not yet well established. Interaction possibilities could be made more efficient, if statistical information is provided, that represents a further guideline for the user. Numerical summaries that are accommodated to the type of a selection can assist to make decisions.

The introduced statistical library is one step towards this vision of the integration of basic statistical routines into visual data mining applications. To continue the exploration of the benefits of this collaboration concrete applications that use the capabilities of the implemented routines have to be created. This would allow the extension of the interface that focuses on the needs of information visualization techniques. Examples that could serve as starting points for this development have been discussed.

Acknowledgements

This work has been realized at the VRVis Research Center in Vienna, Austria (http://www.VRVis.at/), which is funded by the Austrian research program called Kplus, to support the basic research on visualization (http://www.VRVis.at/vis/).

Special thanks are given to Helwig Hauser, who gave me the opportunity to work on the combination of statistical routines and information visualization. His support and his vision how both sciences could collaborate made this work possible.

I also want to thank Peter Filzmoser for his input and help concerning statistical routines and their integration in visualization applications. His ideas for the improvement of data exploration by visual techniques were a crucial contribution to increase the quality of this work.

Additional thanks go to Harald Piringer, whose dedication for improving implementations and whose expert knowledge concerning the work with large datasets helped to realize the assembly of statistical routines in a C++ library.

Bibliography

[1] 3-d fluid flow data, (XmdV data archive: http://davis.wpi.edu/xmdv/datasets/uvw.html).

[2] GGobi - Data visualization system (http://www.ggobi.org/, last visited 2007-01-29).

[3] GTK+ The GIMP Toolkit (http://www.gtk.org/, last visited 2007-01-29).

[4] Miner3D (http://www.miner3d.com/, last visited 2007-01-29).

[5] The R Project for Statistical Computing (http://www.r-project.org/, last visited 2007-01-29).

[6] The source for Java developers (http://java.sun.com/, last visited 2007-01-29).

[7] SpotFire Decisionsite (http://www.spotfire.com/products, last visited 2007-01-29).

[8] Mokhtar B. Abdullah. On a Robust Correlation Coefficient. *The Statistician*, 39(4):455–460, 1990.

[9] Charn C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data 2001, Santa Barbara, California, United States, May 21–24, 2001*, pages 37–46, 2001.

[10] Christopher Ahlberg. Spotfire: An information exploration environment. *SIGMOD Record*, 25(4):25–29, 1996.

[11] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Human Factors in Computing Systems. Conference Proceedings CHI'94*, pages 313–317, New York, NY, USA, 1994. ACM, ACM Press.

[12] Mihael Ankerst, Stefan Berchtold, and Daniel A. Keim Mihael. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings IEEE Symposium on Information Visualization 1998*, pages 52–60. IEEE, 1998.

[13] Daniel Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, January 1985.

[14] Vic Barnett and Toby Lewis. *Outliers in statistical data*. John Wiley and Sons Ltd, New York, 1984.

[15] Jeff Beddow. Shape coding of multidimensional data on a microcomputer display. In *IEEE Visualization '90 Proceedings*, pages 238–246, 1990.

[16] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[17] James Christian Bezdek. *Fuzzy mathematics in pattern classification*. PhD thesis, Cornell University, 1973.

[18] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. GTM: A principled alternative to the self-organizing map. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 354–360. The MIT Press, Cambridge, MA, 1997.

[19] Justine Blackmore and Risto Miikkulainen. Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map. Technical Report TR AI92–192, University of Texas at Austin, Austin, TX, 1992.

[20] Paul S. Bradley and Usama M. Fayyad. Refining initial points for K-Means clustering. In *Proc. 15th International Conf. on Machine Learning*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998.

[21] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In Weidong Chen, Jeffery Naughton, and Philip A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data: May 16–18, 2000, Dallas, Texas*, volume 29(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pages 93–104, pub-ACM:adr, 2000. ACM Press.

[22] Andreas Buja and Daniel Asimov. Grand tour methods: an outline. In *Proceedings of the Seventeenth Symposium on the interface of computer sciences and statistics on Computer science and statistics*, pages 63–67, New York, NY, USA, 1986. Elsevier North-Holland, Inc.

[23] Miguel A. Carreira-Perpinan. A review of dimension reduction techniques. Technical report, University of Sheffield, January 1997.

[24] Matthew Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *IEEE Visualization '96 Proceedings*, pages 127–132, 1996.

[25] Herman Chenroff. The use of faces to represent points in k-dimensional space. *Journal of the American Statistical Assoc.*, 68(342):361–368, 1973.

[26] William S Cleveland. Robust locally weighted regression and smoothing scatter plots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[27] William S. Cleveland. *The Elements of Graphing Data*. Hobart Press, New Jersey, U.S.A., 1994.

[28] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. Lawrence Erlbaum Associates, New Jersey, U.S.A., 1988.

[29] William J. Conover. *Practical Nonparametric Statistics (2nd Edition)*. John Wiley and Sons Ltd, New York, 1980.

[30] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (Second Edition)*. MIT Press, Cambridge, Massachusetts, 2001. Second Edition.

[31] Peter Dalgaard. *Introductory Statistics with R*. Springer Verlag, New York, 2002.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[33] Michael Dittenbach, Dieter Merkl, and Andreas Rauber. Growing hierarchical self-organizing map. In *Proceedings of the International Joint Conference on Neural Networks*, volume 6, pages 15–19, Piscataway, NJ, 2000. Technische Universitat Wien, IEEE.

[34] Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. In V. Skala, editor, *Journal of WSCG*, volume 10, pages 147–154, 2002.

[35] Jennifer G. Dy and Carla E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proc. 17th International Conf. on Machine Learning*, pages 247–254. Morgan Kaufmann, San Francisco, CA, 2000.

[36] Jennifer G. Dy and Carla E. Brodley. Visualization and interactive feature selection for unsupervised data. In *Proceedinmgs of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 360–364, 2000.

[37] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[38] Steven K. Feiner and Clifford Beshers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In Scott E. Hudson, editor, *User interface software and technology*, pages 76–83. ACM Press, Oktober 1990. Conference: Proceedings of the the third annual ACM SIGGRAPH symposium, Snowbird, UT.

[39] Peter W. Frey and David J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6:161–182, 1991.

[40] Bernd Fritzke. Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Netw.*, 7(9):1441–1460, 1994.

[41] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *IEEE Visualization '99 Proceedings*, pages 43–50, 1999.

[42] Keinosuke Fukunaga. Statistical pattern recognition. In *Handbook of Pattern Recognition and Computer Vision*, page Chapter I:2, 1997.

[43] George W. Furnas. Generalized fisheye views. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'86*, pages 16–23, 1986.

[44] George W. Furnas and Andreas Buja. Prosection views: Dimensional inference through sections and projections, October 03 1994.

[45] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84, New York, NY, USA, 1998. ACM Press.

[46] Diansheng Guo. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization*, 2(4):232–246, 2003.

[47] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Special issue on special feature):1157–1182, 2003.

[48] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster validity methods: part I. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 31(2):40–45, June 2002.

[49] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering validity checking methods: Part II. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 31(3):19–27, 2002.

[50] John A. Hartigan. *Clustering Algorithms*. John Wiley and Sons Ltd, New York, 1975.

[51] John A. Hartigan and M. A. Wong. Algorithm AS136. A K-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[52] John Haslett and Adrian E. Raftery. Space-time modelling with long-memory dependence: Assessing Ireland's wind power resource. *Applied Statistics*, 38(1):1–50, 1989.

[53] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization (INFOVIS'02)*, pages 127–130. IEEE Computer Society, 2002.

[54] Alexander Hinneburg and Daniel A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, *Proceedings of the Twenty-fifth International Conference on Very Large Databases, Edinburgh, Scotland, UK, 7–10 September, 1999*, pages 506–517, Los Altos, CA 94022, USA, 1999. Morgan Kaufmann Publishers.

[55] Alexander Hinnenburg, Daniel A. Keim, and Markus Wawryniuk. HD-eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications*, 19(5):22–31, September 1999.

[56] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.

[57] Peter J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[58] Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50(4), 1996.

[59] Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

[60] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 361–378. IEEE Computer Society Press, 1990.

[61] Edward J. Jackson. *A User's Guide to Principal Components*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley and Sons Ltd, New York, 1991.

[62] Anil K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[63] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization (INFOVIS'05)*, pages 125–132. IEEE Computer Society, 2005.

[64] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 4 edition, 1998.

[65] Theodore Johnson, Ivy Kwok, and Raymond T. Ng. Fast computation of 2-dimensional depth contours. In *KDD*, pages 224–228, 1998.

[66] Ian T. Jolliffe. Principal component analysis. In *Principal Component Analysis*. Springer Verlag, New York, 1986.

[67] MC Jones and Robin Sibson. What is projection pursuit? (with discussion). *Journal of the Royal Statistical Society Series A*, 150(1):1–37, 1987.

[68] Eser Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-01)*, pages 107–116, 2001.

[69] Leonard Kaufman. Finding groups in data: an introduction to cluster analysis. In *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[70] Daniel Keim, Ming C. Hao, Julian Ladisch, Meichun Hsu, and Umeshwar Dayal. Pixel bar charts: A new technique for visualizing large multi-attribute data sets without aggregation. Technical Report HPL-2001-92, Hewlett Packard Laboratories, April 25 2001.

[71] Daniel A. Keim. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, 5(1):58–77, 1996.

[72] Kathleen M. Kerr and Gary A. Churchill. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments, September 21 2000.

[73] Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*.

[74] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB '98)*, pages 392–403, East Sussex - San Francisco, August 1998. Morgan Kaufmann.

[75] Kurt Koffka. *Principles of Gestalt Psychology*. Harcourt, Brace and World, New York.

[76] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990.

[77] Joseph B. Kruskal and Myron Wish. *Multidimensional Scaling*. Sage Piblications, Beverly Hills, California, U.S.A., 1978.

[78] Jeffrey LeBlanc, Matthew O. Ward, and Norman Wittels. Exploring N-dimensional databases. In *IEEE Visualization '90 Proceedings*, pages 230–237, 1990.

[79] Michael D. Lee, Rachel E. Reilly, and Marcus A. Butavicius. An empirical evaluation of chernoff faces, star glyphs, and spatial visualisations for binary data. In Tim Pattison and Bruce Thomas, editors, *Conferences in Research and Practice in Information Technology*, volume 24, pages 1–10, Adelaide, Australia, 2003. Australian Computer Society.

[80] Richard A. Olshen Leo Breiman, Jerome Friedman and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall / CRC Press, 1984.

[81] Jianchang Mao and Anil K. Jain. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks*, 7(1):16–29, January 1996.

[82] Douglas C. Montgomery and George C. Runger. *Applied statistics and probability for engineers (3rd Edition)*. John Wiley and Sons Ltd, New York, 2003.

[83] Christopher J. Morris and David S. Ebert. An experimental analysis of the effectiveness of features in chernoff faces, January 21 2000.

[84] Alistair Morrison, Greg Ross, and Matthew Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *Proc. IEEE Symp. Information Visualization, InfoVis*, pages 152–158. IEEE Computer Society, 2002.

[85] Casper Thomsen Nicolaj Sondberg-Madsen and Jose M. Pena. Unsupervised feature subset selection. In *Proceedings on the Workshop on Probabilistic Graphical Models for Classification*, pages 71–82. Proceedings on the Workshop on Probabilistic Graphical Models for Classification (in ECML/PKDD-03), 2003.

[86] Erkki Oja and Zhijian Yuan. The fastICA algorithm revisited – convergence analysis. *IEEE Transactions on Neural Networks*, 17(6):1370–1381, 2007.

[87] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In José R. Dorronsoro, editor, *Artificial Neural Networks - ICANN 2002, International Conference, Madrid, Spain, August 28-30, 2002, Proceedings*, volume 2415 of *Lecture Notes in Computer Science*, pages 871–876. Springer Verlag, 2002.

[88] Dan Pelleg and Andrew Moore. *X*-means: Extending *K*-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[89] Dan Pelleg and Andrew W. Moore. Accelerating exact k -means algorithms with geometric reasoning. In *KDD*, pages 277–281, 1999.

[90] Ronald M. Pickett and Georges G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Conference on Systems, Man, and Cybernetics. Proceedings of the 1988 IEEE International*, volume 1, pages 514–519. IEEE Press, 1988.

[91] Harald Piringer. Design guidelines and concepts of the infovis library. Technical Report TR-VRVis-2005-034, currently in submission, VRVis Research Center, 2005.

[92] Harald Piringer, Robert Kosara, and Helwig Hauser. Interactive focus+context visualization with linked 2D/3D scatterplots, May 05 2004.

[93] Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. A visualization technique for self-organizing maps with vector fields to obtain the cluster structure at desired levels of detail. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 1558–1563, Montreal, Canada, July 31–August 5 2005. IEEE Computer Society.

[94] Georg Pölzlbauer, Andreas Rauber, and Michael Dittenbach. Advanced visualization techniques for self-organizing maps with graph-based methods. In Jun Wang, Xiaofeng Liao, and Zhang Yi, editors, *Proceedings of the Second International Symposium on Neural Networks (ISNN'05)*, volume 3497 of *Lecture Notes in Computer Science*, pages 75–80. Springer-Verlag, 2005.

[95] William H. Press, Saul A. Teukolsky, William T. Vetterling, and title = Brian P. Flannery.

[96] Sridhar Ramaswamy, Rajeev Rastogi, Kyuseok Shim, and Taejon Korea. Efficient algorithms for mining outliers from large data sets, March 03 0.

[97] Matt Rasmussen and George Karypis. gCLUTO – an interactive clustering, visualization, and, June 29 2004.

[98] Alvin C. Rencher. *Methods of Multivariate Analysis (Second Edition)*. John Wiley and Sons Ltd, New York, 2002.

[99] Peter J. Rousseeuw. Least median of squares regression. *J. Amer. Statis. Ass.*, 79(388):871–880, 1984.

[100] Peter J. Rousseeuw. Multivariate estimation with high breakdown point. In W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, editors, *Mathematical Statistics and Applications*, volume B, pages 283–297, Budapest, 1985. Akadémiai Kiadó. Fourth Pannonian Symposium on Mathematical Statistics and Probability, Bad Tatzmannsdorf, Austria, September 4-10, 1983.

[101] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator, 1998.

[102] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons Ltd, New York, 1987.

[103] Ida Ruts and Peter J. Rousseeuw. Computing depth contours of bivariate point clouds. *Computational Statistics & Data Analysis*, 23(1):153–168, 1996.

[104] Manojit Sarkar and Marc H. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12):73–84, 1994.

[105] Dave Schreiner. *OpenGL(R) Reference Manual: The Official Reference Document to OpenGL, Version 1.4*. Addison-Wesley Longman Publishing Co., Inc., Boston, 2004.

[106] Jinwook Seo and Ben Shneiderman. Understanding hierarchical clustering results by interactive exploration of dendrograms: A case study with genomic microarray data, January 21 2003.

[107] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 65–72. IEEE Computer Society, 2004.

[108] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[109] Farrell E. J. Goldwyn R. M. Friedman H. P. Siegel, J. H. The surgical implication of physiologic patterns in myocardial infarction shock. *Surgery*, 72:126–141, 1972.

[110] Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy : The principles and practice of numerical classification*. W. H. Freeman and Company, San Francisco, 1973.

[111] T. C. Sprenger, R. Brunella, and Markus H. Gross. H-BLOB: a hierarchical visual clustering method using implicit surfaces. In *IEEE Visualization '00 Proceedings*, pages 61–68, 2000.

[112] Deborah F. Swayne, Andreas Buja, and Duncan Temple Lang. Exploratory visual analysis of graphs in GGobi, March 05 2003.

[113] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Longman Publishing Co., Inc., Boston, 1977.

[114] Alfred Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, October 08 1999.

[115] Alfred Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proceedings Workshop on Self-Organizing Maps (WSOM 2003)*, pages 225–230, 2003.

[116] Alfred Ultsch. U*-matrix: a tool to visualize clusters in high dimensional data. Technical report, Dept. of Mathematics and Computer Science, Philipps-University Marburg,, 2003.

[117] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586, May 2000.

[118] Ellen M. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing and Management: an International Journal*, 22(6):465–476, 1986.

[119] Edward J. Wegman and Qiang Luo. High dimensional clustering using parallel coordinates and the grand tour. Technical Report 124, Center for Computational Statistics, George Mason University, Fairfax, Virginia 22030, U.S.A., April 1996.

[120] Jing Yang, Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *IEEE Symposium on Information Visualization, 2003*, pages 105–112. IEEE Computer Society, 2003.

[121] Jing Yang, Matthew O. Ward, Elke A. Rundensteiner, and Shiping Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of the symposium on Data visualisation*, pages 019–028. Eurographics Association, 2003.

[122] Jing Yang, Matthew O. Ward, Elke A. Rundensteiner, and Anilkumar Patro. Interring: a visual interface for navigating and manipulating hierarchies. *Information Visualization*, 2(1):16–30, 2003.

[123] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114, New York, NY, USA, 1996. ACM Press.