LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Friedrich Leisch

Neighborhood Graphs, Stripes and Shadow Plots for Cluster Visualization

Technical Report Number 061, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Neighborhood Graphs, Stripes and Shadow Plots for Cluster Visualization

Friedrich Leisch

Institut für Statistik, Ludwig-Maximilians-Universität München
Ludwigstrasse 33, 80539 Munich, Germany

This is a preprint of an article that has been accepted for publication in
Statistics and Computing.
Please use the journal version for citation.

Abstract

Centroid-based partitioning cluster analysis is a popular method for segmenting data into more homogeneous subgroups. Visualization can help tremendously to understand the positions of these subgroups relative to each other in higher dimensional spaces and to assess the quality of partitions. In this paper we present several improvements on existing cluster displays using neighborhood graphs with edge weights based on cluster separation and convex hulls of inner and outer cluster regions. A new display called shadow-stars can be used to diagnose pairwise cluster separation with respect to the distribution of the original data. Artificial data and two case studies with real data are used to demonstrate the techniques.

Key Words: cluster analysis, partition, centroid, convex hull, R.

1 Introduction

The goal of cluster analysis is to either find homogeneous subgroups of the data, which in the best of all cases in turn are as different as possible from each other; or to impose an artificial grouping on the data. In any case we want to increase our understanding of the data by a divide & conquer approach which partitions a potentially complex and large data set into segments (i.e., clusters) that are easier to understand or handle.

Data visualization can help a lot to understand multivariate data structures, hence it is no surprise that cluster analysis and data visualization often go hand in hand. Standard textbooks on cluster analysis like Gordon (1999) or Everitt et al. (2001) are full of figures, see also Leisch (2008) for a recent survey on cluster visualization. Results from partitioning cluster analysis can be visualized by projecting the data into 2-dimensional space (e.g., CLUSPLOT, Pison et al., 1999). Cluster membership in the projection is usually represented by different colors and glyphs, or by dividing clusters onto several panels of a Trellis display (Becker et al., 1996). In addition, silhouette plots (Rousseeuw, 1987) are a popular tool for diagnosing the quality of a partition. Parts of the popularity of self-organizing feature maps (Kohonen, 1989) with practitioners in various fields can be explained by the fact that the results can be "easily" visualized.

In this paper we introduce several improvements and modifications of existing cluster visualization techniques, and propose a new diagnostic display we call shadow-stars. The exact choice of distance measure or partitioning cluster algorithm is not important. The only condition is that it is centroid-based, i.e., clusters are represented by prototypes and data points are assigned to the cluster corresponding to the closest prototype. Many popular clustering algorithms like k-means (MacQueen, 1967; Hartigan and Wong, 1979), partitioning around medoids (PAM, Kaufman and Rousseeuw, 1990) or neural gas (Martinetz and Schulten, 1994) fall into

this category.

All methods introduced in this paper have been implemented in the statistical computing environment R (R Development Core Team, 2008) and will be released as part of the R extension package `flexclust` (Leisch, 2006) on the Comprehensive R Archive Network (CRAN, `http://cran.R-project.org`) under the terms of the GPL.

The rest of this paper is organized as follows: Section 2 give a reminder of centroid-based cluster analysis, neighborhood graphs and shadow values, which form the basis for the following sections. Sections 3– 6 introduce several new ways to visualize and diagnose cluster solutions. Stripes plots directly show the distance of data points to cluster centroids, while shadow plots relate the distance to the closest and second-closest centroid. Shadow stars can be used to assess which cluster's are close to each other. Convex cluster hulls allow to shade cluster regions for non-elliptical cluster shapes. These new graphical technices are demonstrated on data from automobile marketing in Section 7 and data from German parliamentary elections in Section 8.

2 Neighborhood Graphs and Shadow Values

Assume we are given a data set $X_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_n \in \mathbb{R}^p$ and a set of centroids $C_K = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, $\mathbf{c}_k \in \mathbb{R}^p$ which is the result of a centroid-based cluster analysis like K-means. Let $d(\mathbf{x}, \mathbf{y})$ denote a distance measure on $\mathbb{R}^p$ ($\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^p$), let

$$c(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{c} \in C_K} d(\mathbf{x}, \mathbf{c})$$

denote the centroid closest to $\mathbf{x}$, and

$$A_k = \{\mathbf{x}_n \in X_N \,|\, c(\mathbf{x}_n) = \mathbf{c}_k\}$$

be the set of all points where $\mathbf{c}_k$ is the closest centroid. For simplicity of notation we assume that all clusters are non-empty, such that $|A_k| > 0, \forall k = 1, \ldots, K$ (our software implementation automatically removes empty clusters accordingly). Most cluster algorithms will try to find a set of centroids C_K for fixed K such that the average distance

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\mathbf{x}_n, c(\mathbf{x}_n)) \rightarrow \min_{C_K},$$

of each point to the closest centroid is minimal. However, for the following it is not important whether such an optimum has actually been reached.

Leisch (2006) introduces a *neighborhood graph* of the centroids where each centroid forms a node and two nodes are connected by an edge if there exists at least one data point for which those two are closest and second closest, see also Martinetz and Schulten (1994). Let

$$\tilde{c}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{c} \in C_K \setminus \{c(\mathbf{x})\}} d(x, \mathbf{c})$$

denote the second-closest centroid to $\mathbf{x}$, further let

$$A_{ij} = \{\mathbf{x}_n \in X_N \,|\, c(\mathbf{x}_n) = \mathbf{c}_i,\, \tilde{c}(\mathbf{x}_n) = \mathbf{c}_j\}$$

be the set of all points where $\mathbf{c}_i$ is the closest centroid and $\mathbf{c}_j$ is second-closest. Now we define for each observation $\mathbf{x}$ its **shadow value** $s(\mathbf{x})$ as

$$s(\mathbf{x}) = \frac{2d(\mathbf{x}, c(\mathbf{x}))}{d(\mathbf{x}, c(\mathbf{x})) + d(x, \tilde{c}(\mathbf{x}))}$$

The name "shadow" was chosen because the shadow of an object is similar to its silhouette, and the shadow plots constructed below are similar both in spirit and interpretation to the well known silhouette plots (Rousseeuw, 1987). If $s(\mathbf{x})$ is close to 0, then the point is close to its cluster centroid. If $s(\mathbf{x})$ is close to 1, it is almost equidistant to the two centroids. Thus, a cluster that is well separated from all other clusters should have many points with small shadow values.

Another memory aid is that the shadow value of a point gives the relative location of the shadow the point casts upon the line connecting $c(\mathbf{x})$ and $\tilde{c}(\mathbf{x})$. A cluster with large shadow values "casts a large shadow on its neighbouring clusters", and hence is close to them.

The average shadow value of all points where cluster i is closest and j is second-closest can be used as a simple measure of cluster similarity:

$$s_{ij} = |A_i|^{-1} \sum_{x \in A_{ij}} s(\mathbf{x})$$

The denominator $|A_i|$ rather than $|A_{ij}|$ is used such that a small set A_{ij} consisting only of badly clustered points with large shadow values does not induce large cluster similarity. If $s_{ij} > 0$, then at least one data point in segment i has $\mathbf{c}_j$ as second-closest centroid and segments i and j

are neighbours. If s_{ij} is close to $|A_{ij}|/|A_i|$, then those points that are "between" segments i and j are almost equidistant to the two centroids. The graph with nodes $\mathbf{c}_k$ and edge weights s_{ij} is a directed graph, to simplify matters we use the corresponding undirected graph with average values of s_{ij} and s_{ji} as edge weights for the moment.

Figure 1 shows neighborhood graphs for two data sets with 5 Gaussian clusters each. The graphs in panels above each other are identical, the different cluster hulls will be explained in Section 6. The centers of the original 5 clusters are identical in both data sets, only the variance changes. Both data sets have been clustered using K-means with 7 centers. The "wrong" number of clusters was intentional to show the effect on the graph and get overlapping clusters.

Both graphs show the ring-like structure of the data, the thickness of the lines is proportional to the edge weights and clearly shows how well the corresponding clusters are separated. Triangular structures like in the left panel correspond to regions with almost uniform data density that are split into several clusters. Centroid pairs connected by one thick line and only thin lines to the rest of the graph like 2/7 and 3/6 in the right panel correspond to a well-separated data cluster that has wrongly been split into two clusters.

3 Stripes Plots

A simple, yet very effective plot for visualizing the distance of each point from its closest and second-closest cluster centroids is a stripes plot as shown in Figure 2. For each cluster $k = 1, \ldots, K$ we have a rectangular area, which we vertically divide into K smaller rectangles. First we draw a horizontal line segment hat height $(\mathbf{x}_n, c(x_n))$ for each observation $x_n \in A_k$. In addition, we plot a horizontal line segment for each observation $x_n \in A_{jk}$, $j = 1, \ldots, K$, æ $\neq k$ at height $d(\mathbf{x}_n, \mathbf{c}_k)$. These are the points which have cluster k as their second-best. The horizontal position within the rectangular area and the color (please use the online version of this article for colored figures) always mark the cluster $c(\mathbf{x}_n)$ of the observation.

E.g., have a look at cluster 1 in the top panel. The leftmost stripe corresponds to points that have been assigned to cluster 1. It is marked by a slightly darker background and a box around the stripe. Points in clusters 2 and 5 have cluster 1 as second-best centroid. These observations form the other two stripes within the rectangular area for cluster 1. Points in cluster 2 are farther away from cluster 1, while many points in cluster 5 have a similar distance to centroid 1 as points which have actually been assigned to cluster 1.

The overall impression of the top panel in Figure 2 is that no cluster is well separated from all others. We can also infer which one of the other clusters is close to it. The picture is different for the bottom panel: clusters 1, 4, and 5 are well separated from the rest (the lowest block of stripes is far away from the rest), while clusters 2/7 and 3/6 are close to each other, respectively.

Of course all this information is easier to see in Figure 1, however the stripes plot is dimension-independent and works well even for high-dimensional data where projections to 2d may fail. The implementation of the stripes plot in our software is very flexible. The user can zoom into the bars to see only distances from the closest cluster centroid of each point, or only see distances to closest and second-closest centroid. It is also possible to choose a categorical background variable for the color-coding. This gives a quick overview of how the classes in the background variable are distributed over the clusters, and if they are close to the centroid or far away (see Section 8 below).

4 Shadow Plots

Another way to visualize the separation are cluster silhouettes (Rousseeuw, 1987). The silhouette value of $\mathbf{x}$

$$\mathrm{sil}(\mathbf{x}) = \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max(a(\mathbf{x}), b(\mathbf{x}))}$$

is defined as the scaled difference between the average dissimilarity $a(\mathbf{x})$ of $\mathbf{x}$ to all points in its own cluster to the smallest average dissimilarity $b(\mathbf{x})$ to the points of a different cluster. For shadow values we get

$$\begin{aligned} 1 - s(\mathbf{x}) &= 1 - \frac{2d(\mathbf{x}, c(\mathbf{x}))}{d(x, c(\mathbf{x})) + d(\mathbf{x}, \tilde{c}(\mathbf{x}))} \\ &= \frac{d(\mathbf{x}, \tilde{c}(\mathbf{x})) - d(\mathbf{x}, c(\mathbf{x}))}{d(x, c(\mathbf{x})) + d(\mathbf{x}, \tilde{c}(\mathbf{x}))} \end{aligned}$$

and a plot of $1-s(x)$ can be used to approximate silhouettes. The main difference between silhouette values and shadow values is that we replace

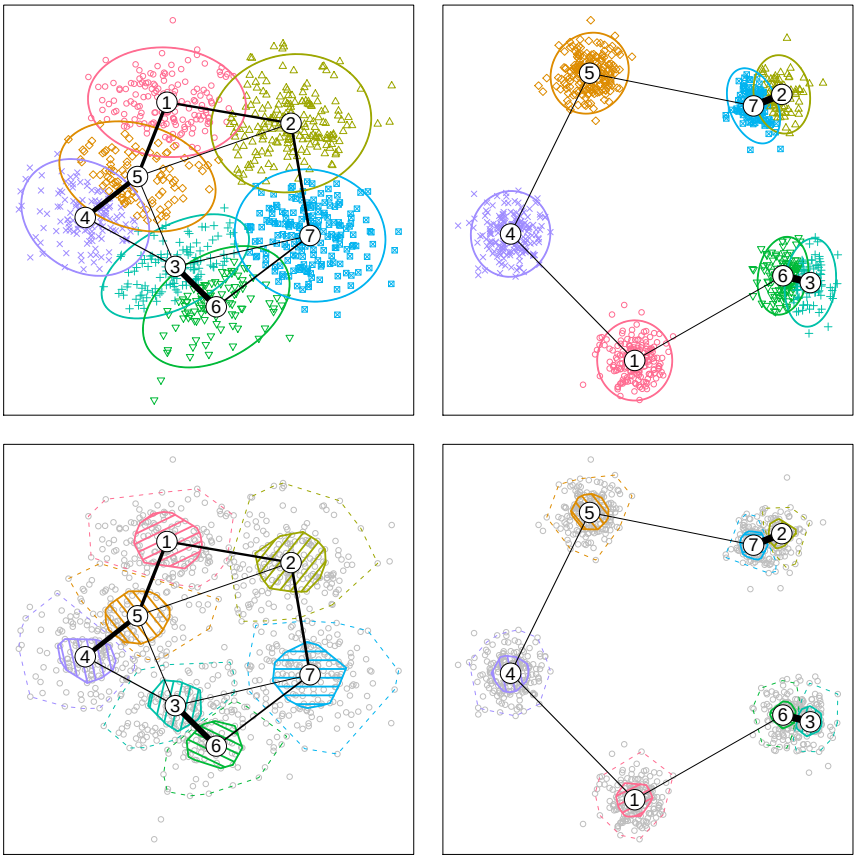

Figure 1: The neighborhood graphs for 7-cluster partitions of 5 Gaussians with poor (left top and bottom) and good (right top and bottom) separation.

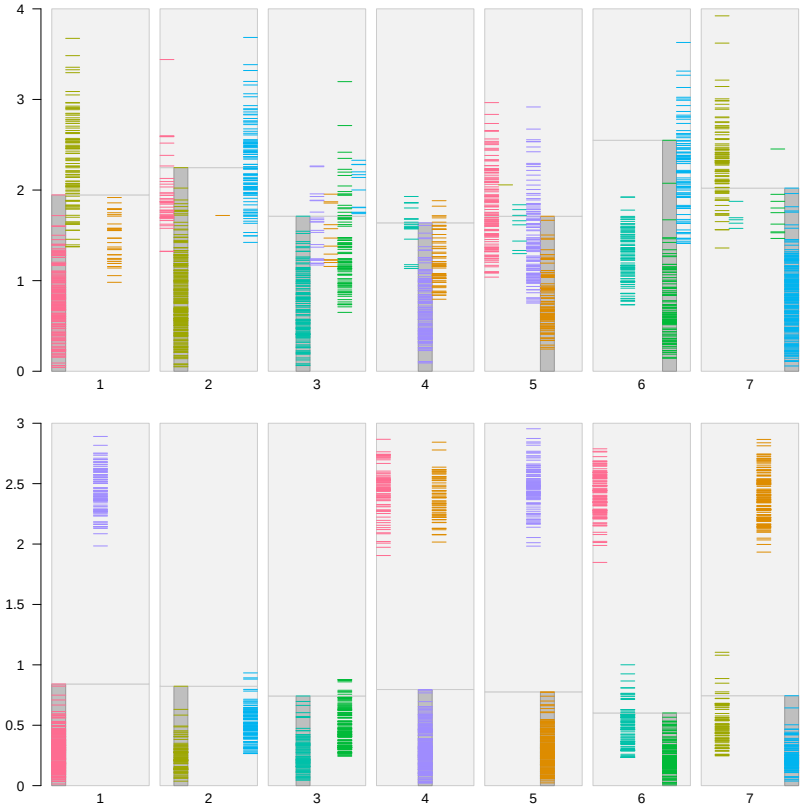

Figure 2: Stripes plots for the Gaussian data with poor (top) and good (bottom) separation. The seven clusters are on the x-axis, distance from centroid is on the y-axis.

average dissimilarities to points in a cluster by dissimilarities to point averages (=centroids). One advantage of shadow values is that they need $\mathcal{O}(NK)$ operations while silhouettes need $\mathcal{O}(N^2)$, and typically we have $K << N$. Package `flexclust` has implementations for both traditional silhouttes, as well as our new *shadow plots* which directly visualize the shadow values $s(x)$ (rather than $1 - s(x)$).

Figure 3 shows *shadow plots* for both partitions. The shadow values $s(x)$ in each cluster are sorted from high to low and plotted from left to right. To decrease memory consumption the actual values are interpolated for larger data sets. The width of the vertical stripe of each cluster is proportional to the size of the cluster. Clusters that are well separated should have many points with small shadow values $s(\mathbf{x})$, and the filled area below the curve should be small. The light rectangle behind the polygon marks the average shadow value of the cluster, hence the area of the rectangle in light color is the same as the area under the shadow line filled with dark color. This visual aid helps a lot to quickly compare the areas under the polygon.

The upper panel of Figure 3 shows the shadow plot for the Gaussian data with poor separation. Points are almost uniformly spread between the closest and second centroid, so the curves in all stripes go almost linearly from 1 to 0. The lower panel has 4 clusters with poor separation (2,3,6,7) looking similar to the shadows in the upper panel. The 3 clusters with good separation (1,4,5) clearly have a different shadow. The curve starts around 0.5 rather than one, the filled area consequently is much smaller.

5 Shadow-Stars

The main reason for our definition of shadow values is that they use the centroids as anchor points and have a geometric interpretation with respect to them. The distribution of the shadow values of all points in A_{ij} and A_{ji} gives an impression how connected or separated clusters i and j are. Points with $s(\mathbf{x}) \approx 0$ are close to the centroid, while points with $s(\mathbf{x}) \approx 1$ are equidistant to $c(\mathbf{x})$ and $\tilde{c}(\mathbf{x})$. This can be visualized in a new display, the *shadow-stars*: The centroids again are used as nodes in a graph, which are connected by stripe plots of shadow values. If A_{ij} is not empty, then a (virtual) line segment from $\mathbf{c}_i$ to $\mathbf{c}_j$ is drawn. The width of the line segments is proportional to the size of A_{ij}, such that larger groups of data can be identified more easily. On the half segment closer to $\mathbf{c}_i$ the shadow values in A_{ij} are drawn as line segments similar to the stripes plot. On the other half of the line segment the same is done with the shadow values of the points in A_{ji}. If cluster i is well separated from cluster j, then the shadow values will be small and the stripes will be concentrated close to the centroid. On the other hand, if the two clusters are not well separated, then the stripes of shadow values will either have an almost uniform distribution or be concentrated close to the middle of the edge.

Figure 4 shows shadow-stars for the Gaussian data. The clusters in the left panel are not well separated, the shadow values are distributed over the line segments connecting neighboring clusters. For the well separated clusters in the right panel the shadow values concentrate their mass close to the centroids. It can still be seen from the plot which cluster is next to which other cluster, but it is also obvious that they are far away from each other in terms of cluster separation.

Again, the software implementation is very flexible, and the user can specify arbitrary functions which are used to visualize the shadow values on each half-edge of the graph. If we use only the average shadow value as thickness of the edge, we get an asymmetric graph we call a *shadow skeleton*. If we use violin plots (Hintze and Nelson, 1998) for the distribution of the shadow values, we get *shadow violins*, see Figure 5.

6 Convex Cluster Hulls

When clustering non-Gaussian data and/or using distances other than Euclidean distance, spanning ellipses or confidence ellipses like in the CLUSPLOT by Pison et al. (1999) can be a misleading representation of cluster regions, because clusters may have arbitrary convex shapes, where the term convex is with respect to the distance measure used. If clusters are projected into 2-dimensional space, then bagplots (Rousseeuw et al., 1999) can be used as a nonparametric alternative to ellipses. The main challenge in constructing the equivalent of a box & whisker plot for 2 dimensions is that $\mathbb{R}^2$ has no total ordering. Bagplots solve this by imposing one possible ordering onto data, such

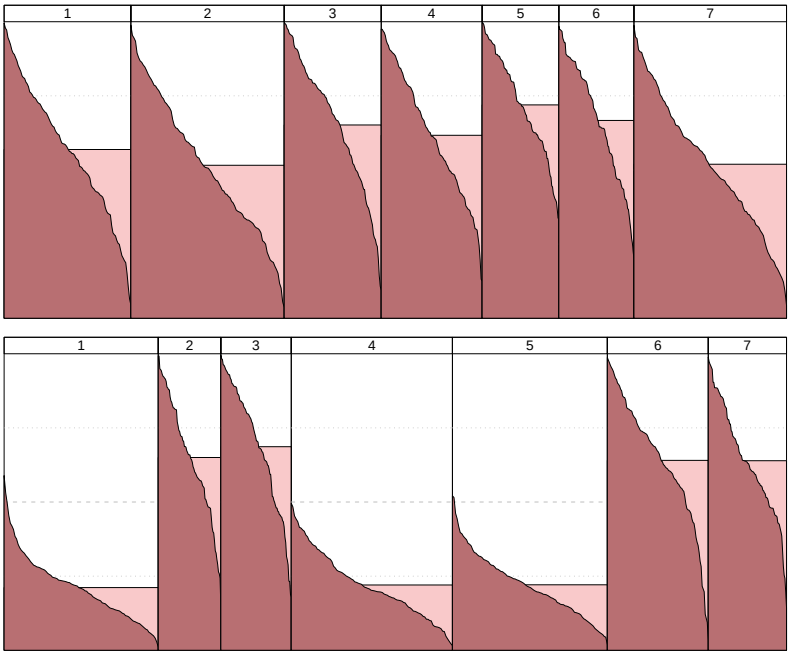

Figure 3: Shadow plots for the Gaussian data with poor (top) and good (bottom) separation.

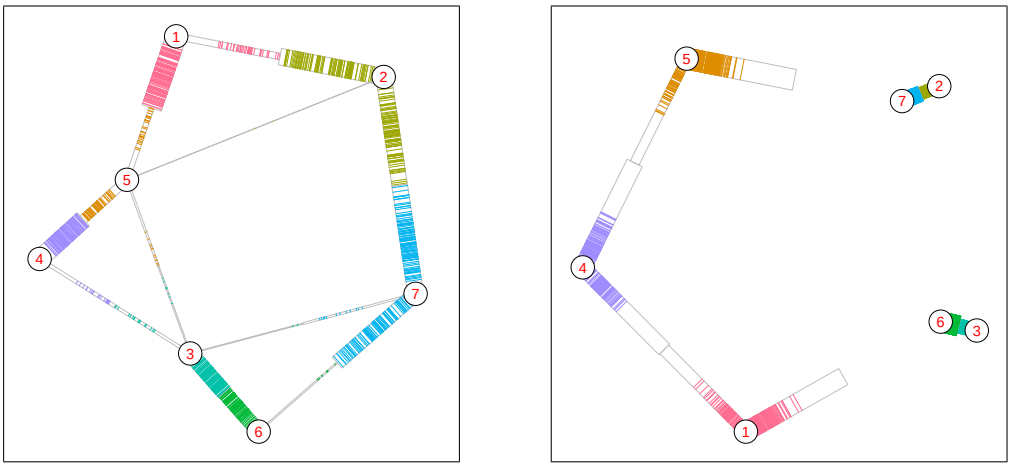

Figure 4: Shadow-stars for the Gaussian data with poor (left) and good (right) separation.

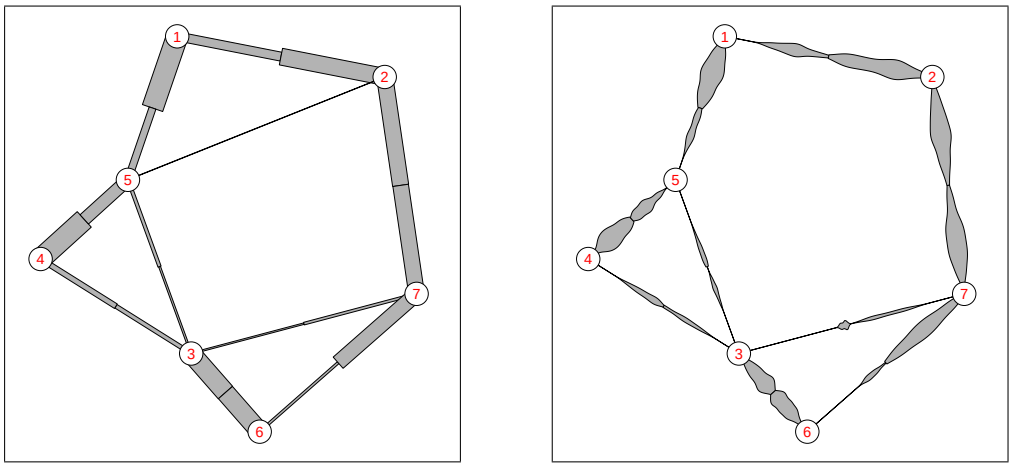

Figure 5: Shadow skeleton (left) and shadow violins (right) for the Gaussian data with poor separation.

that the definition of the "inner 50%" of data becomes feasible.

For data partitioned using a centroid-based cluster algorithm there is a natural total ordering for each point in a cluster (Leisch, 2008): The distance $d(\mathbf{x}, c(\mathbf{x}))$ of the point to its respective cluster centroid. Let

$$m_k = \text{median}\{d(\mathbf{x}_n, \mathbf{c}_k) | \mathbf{x}_n \in A_k\}$$

be the median distance of all points in cluster k to $\mathbf{c}_k$. We visualize the *inner area* a cluster occupies by the convex hull of all data points where $d(\mathbf{x}_n, \mathbf{c}_k) \leq m_k$, this corresponds to the box in a boxplot. After some experimentation we chose to define the *outer area* of a cluster as the convex hull of all data points that are no more than $2.5m_k$ away from $\mathbf{c}_k$, this corresponds to the whiskers in a boxplot. Points outside this area are considered as outliers.

Figure 1 compares the convex hulls of the clusters (bottom row) with 95% confidence ellipses (top row). As the data are a mixture of bivariate Gaussians, confidence ellipses are a "valid" visualization of the clusters, but only if the true number of clusters is known and found. As we have deliberately used a wrong number of clusters, several clusters have a non-elliptical shape, especially those splitting an underlying true cluster into two. To make the inner area more visible we use diagonal shading lines. As a new feature we use an angle of $k\pi/K$ for the lines of cluster k, such that overlapping clusters can be distinguished more easily, especially when no colors are available. Obviously, this could be improved upon by selecting orthogonal directions for clusters which are close to each other.

7 Example 1: Automobile Customer Survey

As first example demonstrating the proposed visualization techniques we use data from an automobile customer survey from 1983. A German car manufacturer sent a questionnaire to 2000 customers who had bought a new car approximately 3 months earlier, 793 of which returned a form without missing values. The full data set with 46 variables can be downloaded from the Statistics Department of the University of Munich at `http://www.statistik.lmu.de/service/datenarchiv/`. In the following we consider 21 binary variables on the characteristics of the vehicle and manufacturer: clearness, efficiency, driving properties, service, interior, quality, technology, model stability, comfort, reliability, handling, reputation of manufacturer, concept, character, power, resell value, styling, safety, sport, fuel consumption, and space. Each customer was asked to mark the most important characteristics that led him to buy one of the companies cars.

Figure 6 shows biplots of a principal component (PC) analysis of the data. The left panel gives a scatterplot of all data points projected on PC1 and PC2. There are 2 main directions in the plot, pointing

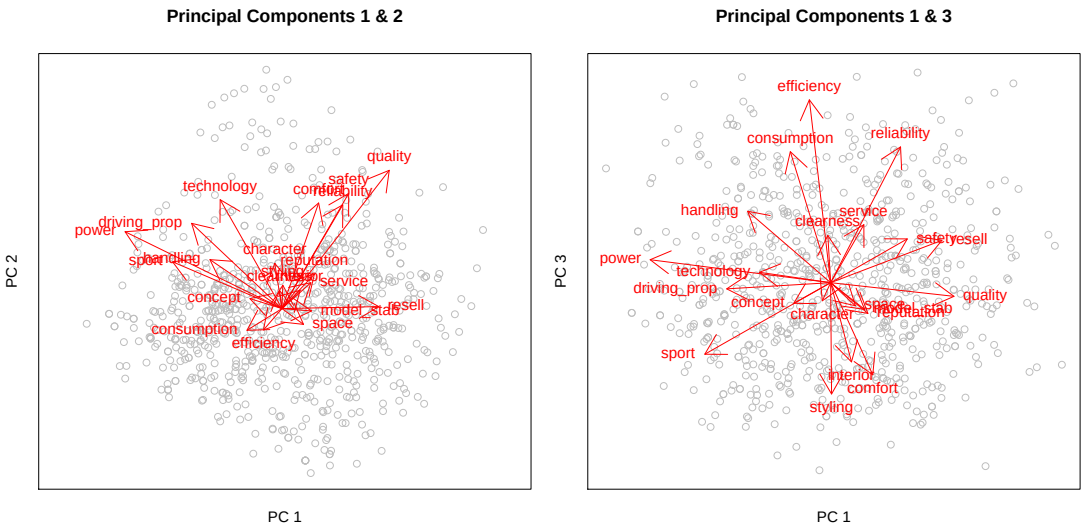

Figure 6: Principal component biplots of the automobile data.

up & right are quality-oriented variables like comfort/safety/reliability/etc, pointing up & left are the power-oriented variables technology/sport/handling/etc. Note that a lot of points are in the lower half of the plot, which basically is the negative direction for almost all variables. This is due to the fact that each consumer was asked to choose the most important characteristics, hence there are much more zeros than ones in the data set. The right panel of the plot shows PC1 and PC3: the x-Axis with PC1 is still quality (right) versus power (left), the y-axis with PC3 contrasts efficiency (top) versus styling/comfort (bottom).

Both scatterplots do not indicate the presence of any "natural clusters", any partition will probably divide the data into "arbitrary clusters" (Kruskal, 1977). Nevertheless it makes a lot of sense from a marketing researcher's point of view to cluster the data in order to partition the market into smaller subgroups of consumers which subsequently can be addressed by marketing actions tailored for the respective market segment. E.g., Mazanec et al. (1997) do not assume the existence of natural segments claiming that distinct segments rarely exist in empirical data sets and redefining market segmentation to be a construction task rather than a search mission for natural phenomena. Of course it is still of interest whether any of the market segments is markedly different from the rest and what their relations in 21-dimensional space are.

We present the results of a 5-cluster solution using the "neural gas" algorithm by Martinetz et al. (1993), because the resulting partition was easiest to interpret after trying several partitioning cluster algorithms with varying number of clusters. As the focus of this paper is on the introduction of new cluster visualization techniques, the exact choice of algorithm and number of clusters is not really important.

Neural gas is similar to k-means, but updates in each iteration not only the closest centroid, but also the second-closest: It repeatedly chooses a random observation, moves the closest centroid towards the observation, moves the second-closest centroid a little bit less then the closest etc. How many centroids are moved depends on hyperparamters of the algorithm, see the original publication for details.

Figure 7 shows the neighborhood graph corresponding to the 5 cluster neural gas solution projected into the spaces spanned by PC1 & PC2, and PC1 & PC3, respectively. Colors and plotting symbols for the data points are different for each cluster, yet no clear structure can be seen for most parts of the plots. The only structure that can easily be seen is that cluster 2 is "on top" of the left panel. The remaining four clusters all have regions with higher density, but this is not easily seen and the ellipses have large overlapping regions. An obvious (yet overhasty) conclusion would be that the principal component projection obscures the partition

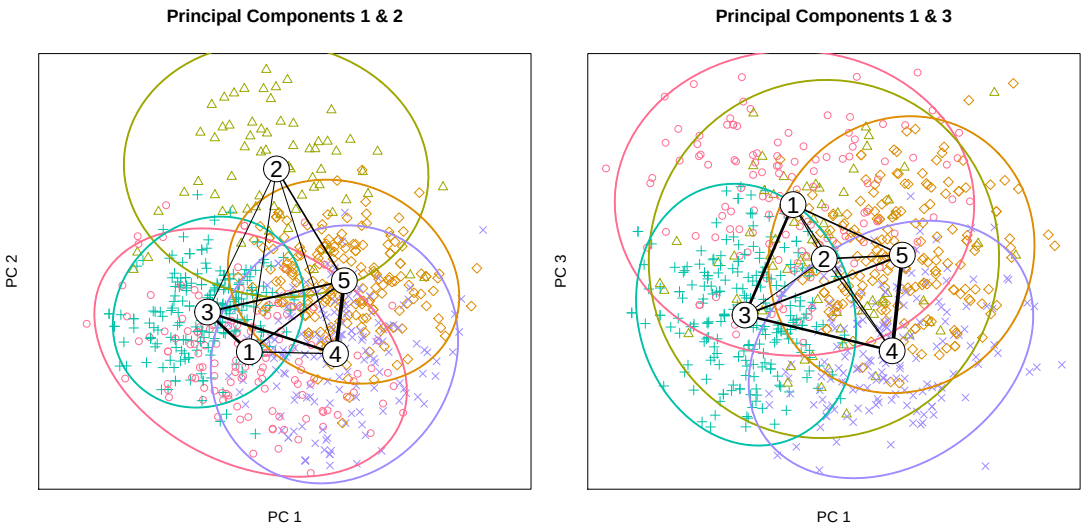

Figure 7: Neighborhood graph for the automobile data with 95% confidence ellipses for the clusters.

and a better projection is necessary to see the data structure (if possible at all).

Figure 8 shows the same neighborhood graph with convex hulls of the clusters and no point symbols. The shaded areas correspond to the convex hulls of the inner 50% of each cluster, the dashed lines to the convex hull of all points within 2.5 median distance from the centroid. Again, the left panel basically differentiates between cluster 2 and the rest. In the right panel the convex hull of cluster 2 has been omitted because it is in the middle and overlaps with all other clusters. It can clearly be seen that the remaining four clusters divide the space spanned by PC1 and PC3 into 4 regions approximately corresponding to high/low values on x- and y-axis. There is overlap due to projection, but there are also large "pure" regions. Together with the projected axes of the original variables from Figure 6 one could now proceed to construct a perceptual map for marketing purposes.

Of course, the overlap of the ellipses in Figure 7 can be reduced by only using 50% confidence regions, and not plotting point symbols would also make for a "clearer picture". However, Figure 8 shows that the confidence regions of the clusters are not elliptical and that especially the centroids are nowhere near the "middle" of their clusters. So smaller ellipses would have less overlap, but they would simply shade the wrong region in the plot.

Figures 9 and 10 show stripes and shadow plots for the automobile data, respectively. The interpretation is the same in both cases, all points are "far away" from their cluster centroids, none of the clusters is well-separated from the rest. All shadow values are large, the points have similar distances to the closest and second-closest centroid. The shadow star in Figure 11 can be used to identify which segments are more similar to each other than others. All together confirm once more that our grouping is artificial, we have constructed market segments rather than found naturally existing ones.

Note that how the high dimensional data are projected into 2 dimensions is not a focus of this work. We used principal component analysis in this example, but other projection methods like the asymmetric projections by Hennig (2004) would work equally well.

8 Example 2: German Elections

As second example we use the German parliamentary elections of September 18, 2005. The data consist of the proportions of "second votes" obtained by the five parties that got elected to the Bundestag (the first chamber of the German parliament) for each of the 299 electoral districts. The data set is directly available in R package `flexclust`. The "second votes" are actually more important than the "first votes"

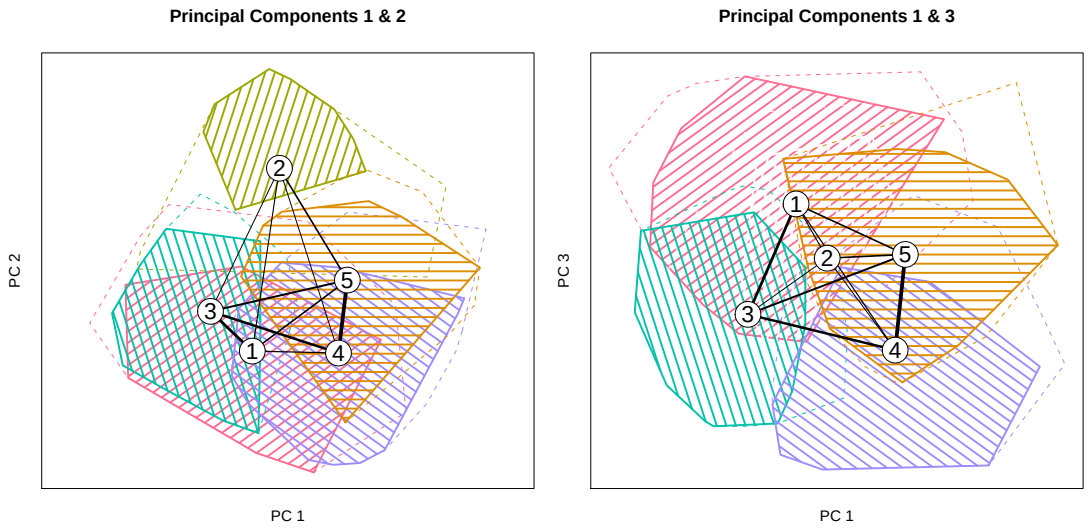

Figure 8: Neighborhood graph for the automobile data with convex hulls for the clusters: data points are omitted, the inner convex hull is shaded.

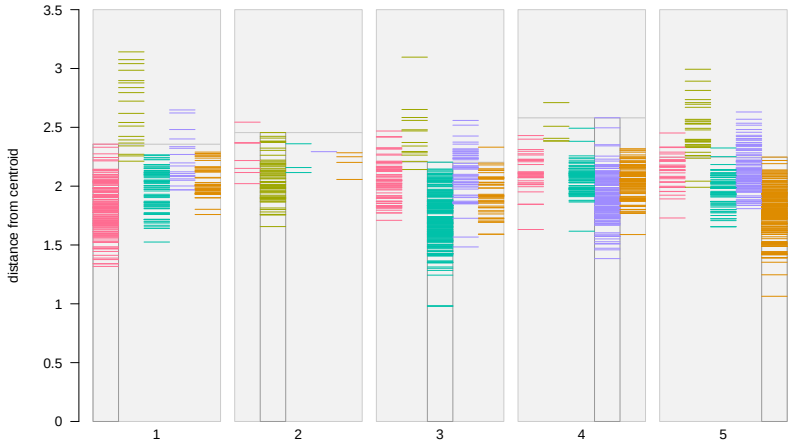

Figure 9: Stripes plot for the automobile data, only distances to closest and second-closest centroid are shown.

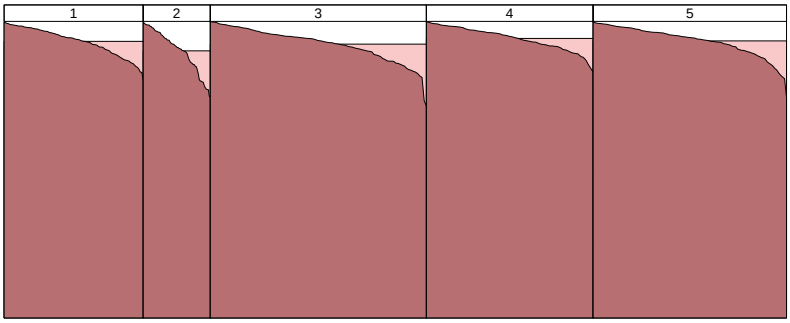

Figure 10: Shadow plot for the automobile data.

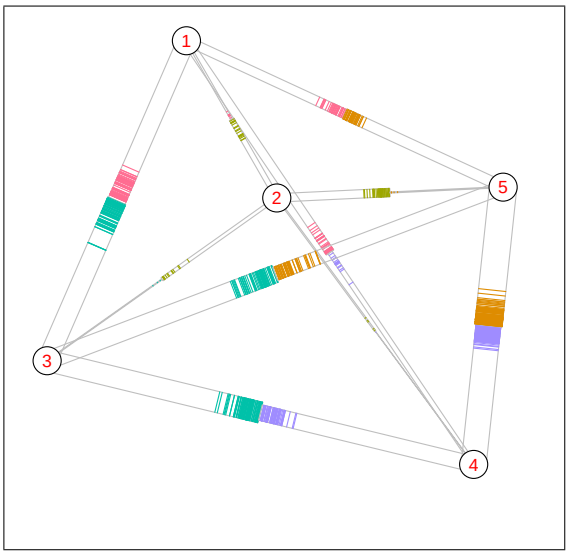

Figure 11: Shadow-stars for the automobile data.

because they control the number of seats each party has in parliament. Before election day, the German government comprised a coalition of Social Democrats (SPD) and the Green Party (GRUENE); their main opposition consisted of the conservative party (Christian Democrats, UNION) and the Liberal Party (FDP). The latter two intended to form a coalition after the election if they gained a joint majority, so the two major "sides" during the campaign were SPD+GRUENE versus UNION+FDP. In addition, a new "party of the left" (LINKE) canvassed for the first time; this new party contained the descendents of the Communist Party of the former East Germany and some left-wing separatists from the SPD in the former West Germany. A projection of the data onto the first two principal components is shown in the left plot of Figure 12. The point cloud in the lower left corner mainly correspond to districts in eastern Germany, where support for LINKE was strong, while the upper diagonal cloud corresponds mainly to districts in western Germany and contrasts the support for the two major parties: SPD (up) versus UNION (down). The final outcome of the election was UNION (226 seats in parliament), SPD (220), FDP (61), LINKE (54), and GRUENE (51). UNION and SPD formed a "large coalition" ("große Koalition"), because none of the above mentioned preferences had a majority.

The right panel of Figure 12 shows a four cluster solution from the k-means algorithm. Cluster 1 captures eastern Germany, while clusters 2–4 break western Germany in three parts. The principal component projection shows the structure of the data rather accurate, which makes it easier to relate the new plots introduced in this paper to the real structure of the data. The stripes plot in Figure 13 nicely shows the relations between the clusters: cluster 1 is far away from the rest, while cluster 3 is located "between" clusters 2 and 4.

Figure 14 shows another variant of the stripes plot. Here we are not interested in the relative locations of the clusters with respect to each other, but in a categorical background variable. In this case we have highlighted all 45 electoral districts from Bavaria. These are mainly in cluster 2, and approx. one quarter is in cluster 3. We also see that the Bavarian districts in cluster 3 are not close to the centroid, but have medium distance to it. Finally, Figure 15 shows shadow violins. Again it can clearly be seen that cluster 1 is well separated from the rest, while clusters 2–4 form a continuum. It also shows that cluster 1 is closer to 3 and 4 than to cluster 2.

Of course, most of the information contained in Figures 13–15 can also be seen in the linear projection in Figure 12 (which is the reason we have chosen this particular example in the first place). However, the stripes plot is independent from the dimensionality of the input space and works also when simple projections of the data like PCA fail.

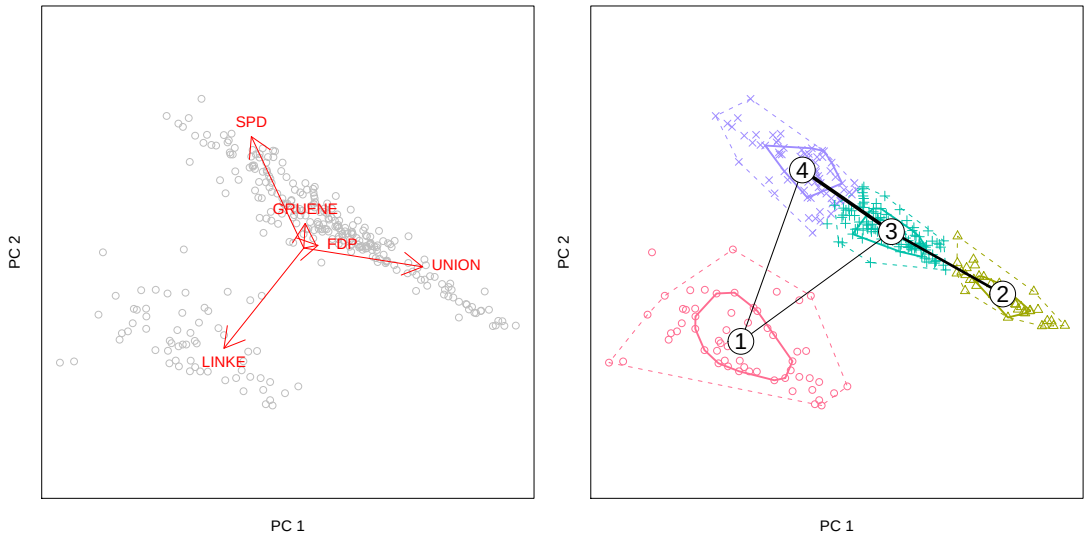

Figure 12: Principal component biplot of the German election data (left), and a four cluster solution (right).

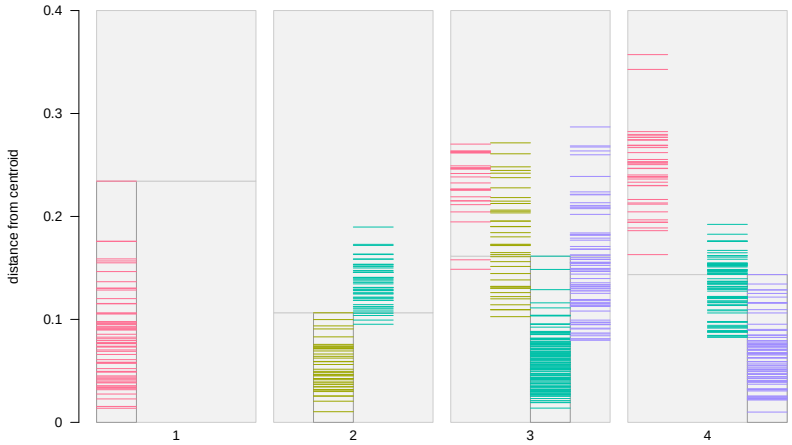

Figure 13: Stripes plot for the German election data, only distances to closest and second-closest centroid are shown.

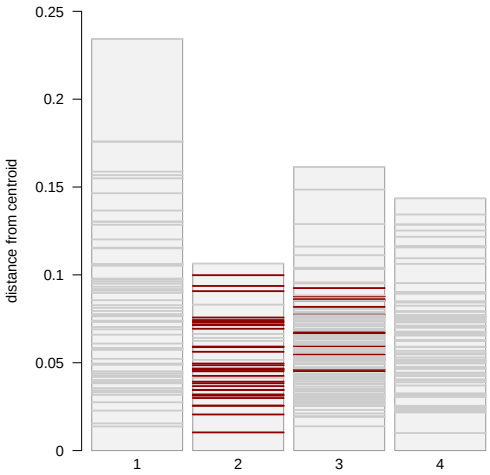

Figure 14: Stripes plot for the German election data, only distances to closest centroid are shown, districts from Bavaria are highlighted.

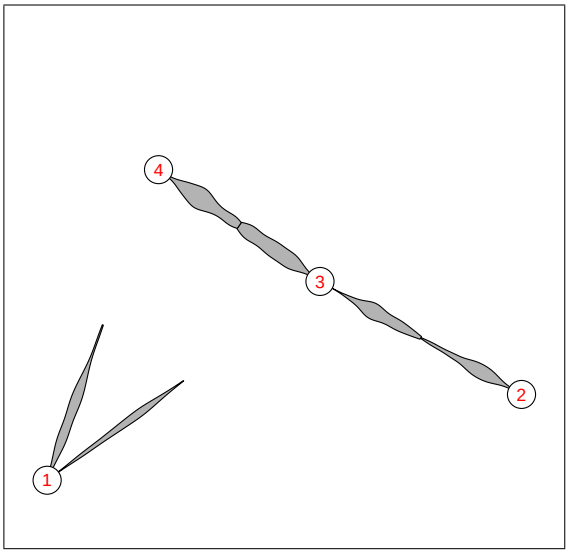

Figure 15: Shadow violins for the German election data.

9 Conclusions

We have extended the CLUSPLOT display by Pison et al. (1999) in two directions: Instead of connecting each cluster centroid with all the others, we connect only neighboring segments and obtain a graph that is more informative about the relative position of the clusters before projecting them into two dimensions. In addition the line width of the edges are proportional to the number of points that are in between the two clusters, such that thick lines connect clusters that are poorly separated from each other.

For non-elliptical clusters the convex hulls of inner and outer data points can be used as a 2-dimensional equivalent of a boxplot. Especially for larger data sets and partitions that cannot be easily projected into 2-d, plots can be easier to read if the original data points are only included in lighter colors or completely omitted. While this complexity reduction step is routine when comparing several samples using boxplots, it is much less common for 2-dimensional visualizations, because one has to impose an ordering onto $\mathbb{R}^2$. For clustered data a natural ordering exists through the distance of each point from its cluster centroid.

Shadow values can be used as a computationally more efficient approximation to silhouette values. Because shadow values are anchored at the cluster centroids, they allow for the definition of a completely new cluster visualization called shadow-stars. Compared with traditional silhouette plots they give not only information on how well a cluster is separated from the others, but also to which clusters it is close, if any. A natural question when the silhouette of a cluster indicates poor separation is to ask which other clusters are close; shadow-stars can help to efficiently encode this information graphically. A next step will be to investigate how these graphical methods can be used to compare different clusterings of the same data set with each other. Do different cluster algorithms and/or distance measures produce solutions with more or less overlap?

References

Becker, R., Cleveland, W., and Shyu, M.-J. (1996), "The visual design and control of trellis display," *Journal of Computational and Graphical Statistics*, 5, 123–155.

Everitt, B. S., Landau, S., and Leese, M. (2001), *Cluster Analysis*, London, UK: Arnold, 4 edition.

Gordon, A. D. (1999), *Classification*, Boca Raton, USA: Chapman & Hall / CRC, 2 edition.

Hartigan, J. A. and Wong, M. A. (1979), "Algorithm AS136: A k-means clustering algorithm," *Applied Statistics*, 28, 100–108.

Hennig, C. (2004), "Asymmetric linear dimension reduction for classification," *Journal of Computational and Graphical Statistics*, 13, 1–17.

Hintze, J. L. and Nelson, R. D. (1998), "Violin plots: A box plot-density trace synergism," *The American Statistician*, 52, 181–184.

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

Kohonen, T. (1989), *Self-organization and Associative Memory*, New York, USA: Springer Verlag, 3 edition.

Kruskal, J. (1977), "The relationship between multidimensional scaling and clustering," in *Classification and Clustering*, ed. J. V. Ryzin, Academic Press, Inc., New York, pp. 17–44.

Leisch, F. (2006), "A toolbox for k-centroids cluster analysis," *Computational Statistics and Data Analysis*, 51, 526–544.

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

MacQueen, J. (1967), "Some methods for classification and analysis of multivariate observations." in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, eds. L. M. L. Cam and J. Neyman, University of California Press, Berkeley, CA, USA, pp. 281–297.

Martinetz, T. and Schulten, K. (1994), "Topology representing networks," *Neural Networks*, 7, 507–522.

Martinetz, T. M., Berkovich, S. G., and Schulten, K. J. (1993), ""Neural-Gas" network for vector quantization and its application to time-series prediction," *IEEE Transactions on Neural Networks*, 4, 558–569.

Mazanec, J., Grabler, K., and Maier, G. (1997), *International City Tourism: Analysis and Strategy*, Pinter/Cassel.

Pison, G., Struyf, A., and Rousseeuw, P. J. (1999), "Displaying a clustering with CLUSPLOT," *Computational Statistics and Data Analysis*, 30, 381–392.

R Development Core Team (2008), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, URL `http://www.R-project.org`, ISBN 3-900051-07-0.

Rousseeuw, P. J. (1987), "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, 20, 53–65.

Rousseeuw, P. J., Ruts, I., and Tukey, J. W. (1999), "The bagplot: A bivariate boxplot," *The American Statistician*, 53, 382–387.