Exploring cluster analysis

Hadley Wickham, Heike Hofmann, Di Cook
Department of Statistics, Iowa State University
hadley@iastate.edu, hofmann@iastate.edu, dicook@iastate.edu

2006-03-18

Abstract

This paper presents a set of tools to explore the results of cluster analysis. We use R to cluster the data, and explore it with textual summaries and static graphics. Using Rggobi2 we have linked R to GGobi so that we can use the dynamic and interactive graphics capabilities of GGobi. We then use these tools to investigate clustering results from the three major families of clustering algorithms.

1 Introduction

Cluster analysis is a powerful exploratory technique for discovering groups of similar observations within a data set. It is used in a wide variety of disciplines, including marketing and bioinformatics, and its results are often used to inform further research. This paper presents a set of linked tools for investigating the clustering algorithm results using the statistical language R [14], and the interactive and dynamic graphics software GGobi [16] .

R and GGobi are linked with the R package RGGobi2 [1]. This package builds on much prior work connecting R and S with XGobi and GGobi [17, 18], and provides seamless transfer of data and metadata between the two applications. This allows us to take advantage of strengths of each application: the wide range of statistical techniques already programmed in R, including many clustering algorithms, and the rich set of interactive and dynamic graphics tools that GGobi provides. The R code used for the analyses and graphics in this paper has been built into an R package, `clusterExplorer`, which is available from the accompanying website http://had.co.nz/cluster-explorer, along with short videos illustrating dynamic techniques that are not amenable to static reproduction.

In this paper we provide a brief introduction to clustering algorithms and their output. We describe graphical tools, both static and dynamic/interactive, that we can use to explore these results. We then use these visual tools to explore the results of three clustering algorithms on three data sets.

2 Clustering algorithms

It is hard to define precisely what a cluster is, but obvious clusters are intuitively reasonable, as in figure 1. Here we would hope that any reasonable clustering algorithm would find the three obvious clusters. However, real data is rarely as clear cut and it is unusual to see such an apparent underlying structure. For this reason, we typically want cluster analysis to organise the observations into representative groups.

It is generally hard to tell if the generated clusters are good or bad. However, it is more important that the clusters are useful for the problem at hand. Typically there is no one true clustering that we are trying to recover, so it is common to use multiple clustering techniques each of which may construct different clusters and give us different insights into the problem. Once we have these multiple clusters we need to be able to compare between them, and also investigate how the clusters partition the original data space. We discuss useful techniques for these problems in the following section.

There is much literature dedicated to finding the "best" clustering, or reclaiming the "true" number of clusters. While these results are useful for homing in on good candidates, it is wise to exercise some caution as the assumptions may not hold true in practice. We recommend that you use them as a rough guideline for suggesting interesting clusterings to explore, and we encourage you to investigate multiple clustering algorithms.

Many of the clustering methods have difficulty with data that has high co-dimensionality, or multicollinearity. In general, results will be better if you can remove extraneous variables. However, you can not tell which variables are extraneous until you have clustered the data, which may have been affected by the extra variables. It is useful to take an iterative

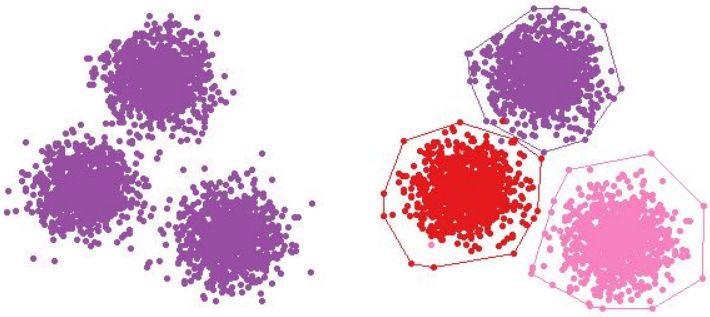

Figure 1: Sometimes cluster structure is obvious! The right plot overlays convex hulls on the clusters.

approach, removing a variable after it becomes clear that the variable contains little useful information.

Finally, it is worth remembering that the choice of distance metric will have as much or more effect on the final clusters as the choice of clustering algorithm. Some clustering methods work directly on the distance matrix allowing enormous flexibility in this choice. There are many distance measures, particularly for discrete data, and it is worth considering which is most applicable to your problem. For those clustering methods which implicitly work on Euclidean distances, transformation of the data can reproduce other distance measures. For example, scaling to common variance effectively changes the distance metric to correlation. This is recommended when you have variables measured on different scales.

3 Interactive investigation tools

The aim of this paper, and accompanying R package, is to provide tools to enable comparison of different cluster assignments. These can come from:

- Different clustering algorithms (eg. hierarchical, k-means, model based).
- Different algorithm parameters algorithms (eg. metric, distance calculation).
- Different numbers of clusters.
- Additional classification information not used during the clustering.

We also want to be able to explore what makes different clusters different, and how the clusters divide up the original data space. By using R and GGobi together we can provide a variety of methods to aid exploration and comparison:

- Textual summaries: confusion matrices, cluster means and other summary statistics.
- High quality static graphics generated by R: fluctuation diagrams [10], parallel coordinates plots, boxplots, barcharts and spineplots [11].
- Dynamic and interactive graphics in GGobi: animations cycling between different cluster assignments, tours to explore the clustering in high dimensions, manual tuning of clusters using brushing, animations using color to explore misclassified cases. Unfortunately these can not be illustrated on the static printed page, but videos can be found on the paper's website.

A particularly useful feature in GGobi is the grand tour [2, 5, 6]. The grand tour randomly rotates through all possible ways of projecting the original high dimensional data onto fewer dimensions. When plotting data we usually project it down onto two dimensions. One way of projecting the data is the scatterplot matrix, which looks at each face of the data cube. Another way is to use the grand tour and look at it from every angle. It is especially important to do this for cluster analysis as the clustering may appear to be excellent in certain views, but have substantial overlap in others. One demonstration of this is figure 7.

4 Exploring cluster assignment

Cluster algorithms output a list of values assigning each observation to a cluster. This list is categorical, not ordinal, and while different clustering methods may recover the same clusters they might not give them the same cluster identifier. For this reason, it is useful to be able to match up similar clusters so that they have similar identifiers. This will reduce spurious

	A				
B	1	2	3	4	5
1	0	0	3	0	14
2	0	0	1	0	0
3	0	9	5	0	0
4	8	2	1	0	0
5	0	0	3	16	0

Rearrange rows ⇒

	A				
B	1	2	3	4	5
4	8	2	1	0	0
3	0	9	5	0	0
2	0	0	1	0	0
5	0	0	3	16	0
1	0	0	3	0	14

Table 1: Simulated data illustrating manual arrangement of a confusion matrix to aid interpretation.

visual differences between plots. Table 1 gives an example of how this rearrangement can be accomplished by hand.

Ideally, we want to able to do this rearrangement automatically for a large number of tables produced by different clustering algorithms. Unfortunately this problem is NP complete [15], and we must either limit ourselves to small numbers of clusters or use heuristics. For the small numbers of clusters used in this paper, we search through the space of all possible permutations, looking for the one with the smallest off diagonal, or equivalently the largest diagonal, sum. This is practical for up to eight clusters, after which generating the permutations and calculating the diagonal sums becomes prohibitively time consuming. We are investigating a heuristic method for larger numbers of clusters.

It is also nice to display this in a graphical form. One tool commonly used to do this is the heatmap, which we strongly discourage on perceptual grounds. A far more effective tool is the fluctuation diagram [10]. Where the heatmap maps the value to colour, the fluctuation diagram maps value to length on a common axis, which is easier to perceive [4]. Figure 2 demonstrates the two.

These methods can also be used for any other technique that produces a list of identifiers. For example, in supervised classification, they can be used to compare true and predicted values.

5 Examples

Here we demonstrate clustering methods from the three major families:

- Partitioning, with k-means clustering.
- Hierarchical, with agglomerative hierarchical clustering.
- Model based, with a normal mixture model approach

It is worthwhile to mention an alternative method to these automated techniques, which is is manual clustering as illustrated in [5, 21]. This method is much more time consuming, but is more likely to produce meaningful clusters. It makes few assumptions about cluster structure, and these assumptions can be easily modified if necessary.

To illustrate these three methods, we will use three different datasets, two from animal ecology, flea beetles and Australian crabs, and one demographic, US arrest rates in 1973.

The flea beetle data was originally described in [12]. It consists of measurements of six beetle body parts, from 74 beetles from three known species. The three species are clearly separated into three distinct groups, as shown in figure 3. This data set should be an easy test of a classification algorithm.

The Australian crabs data set [3] records five body measurements for 200 crabs of known sex and species. The shape is difficult to show statically, but by watching the grand tour for a while we see that the data is composed of four pencil shaped rods which converge to a point. Each one of the pencils is a separate combination of sex and species. The images in figure 4 attempt to show this using static graphics. There are four distinct groups in this example, but there is also high codimensionality, and we might expect algorithms to have more difficulty than with the flea data.

The third data set contains 1973 data on the number of arrests per 100,000 people for assault, murder and rape for each of the 50 states in the US, as well as the percent of resident of each state who live in urban areas [13]. There are no obvious clusters, a difficult assertion to prove here, but figure 5 shows three representative views. In this case, we are looking for the clustering algorithm to find useful groups.

All of the data sets were standardised by dividing each variable by its standard deviation to ensure that all variables lie on a common scale.

5.1 k-means clustering

The k-means algorithm [9] works by contrasting the sum of squared distances within clusters to the sum of squared differences between clusters, somewhat like

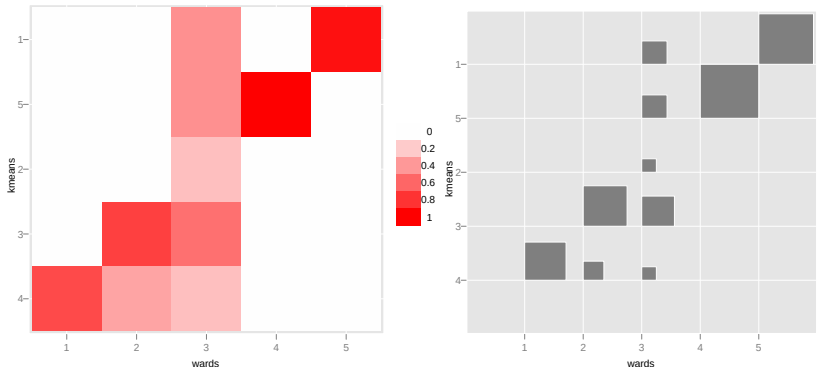

Figure 2: Heatmap on left, fluctuation diagram on right. Note that it is much easier to see subtle differences in the fluctuation diagram.

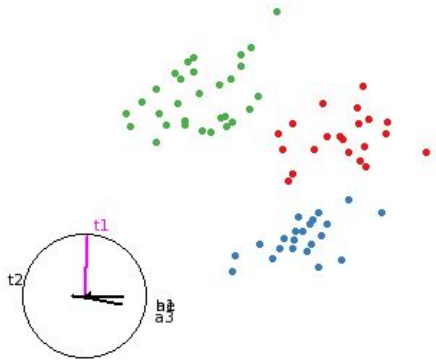

Figure 3: Projection from grand tour of flea data. Points are coloured by species. Note the clean separation into three groups

a 2-way ANOVA. To start k-means randomly assigns each point to a starting cluster, and then iteratively moves points between groups to minimise the ratio of the within to between sums of squares. It tends to produce spherical clusters.

Using the flea beetles data, figure 7 shows the clusters formed by the k-means algorithm with three groups. This clustering is stable, regardless of the initial random configuration selected. You can see that it has failed to retrieve the true clusters present in the data. The first view of the clusters shows this very clearly: you can see two red points amongst the green, and two green points amongst the red. In the second view of the data, the problem is not clear, and the clusters look perfectly adequate. I found these two views using the grand tour: it is dangerous to look at only a few 2D views of the data—there may be messages that you are missing.

Since we have a set of true cluster identifiers, we can compare the the true to the ones we found using the confusion matrix and fluctuation diagram, as shown in table 2 and figure 6.

The R code to produce these figures is very simple:

```
ref <- as.numeric(flea$species)
fl <- scale(as.matrix(flea[,1:6]))
x <- ggobi(flea)$flea
glyph_colour(x) <- ref
glyph_colour(x) <- clarify(
kmeans(fl, 3)$cluster,
ref
```

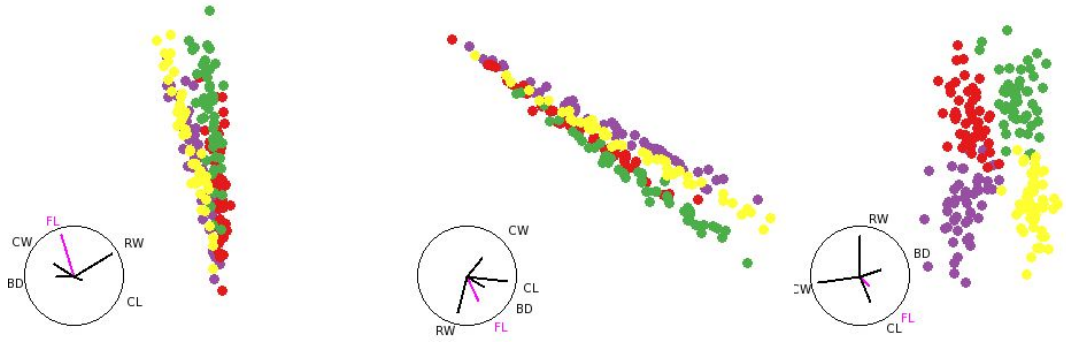

Figure 4: Three views of the crabs data set. The left and centre views show side of views of the four pencils, while the right view shows a zoomed in head on image.

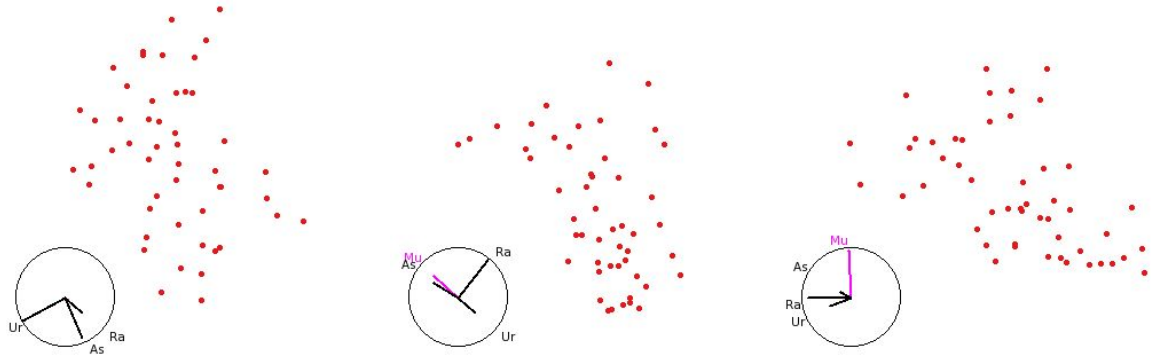

Figure 5: Three views of the arrests data. There are no obvious clusters.

)

Here, we first create a reference vector of the "true" clusters. We then scale the data and send it to GGobi, retaining a reference to the dataset in GGobi. The following two lines colour the points, first with the reference vector, and then with the results of the k-means clustering. The `clarify` function relabels the k-means result to match the reference vector as closely as possible. You can easily modify this code to use whatever clustering algorithm you are interested in.

The k-means algorithm is non-deterministic and running it multiple times may result in multiple cluster configurations, as shown in figure 8. It is interesting to view this as an animation. In practice, it is best to run the k-means algorithm from many random starting positions and then choose the one with the best score.

5.2 Hierarchical clustering

Hierarchical clustering methods work through either progression fusion (agglomerative) or progressive partitioning (divisive). Divisive methods are rarely used, so we focus here on agglomerative methods. Hierachical methods only require a matrix of interpoint distances, and so are easy to use with distance measures other than Euclidean.

Agglomerative methods build up clusters point by point, by joining the two points or two clusters which have the smallest distance [19]. To do this we need to define what we mean by the distance between two clusters (or one cluster and a point). There are a number of common methods: use the closest distance between points in the cluster (simple linkage, creates the minimal spanning tree), the largest distance (complete linkage), the average distance (UPGMA), or the distance between cluster centroids. Each of these methods finds clusters of somewhat different shapes: sin-

	True		
k-means	1	2	3
1	19	0	2
2	0	22	0
3	2	0	29

Table 2: Confusion matrix comparing "true" clusters with those from k-means clustering with three clusters.

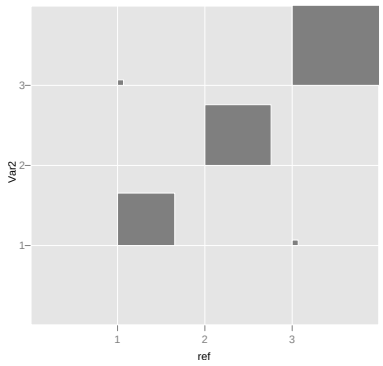

Figure 6: Fluctuation diagram, a visual representation of the data in 2

gle linkage forms long skinny clusters, average linkage forms more spherical clusters.

To illustrate some of these methods, we will use the arrests data, as described above. Figure 9 shows the results of the retrieving four clusters using complete linkage on correlation distance. It also illustrates another method we can use to highlight clusters: displaying the convex hull of the data. This technique should be used with caution as it makes the clusters look very distinct, possibly due to the gestalt principles of connectedness and closure [20].

Finally, we want to see how the clusters differ with respect to the original variables. We can do this interactively with parallel coordinates plots in GGobi, or statically in R. We have much more control over appearance in R and can choose whether the axes should be scaled to a common range, or we can use boxplots instead of lines. This different methods are shown in 10.

5.3 Model based clustering

Another way to define a cluster is to use an explicit density model. If we use a multivariate normal to model this density then we expect clusters to look spheroidal. Determining what the clusters are then becomes a mixture model problem, and in this context is known as model based clustering. This technique can be used with an specified density, but a multivariate normal model is most common to the simple parameterisation of correlation effects.

The model based clustering we use [8, 7], is based on a mixture of multivariate normals. Depending on the restrictions we place on the covariance matrix, we control the shapes, volumes and orientations of the clusters. If we estimate a covariance matrix for each cluster, then each cluster can have a different shape and orientation. If we estimate one covariance matrix, then all clusters must have the same orientation and shape. We can also place additional restrictions on the covariance matrix, for example, to make only spherical clusters.

Unlike the other cluster techniques, model based clustering can leverage its distributional assumptions to provide a way to select the best model and best number of clusters. Figure 11 illustrates this with a plot of the BIC statistic for each model. Model based clustering reclaims the flea beetle specifies clusters perfectly, as shown in figure 12.

Let's try model based clustering on a more difficult example, the Australian crabs data. From inspection of the plots (figure 4) we might expect that the best model will have clusters with similar size and shape, but pointing in different directions.

The best model is elliposidal, equal variance with 4 groups, which seems promising, but the BIC plot,

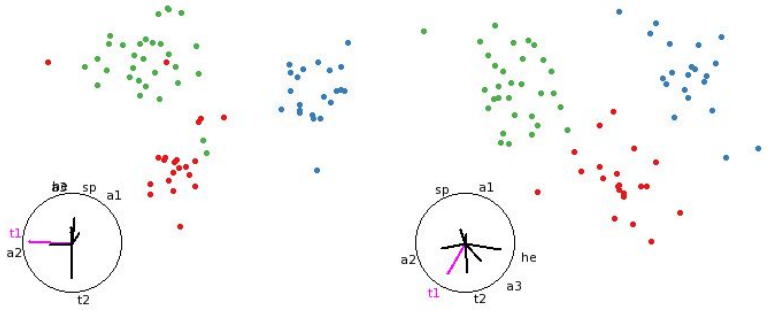

Figure 7: Two views of the k-means clustered data. Look at the errors in clustering! There are two red points and two green points that are obviously grouped erroneously, but we only see this in one of the two views. It is important to look at the results from many different directions!

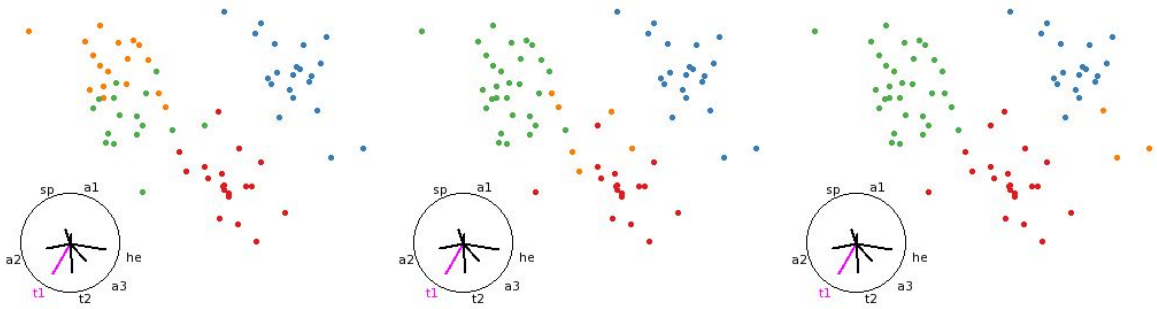

Figure 8: Three sets of results from a k-means clustering with four clusters. The result is highly dependent on the starting configuration.

figure 13, doesn't show any strong patterns. Lets look at what the best clustering found in figure 14.

The model based clustering hasn't reclaimed any the original groups but has instead split the species/sex combinations about half way along the pencil. A fluctuation plot makes this clear, see figure 15

6 Conclusion

This paper has presented an set of techniques for exploring the results of cluster analysis. Using R and GGobi provides a set of powerful tools for both statistical analysis and interactive graphics. It is easy to explore your data interactively, and it is easy to produce high quality output for publication.

We have only scratched the surface of what can be accomplished using RGGobi. In the future, we plan to build up these techniques into a cohesive family. We also plan to investigate interactive tuning of clustering parameters, so that as you adjust algorithm parameters you see the changes reflected immediately in GGobi.

References

[1] Rggobi2 website.

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

[3] N. A. Campbell and R. J. Mahon. A Multivariate Study of Variation in Two Species of Rock Crab of genus *leptograpsus*. *Australian Journal of Zoology*, 22:417–425, 1974.

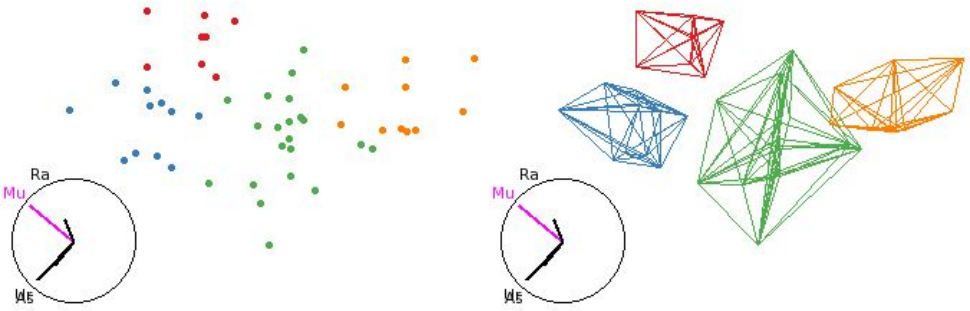

Figure 9: Left shows coloured points. Right shows convex hulls. Hulls help to see space that a group occupies, but makes the groups look very distinct. Uses complete linkage on correlation distances.

[4] William S. Cleveland and M. E. McGill. Graphical perception: Theory, experimentation and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[5] D. Cook, A. Buja, J. Cabrera, and C. Hurley. Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

[6] Dianne Cook, Andreas Buja, Eun-Kyung Lee, and Hadley Wickham. Grand tours, projection pursuit guided tours and manual controls. *Handbook of Computational Statistics: Data Visualization*, To appear.

[7] C. Fraley and A. E. Raftery. Model-based clustering,discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[8] C. Fraley, A.E. Raftery, Dept. of Statistics, and University of Washington. R port by Ron Wehrens. *mclust: Model-based cluster analysis*, 2005. R package version 2.1-11.

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

[10] Heike Hofmann. Exploring categorical data: interactive mosaic plots. *Metrika*, 51(1):11–26, 2000.

[11] Jürgen Hummel. Linked bar charts: Analysing categorical data graphically. *Journal of Computational Statistics*, 11:23–33, 1996.

[12] AA Lubischew. On the Use of Discriminant Functions in Taxonomy. *Biometrics*, 18:455–477, 1962.

[13] D. R McNeil. *Interactive Data Analysis*. Wiley, New York, 1977.

[14] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[15] Harri Siirtola and Erkki Mäkinen. Constructing and reconstructing the reorderable matrix. *Information Visualization*, 4(1):32–48, 2005.

[16] Deborah F. Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. Ggobi: Evolving from xgobi into an extensible framework for interactive data visualization. *Journal of Computational Statistics and Data Analysis*, 43:423–444, 2003. http://authors.elsevier.com/sd/article/S0167947302002864.

[17] DF Swayne, A Buja, and N Hubbell. Xgobi meets s: integrating software for data analysis. *Computing Science and Statistics*, 23:430–434, 1991.

[18] Duncan Temple Lang and Deborah F Swayne. Ggobi meets r: an extensible environment for interactive dynamic data visualization. In *Proceed-*

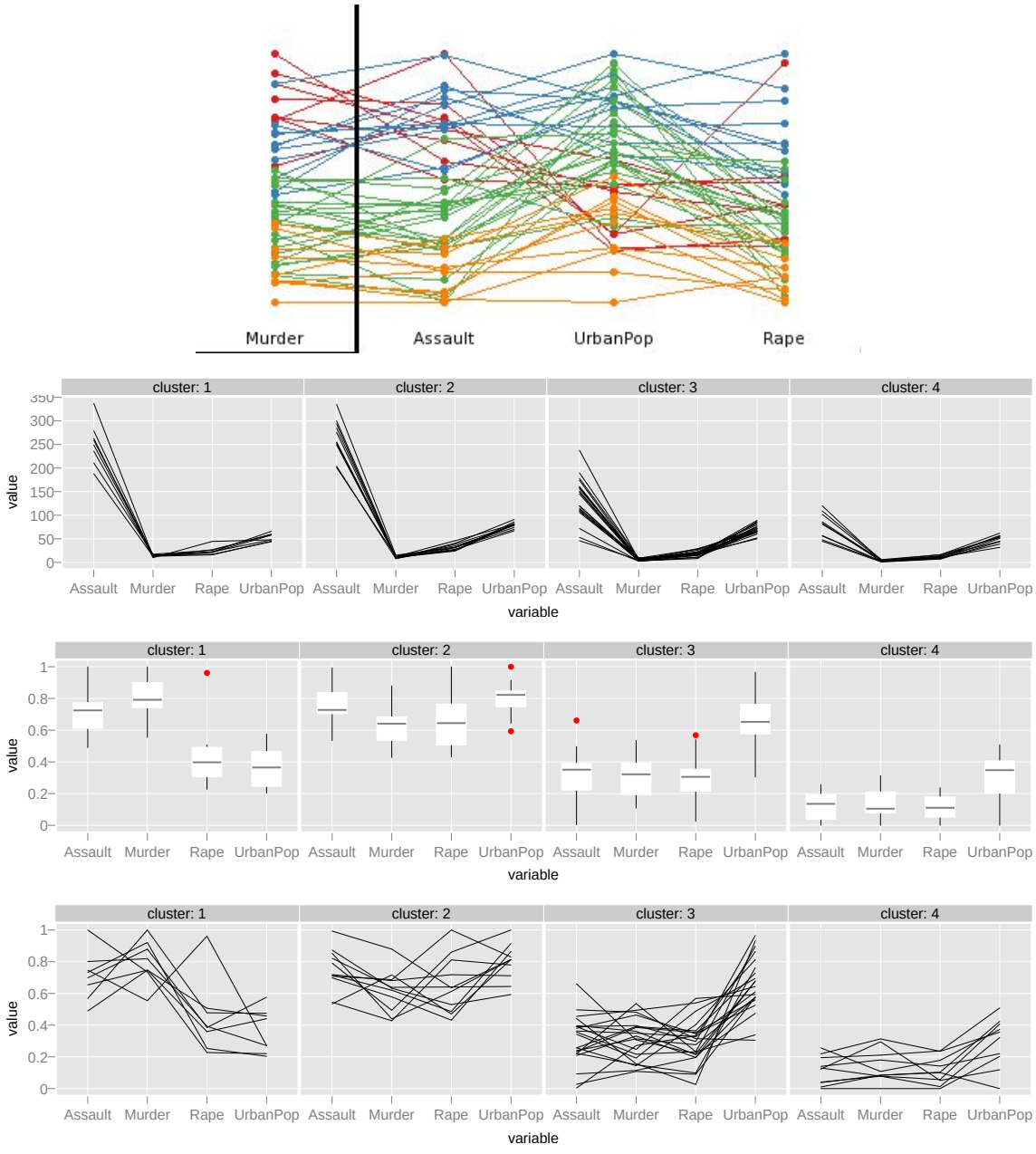

Figure 10: A variety of parallel coordinates plots can we can create with either GGobi or R.

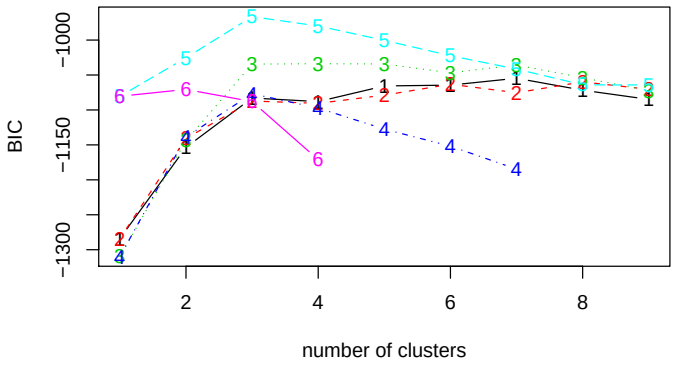

Figure 11: Summary plot for model based clustering of the flea data. Model 5 (EEV, elliposidal, equal variance) is best at almost all numbers of cluster and peaks at three clusters.

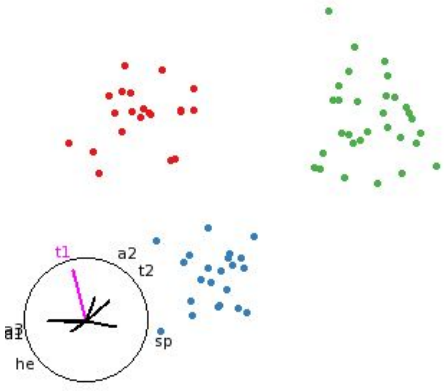

Figure 12: Model based clustering of the flea beetle data. This clustering retrieves the original species clusters perfectly.

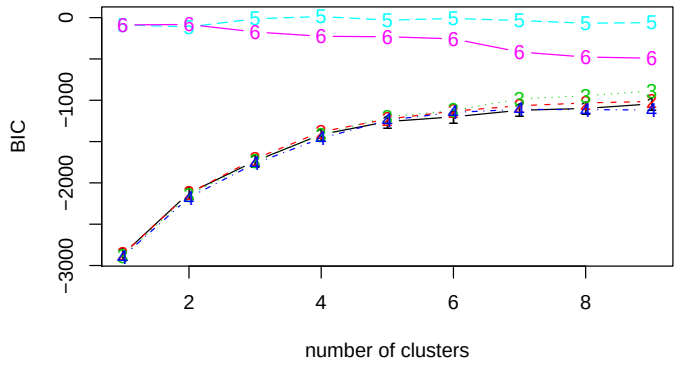

Figure 13: Summary plot for model based clustering of the Australian crabs data. There is no clear winner.

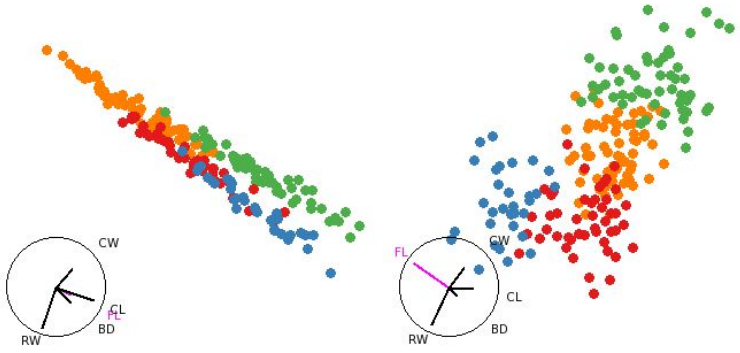

Figure 14: BIC statistic to help choose the number of clusters and the appropriate model

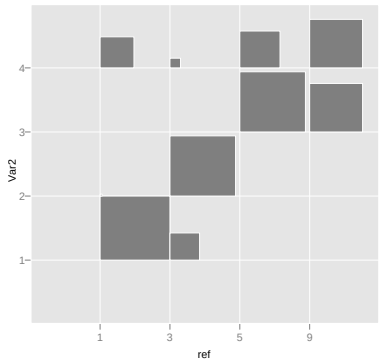

Figure 15: Fluctuation plot emphasising the difference between the "true" crabs clusters and the clusters generated with model based clustering.

ings of the 2nd International Workshop on Distributed Statistical Computing, 2001.

[19] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[20] Colin Ware. *Information Visualization - Perception for Design*. Morgan Kaufmann Publishers, 2nd edition, 2004.

[21] AFX. Wilhelm, EJ Wegman, and J Symanzik. Visual Clustering and Classification: The Oronsay Particle Size Data Set Revisited. *Computational Statistics: Special Issue on Interactive Graphical Data Analysis*, 14(1):109–146, 1999.