Computational Statistics (2004) 19:95–111

© Physica-Verlag 2004

Visualizing non-hierarchical and hierarchical cluster analyses with clustergrams

Matthias Schonlau

RAND, 1700 Main Street, Santa Monica, CA 90407, USA

Summary

In hierarchical cluster analysis dendrogram graphs are used to visualize how clusters are formed. Because each observation is displayed dendrograms are impractical when the data set is large. For non-hierarchical cluster algorithms (e.g. Kmeans) a graph like the dendrogram does not exist. This paper discusses a graph named "clustergram" to examine how cluster members are assigned to clusters as the number of clusters increases. The clustergram can also give insight into algorithms. For example, it can easily be seen that the "single linkage" algorithm tends to form clusters that consist of just one observation. It is also useful in distinguishing between random and deterministic implementations of the Kmeans algorithm. A data set related to asbestos claims and the Thailand Landmine Data are used throughout to illustrate the clustergram.

Key Words: Dendrogram, tree, cluster analysis, non-hierarchical, large data, Kmeans algorithm

1 Introduction

The purpose of cluster analysis is to cluster observations into groups. Unlike in supervised learning tasks (e.g. discriminant analysis) training data with correctly classified observations from which one may learn the group assignments are not available. Even the total number of groups is usually not known. Nonetheless, some algorithms like Kmeans require that the number of groups be pre-specified. In such cases the data analyst often runs the algorithm repeatedly with different number of groups. The decision of how many clusters to form is usually based on a combination of subject matter expertise, the definition of distance, which determines both that measures criterion measuring within cluster heterogeneity, and –in the case of hierarchical cluster analysis- the visual insight gained from a dendrogram (also called "cluster tree").

A dendrogram (e.g. Everitt and Dunn (1991), Hartigan (1975), Johnson and Wichern (1988)) is a tree graph that can be used to examine in hierarchical cluster analysis how clusters are merged. Figure 1 gives an example of a dendrogram with 100 observations. There are 100 leaves each representing one observation. The leaves are spaced evenly along the horizontal axis. The vertical axis gives the distance (or dis-similarity measure) at which any two clusters are joined. For example, in Figure 1 the last two clusters have a distance of just over 100 from one another. If one specifies a maximal distance beyond which two clusters must remain separate then one can compute the number of clusters from the maximal distance. For example, specifying a maximal distance of 40 in Figure 1 implies the formation of five clusters because y=40 intersects the tree five times.

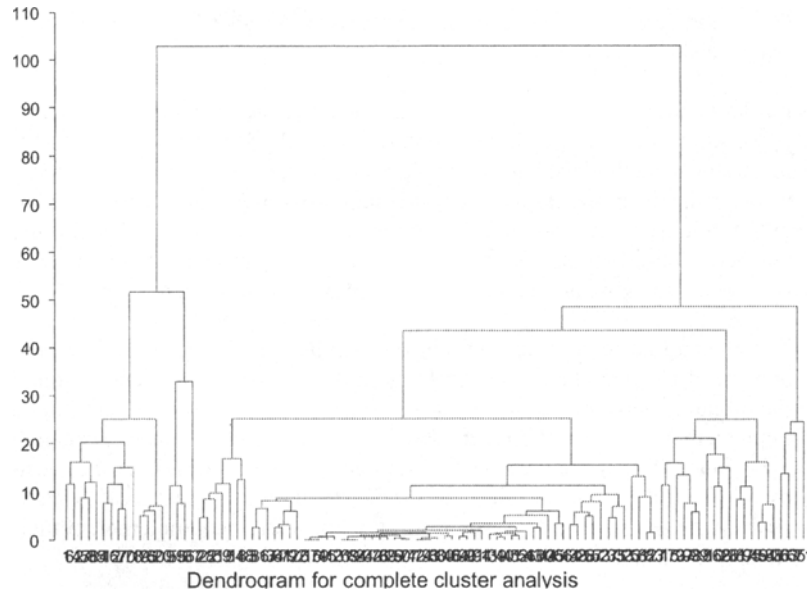

Figure 1: A dendrogram for 100 observations

Dendrograms are impractical when the data set is large because each observation must be displayed as a leaf they can only be used for a small number of observations. Stata, for example, allows up to 100 observations. As Figure 1 shows, even with 100 observations it is difficult to distinguish individual leaves. Also, they are not useful for non-hierarchical cluster analyses. This is true because the vertical axis represents the criterion at which any 2 clusters can be joined. Successive joining of clusters implies a hierarchical structure, meaning that dendrograms are only suitable for hierarchical cluster analysis.

Rousseeuw (1987) has proposed alternative tools to visualize cluster analysis: clusplots and silhouette plots. A clusplot is a bi-variate plot where each record is represented by one point. Either principal components or multi-dimensional scaling is used to project the data onto two dimensions. An ellipse is drawn around each cluster.

The goal of a silhouette plot is to visualize the degree of certainty with which observations are classified into clusters. This is measured by the difference of an observation's average dissimilarity to other members of its cluster and the observation's average dissimilarity to all observation to the next best cluster. These differences are standardized between –1 and 1 and a bar chart of differences is plotted for each cluster. Both of Rousseeuw's plots are implemented in Splus.

Unlike the dendrogram both of these plots refer to a specific choice of the number of clusters K.

For large numbers of observations hierarchical cluster algorithms can be too time consuming. The computational complexity of the three popular agglomerative hierarchical methods (single, complete and average linkage) is of order $O(n^2)$, whereas the most popular non-hierarchical cluster algorithm, Kmeans (MacQueen 1967), is only of the order O(Kn) where K is the number of clusters and n the number of observations (Hand et al., 2001). Therefore Kmeans, a non-hierarchical method, is emerging as a popular choice in the data mining community.

The clustergram examines how cluster members are assigned to clusters as the number of clusters changes. In this regard it is similar to the dendrogram. Unlike the dendrogram this graph can be used for non-hierarchical data also. I implemented the clustergram in Stata[1] and discuss the Stata implementation in Schonlau (2002). This paper introduces a more general version of the clustergram by allowing different meanings of the clustergram's vertical axis. It also shows how clustergrams can give some insight into different implementations of the Kmeans algorithm.

The outline of the remainder of this paper is as follows: Section 2 introduces the clustergram. Section 3 uses the clustergram for different implementations of the Kmeans algorithm. Section 4 introduces a different plotting function of the clustergram and gives examples of how to examine the effect of individual variables on the cluster assignments. The Thailand Landmine Data set is used throughout this section. Section 5 concludes with some discussion.

2 The clustergram

A large number of lawsuits concerning asbestos related personal injuries have been filed in the U.S.. One interesting question is: can companies be clustered into groups on the basis of how many law suits were filed against them? The data consist of the number of asbestos suits filed against 178 companies in the U.S. from 1970 through 2000. I separate the number of asbestos suits by year to create 31 variables for the cluster algorithm. Each variable consists of the log10 number of suits that were filed against a company in a year. Since the variables refer to the same scale they are not standardized.

---

[1]The Stata ado files can be obtained from www.schonlau.net/clustergram.html or by emailing Matthias_Schonlau@rand.org .

In preparation for constructing the clustergram, one needs to run the chosen cluster algorithm multiple times; each time specifying a different number of clusters (e.g. 1 through 10). The clustergram is constructed as follows: For each cluster within each cluster analysis, compute the mean over all cluster variables and over all observations in that cluster. For example, for x=2 clusters compute two cluster means. For each cluster plot the cluster mean versus the number of clusters. Connect cluster means of consecutive cluster analyses with parallelograms. The width of each parallelogram is proportional to the number of observations that it contains. By "width" I mean the difference of the y-values.

Figure 2 illustrates this. Initially (Number of Clusters=1), all observations form a single cluster. The y-value for this cluster corresponds to the grand mean over all observations and all x-variables. This cluster is split into two clusters. Since the lower parallelogram is much thicker than the upper one, there are many more observations that fall into the lower cluster. These two clusters are then split into three clusters. A new cluster is formed in the middle that draws some observations that were previously classified in the lower cluster, and some that were previously classified in the higher cluster. Because the new cluster is formed from observations of more than one previous clusters (i.e. has more than one parent) this is a non-hierarchical split. The vertical axis refers to the log10 number of average number of law suits filed against a company. Therefore "higher" or "lower" clusters refer to clusters with companies that on average have a larger or smaller number of lawsuits.

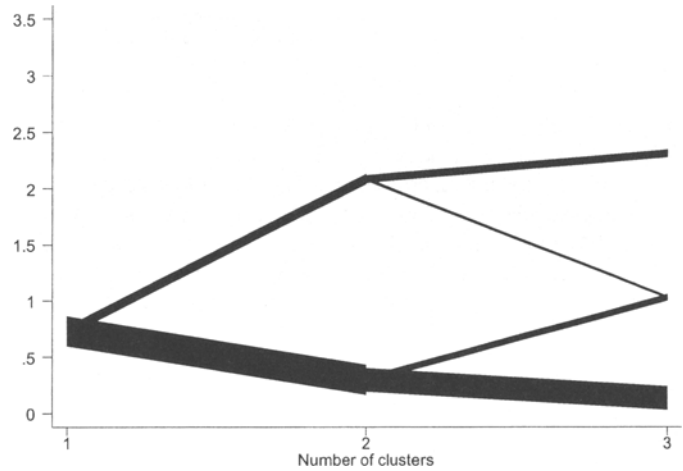

Figure 2: A clustergram for 1–3 clusters for the asbestos data. The cluster assignments stem from the Kmeans algorithm.

To avoid visual clutter the proportionality constant that determines the width of all parallelograms or graph segments can be controlled. The constant should be chosen large enough that clusters of various sizes can be distinguished and small enough that there is not too much visual clutter.

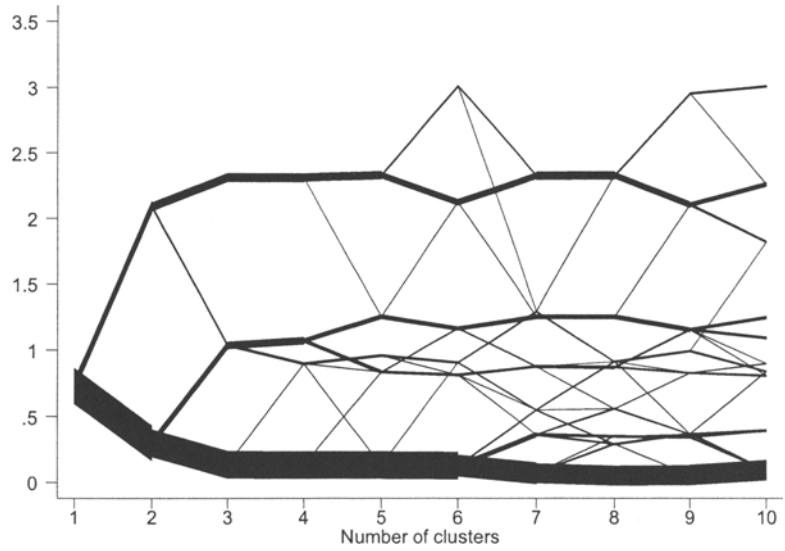

Figure 3: Clustergram with up to 10 clusters for the asbestos data. The Kmeans cluster algorithm was used.

Figure 3 extends Figure 2 to up to 10 clusters. We see that the companies initially split into two clusters of unequal size. (In fact this is the same split as in Figure 2). The cluster with the lowest mean remains the largest cluster by far for all cluster sizes. One can also identify hierarchical splits. The split from 3 to 4 clusters is almost hierarchical (it is not strictly hierarchical because a single company joins from the bottom cluster). Also, there are a number of individual companies that appear to be hard to classify because they switch clusters.

For 6, 9 and 10 clusters a cluster at the top of the graph emerges that does not emerge for 7 or 8 clusters. This highlights a weakness of the Kmeans algorithm. For some starting values the algorithm may not find the best solution. The clustergram in this case is able to identify the instability for this data set.

Figure 4 shows a clustergram for a hierarchical, average linkage cluster analysis.

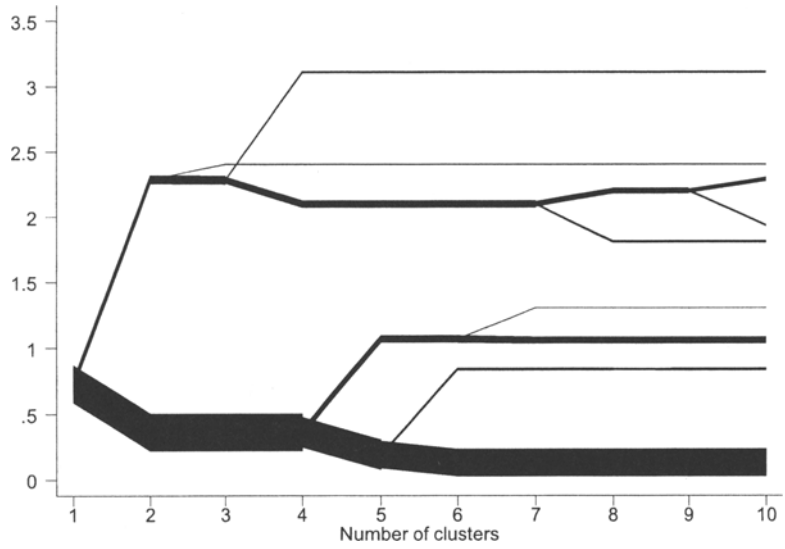

Figure 4: A clustergram for the "average linkage" (hierarchical) cluster analysis for the asbestos data.

Because of the hierarchical nature of the algorithm, once a cluster is split off it cannot join with other clusters later on. Qualitatively, Figure 3 and Figure 4 convey the same picture. Again, the bottom cluster has by far the most members, and the other two or three major streams of clusters appear at roughly the same time with a very similar mean.

Figure 5 shows a clustergram for a hierarchical, single linkage cluster analysis. All clusters are formed by splitting a single company off the largest cluster. If our goal is to identify several non-trivial clusters, this cluster algorithm does not suit this purpose. Figure 5 conveys this information instantly. While it is possible to split off more than single observations when new clusters are formed, this is a fairly typical clustergram for single linkage cluster analysis.

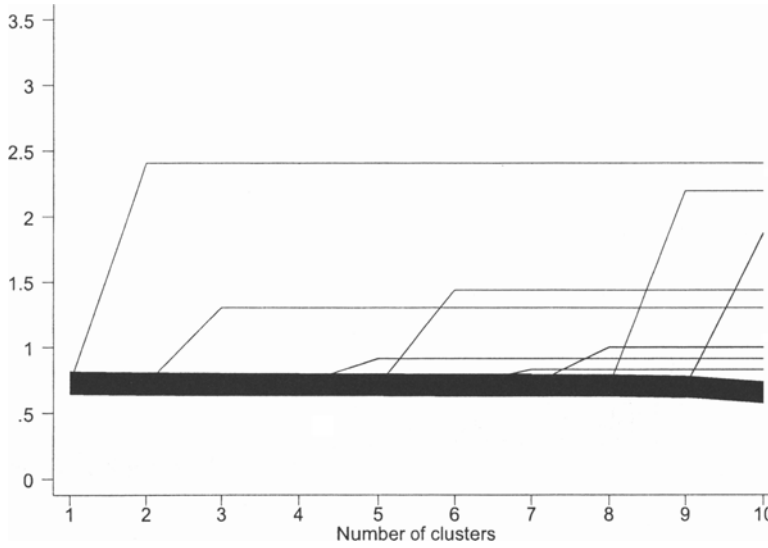

Figure 5: A clustergram for the "single linkage" (hierarchical) cluster analysis for the asbestos data.

Of course, the ultimate decision of the number of clusters is always somewhat arbitrary and should be based on subject matter expertise, the criterion that measures within-cluster homogeneity as well as insight gained from the clustergrams. It is re-assuring that Kmeans and the algorithm "average linkage" lead to qualitatively similar results.

3 Using the Clustergram to assess implementations of the Kmeans algorithm

Given a distance metric the Kmeans algorithm assigns records to one of K clusters. The Kmeans algorithm works as follows: K starting values are selected for the K cluster means. (1) Each record is assigned to a cluster according to the distance function. (2) The cluster means are updated. Steps (1) and (2) are repeated until the assignments in Step (1) do not change in two successive iterations. Unlike the deterministic hierarchical linkage algorithms Kmeans usually uses random starting values.

Figure 6 displays clustergrams based on different SAS (upper left and lower right), Stata (upper right) and Splus (lower left) implementation of the Kmeans

algorithm. Except for different random generators the two implementations in the right column (random SAS and Stata) are equivalent. Each clustergram reruns the Kmeans algorithm 20 times (corresponding to the labels on the horizontal axis)- each time using the random generator to generate new seeds. Also note that the clustergram in Figure 3 looks somewhat different than the Stata implementation on the upper right in Figure 6 because different random seeds were used.

The implementation in the left column look different from the ones in the right column because they appear to be more stable. By more stable I mean that there is generally less re-assignment of observations to different clusters. Noticeably, in the SAS implementation the two highest clusters are completely stable after k=6. The stability stems from the fact that the two implementations on the left of Figure 6 are deterministic, and the two algorithms on the right are not.

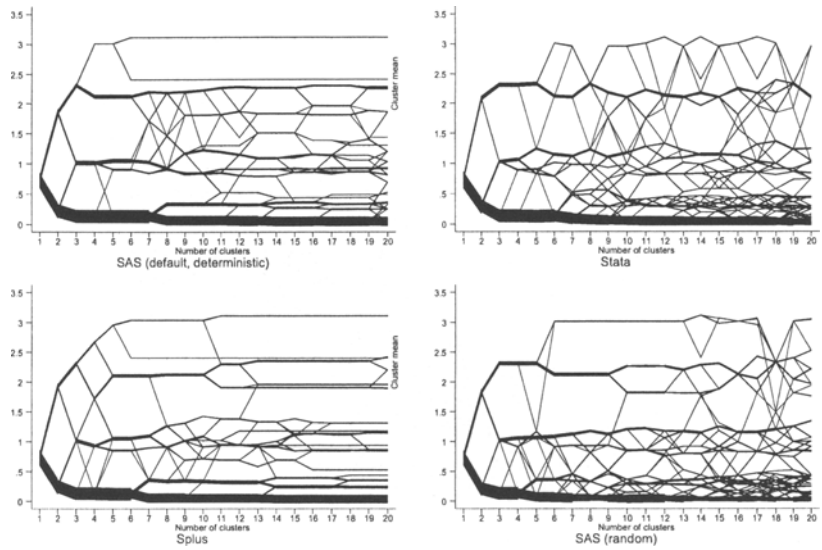

Figure 6: Clustergrams for various Kmeans implementations. Upper left: deterministic (default) SAS implementation; Upper Right: random (default) Stata implementation; Lower left: deterministic (default) Splus implementation; Lower Right: random SAS implementation

For the deterministic SAS implementation all observations are considered sequentially as starting values. Broadly speaking, observations replace current starting values if replacing increases the minimum distance among the starting

values. (For details see Proc Fastclus in the SAS Manual, Version 8). As a result the starting values tend to cover the space better than would be the case in a random implementation. Splus by default uses a hierarchical complete linkage cluster algorithm to determine the starting centroids for the Kmeans algorithm. The algorithm implemented follows Hartigan (1979). It is more complicated than the simple 2-step procedure described earlier. Because it uses the deterministic hierarchical method to obtain starting values, the algorithm is also deterministic.

The fact that some algorithms are most stable does not necessarily mean that they are better. What does better mean? For a fixed number of clusters the algorithm tries to minimize the distances between the data and the cluster means. The value achieved is often called the criterion. It turns out that the deterministic (=default) SAS implementation most of the time achieves a better criterion than the random Kmeans algorithm, in particular for larger numbers of clusters.

4 The Thailand Landmine Data

The Landmine Impact Survey in Thailand (Survey Action Center, 2002) investigated the affect of landmines in communities in Thailand. This survey is part of the Global Landmine Initiative; an initiative aimed to catalogue the global socio-economic impact of landmines. The survey in Thailand was conducted over a 14 months period ending in June 2001. Each of the 530 observations in this data set corresponds to one Thai community/village that was impacted by landmines in some way. The variables used in this section were chosen by the subject matter expert. They are all dummy variables that indicate whether certain community assets were blocked or adversely affected by landmines at the time of the survey. These community assets are (variable names in parentheses): rain fed farmland (rain), pastures (pasture), drinking water sources (drink), other water sources not suitable for drinking (other), land not used for agriculture such as forests (non-agriculture), housing (housing), village infrastructure except roads (infra).

In section 4.1, I inspect cluster assignments stemming from Kmeans and Average linkage analyses for the Thailand Landmine Data. The goal is merely to see how the cluster algorithm assigns the variables rather than justifying the choice of any particular cluster algorithm in view of the fact that the cluster variables are dummy variables. In section 4.2 I compare a dendrogram and a clustergram side by side. In section 4.3 I introduce a different plotting function for the y-axis and illustrate it with the Thailand Landmine Data.

4.1 Examining individual variables with clustergrams

When I was initially approached about this data I was given the cluster variables and the cluster assignments. It later turned out the cluster assignments stem from the Kmeans algorithm. The goal was to investigate how individual variables are distributed across the cluster assignments. Clustergrams with individual variables can be seen in Figure 7. Rather than plotting the average over all observations within a cluster for *all* variables, the average over all observations within a cluster for a single variable is plotted. If this variable is a 0/1 dummy variable, the vertical axis can be interpreted as the percentage of observations that have the value "1". Figure 7 displays four individual variables: nonagriculture (upper left), pasture (upper right), rain (lower left) and other water sources (lower right). The graph in the upper left panel shows that the first two clusters are formed by splitting on the variable nonagriculture. The third cluster is a cluster that contains only observations with pasture=1 (upper right panel). When the data are split into four clusters all but one are pure with respect to the variable "rain". A cluster is "pure" with respect to one variable if all observations have the same value for that variable. When five clusters are formed all but one cluster are pure for the variable "other." If these plots were interactively linked one could imagine highlighting a pure cluster for one variable to find out whether it is also pure for any of the other variables.

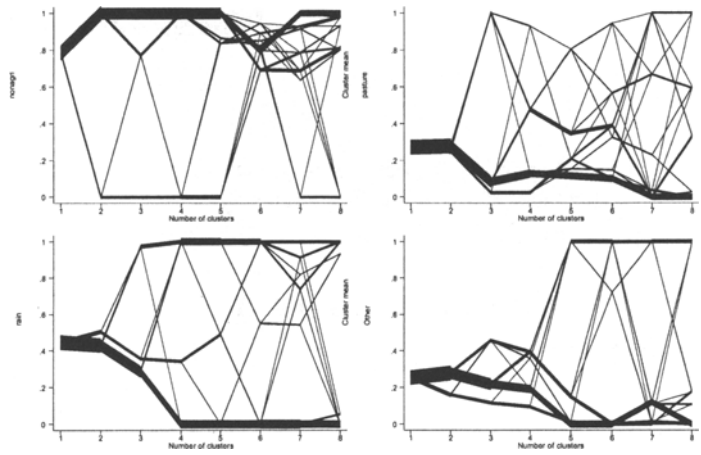

Figure 7: Thailand Landmine Data clustergrams with individual variables: nonagriculture (upper left), pasture (upper right), rain (lower left) and other water sources (lower right). The cluster assignments stem from the Kmeans algorithm.

Figure 8 shows individual clustergrams when the cluster assignments are based on the average linkage algorithm. One can see that a very different picture emerges: there is one large cluster. This large cluster is not pure with respect to any single variable, it might be labeled a "catch all" cluster. As the number of clusters increases small clusters are split of the large one. The small clusters tend to be pure in at least one variable.

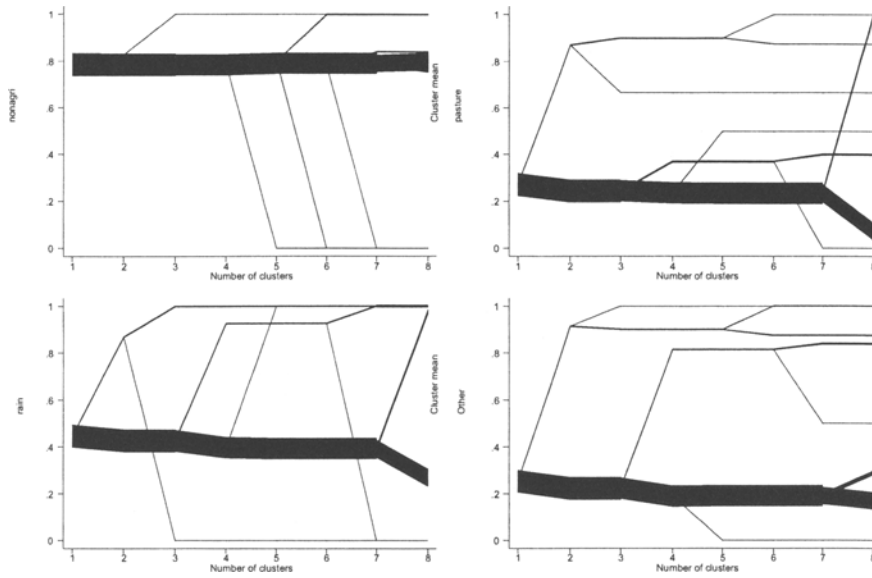

Figure 8: Thailand Landmine Data clustergrams where individual variables are plotted on the vertical axis. The cluster assignments stem from the average linkage algorithm.

4.2 Comparing a dendrogram with a clustergram

Figure 9 compares a clustergram (right panel) with a dendrogram (left panel) for average linkage clustering. It is not possible to show a full dendrogram because the number of observations is too large. Therefore I chose to display a dendrogram with up to 8 clusters; the same number of clusters that is displayed in the

clustergram. Because each cluster is assigned equal space in the dendrogram, it is visually not obvious that one cluster contains most of the observations. However, in the Stata implementation of the dendrogram the number of observations contained in each cluster is given near the bottom of the graph.

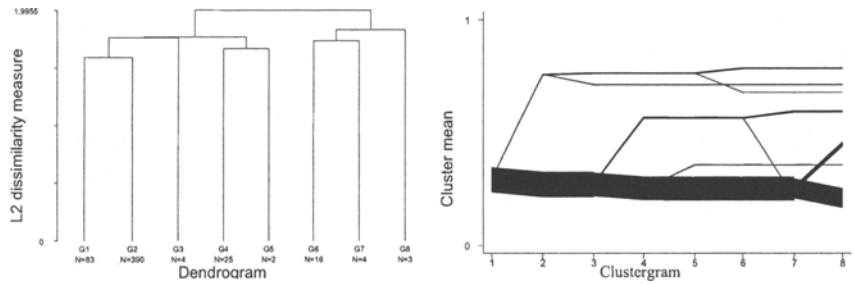

Figure 9: Comparison of a clustergram and a dendrogram for average linkage clustering for the Thailand Landmine Data. A full dendrogram cannot be displayed because the number of observations is too large.

4.3 A clustergram with plotting function proportional to cluster size

Earlier I introduced the clustergram where the vertical axis corresponded to the grand mean over all variables and all observations in a given cluster. Instead of the grand mean other functions can be chosen. In this section I discuss one such function.

To motivate the function consider the layout of dendrograms. As Figure 1 shows all leaves are plotted uniformly across the horizontal axis. Now suppose one were to form two clusters. It is easy to gauge with the eye that the left cluster in Figure 1 is much smaller than the right cluster. Since each observation occupies the same amount of space the width of the cluster gives a visual cue as to its size. The plotting function that I introduce in this section for the clustergram has the same effect. Because of the hierarchical nature of the algorithms - moving from k to k+1 clusters - the order of the k+1 clusters is completely determined by the order of the previous k clusters – except that left and right branch could be interchanged in the new split. Because clustergrams do not require a hierarchical structure the cluster order must be determined differently.

The new plotting function is as follows. As in the clustergram described earlier, the clusters remain in the order that is determined by their grand mean. However,

like in the dendrogram the clusters are spaced out such that the space allocated to each cluster is proportional to the cluster size. The width of the parallelogram also remains proportional to size as before.

Figure 10 gives both the mean and the proportional-to-size plotting function for the average linkage clustering on the Thailand Landmine Data. On the proportional-to-size plot it is easier to identify separate clusters. The left panel of Figure 10 displays a clustergram with the mean as the plotting function, the right panel uses the proportional-to-size plotting function. The first four clusters are clearly distinguishable from one another in the left panel of Figure 10. When the number of clusters equals five, one cannot clearly distinguish the two clusters that the arrow in the left panel points to because of over-plotting. The right panel makes a clear distinction between the two clusters (each has a separate arrow).

In the proportional-to-size clustergram over-plotting can only occur for low-resolution graphs when adjacent clusters all consist of singleton observations (or are very small). In the proportional-to-size clustergram the y-axis can also be used to gauge the proportion of observations in several adjacent clusters.

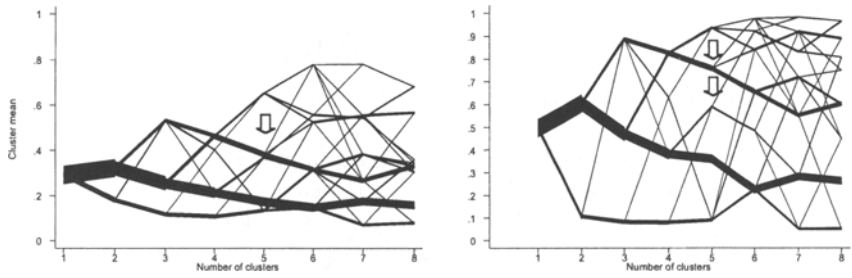

Figure 10: Thailand Landmine Data clustergrams with the two different plotting functions: average (left) and proportional-to-size (right). The cluster assignments stem from the Kmeans algorithm. The arrows indicate a split that is only visible in the proportional-to-size plot.

The question of which plotting function to use depends on the individual problem. When there is one dominant cluster as often occurs in single linkage clustering, the proportional-to-size plotting function wastes too much space. When the grand mean can be meaningfully interpreted, it makes sense to use it as the plotting function. When there is over-plotting due to similar grand means of distinct clusters, or when most of the line segments are concentrated in a small region of the plot (e.g. because of cluster means that are outliers), using the mean as the plotting function is not useful. Clearly, other functions like principal components or weighted means could be used also.

5 Discussion

The clustergram and the dendrogram differ in a number of ways. Most importantly, the dendrogram uses "distance" as the second axis. "Distance" determines the number of clusters. The clustergram gives the number of clusters directly. Distance conveys more information but for non-hierarchical cluster analysis an analogous axis cannot be constructed. Therefore I chose to plot only the number of clusters in the clustergram. If a hierarchical cluster analysis is conducted *and* the data set is small enough to display a full dendrogram this is usually preferable because "distance" conveys more information than "number of clusters."

Second, the layout on the dendrogram's horizontal axis is naturally determined by the tree (except for some freedom in whether to label a branch left or right). The layout of the non-hierarchical tree is not obvious. I chose to present the mean and the proportional-to-size plotting function here. There are functions of the data that could be used, for example the first principal component or the data could be scaled to a single dimension through multi-dimensional scaling. The choice of which function to use is largely a question of which quantity is more easily interpretable. Proportional-to-size may be especially useful when over-plotting obscures important details.

Third, in a clustergram the (non-hierarchical) tree is not extended until each leaf (cluster) contains only one observation. For dendrograms usually the full tree is shown, though it is possible to only show the upper portion of a tree as can be seen in Figure 9. Fourth, in the clustergram the width of the parallelogram indicate cluster size. This is not necessary for the dendrogram. Because all leaves are plotted uniformly across the horizontal axis the width of the cluster already gives a visual cue as to its size.

In the dendrogram the path of each observation through the tree can be observed. This is not true for the clustergram: the paths that individual observations take from parallelogram to parallelogram are not shown. A similar effect could be achieved by highlighting the paths that observations take through the clustergram: if the user (fully) highlights an individual parallelogram then all linked parallelogram are partially highlighted. A linked parallelogram is a parallelogram that shares observations with the original parallelogram. Partially highlighting refers to highlighting an area of the parallelogram proportional to the number of observations that this parallelogram shares with the original parallelogram. The effect is that one can observe how observations that from an individual cluster spread among different clusters as the number of clusters increases. This is only useful for non-hierarchical cluster analysis. For example, a parallelogram with just

one observation could be highlighted to see whether the same observation is often re-assigned as the number of clusters increases. For hierarchical cluster analysis the effect would be to highlight the entire sub-tree generated by the original parallelogram. It is possible to look at several seed parallelogram simultaneously by using different colors. For example, when the data form three clusters the use of three colors would enable us to trace all observations through the remainder of the clustergram.

6 Acknowledgement

I am grateful for support from the RAND statistics group. I am grateful for discussions with Brad Efron, members of the RAND statistics group, participants of the 2002 Augsburg (Germany) workshop on data visualization and for comments from two anonymous referees. I am grateful to Steve Carroll at RAND for involving me in the Asbestos project, which prompted this work. I am grateful to Aldo Benini who was part of the Landmine Impact Project and gave me access to the data.

References

Everitt, B.S., Dunn, G. (1991), *Applied Multivariate Data Analysis*, New York: John Wiley & Sons.

Hand, D., Mannila, H., Smyth P. (2001), *Principles of Data Mining*, Cambridge, MA: Massachusetts Institute of Technology.

Hartigan, J.A. (1975), *Clustering Algorithms*. New York: Wiley.

Hartigan, J.A.,Wong, M.A. (1979), A k-means clustering algorithm. *Applied Statistics*, 28, 100-108.

Johnson R.A., Wichern D.W. (1988), *Applied Multivariate Analysis*, 2nd ed, Englewood Cliffs, NJ: Prentice Hall.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. L.M. LeCam and J. Neyman (eds.) Berkeley: University of California Press, 281-297.

Rousseuw, PJ. (1987), Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65.

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

Survey Action Center (2002), *Landmine Impact Survey Executive Summary: Kingdom of Thailand.* Implemented by the Survey Action Center and Norwegian's Peoples Aid. Certified by the United Nations Certifications Committee. Downloadable from http://www.sac-na.org/resources_report_thailand.html (last accessed on April 29, 2003).