Lab. notes for cluster analysis

Paul Hewson

October 1, 2007

1 Preliminaries and exploratory data analysis

We start off by loading a useful support library, the `cluster` library. We need to load the `flexclust` library to get hold of some interesting data. Make that data available (the `milk` data) and start doing an exploratory data analysis. The suggested code here includes star plots (quite interesting here) and faces (not very useful). You may also consider parallel co-ordinates plots and scatterplots, but we will come back to them later.

```
> library(cluster)
> library(flexclust)
> library(MASS)
> library(TeachingDemos)
> data(milk)
> stars(milk, full = FALSE, draw.segments = TRUE,
+     key.loc = c(10,0.4), main= "Milk data")
> faces(milk, fill = TRUE, ncol = 4, scale = TRUE, nrow = 7)
```

2 Calculating distances and performing hierarchical clustering

The first piece of clustering will demonstrate the defaults, i.e. euclidean distance and complete linkage. A good habit is to create a distance object, then create a `hclust` object and then extract whatever information we need from that `hclust` object. For example, `plot()` will produce a dendrogram and `cutree()`, given an argument for `k` will "cut" the tree at the given number of clusters (you can also cut based on height if you prefer)

```
> milk.dist <- dist(milk)
> milk.hclust <- hclust(milk.dist)
> plot(milk.hclust)
> milk.cut <- cutree(milk.hclust, 3)
```

It is of interest investigating the stability of your cluster solution to different choices of distance measure, for example using the Manhattan distance:

```
> milk.dist.man <- dist(milk, method = "manhattan")
> milk.hclust.man <- hclust(milk.dist.man)
> plot(milk.hclust.man)
> milk.cut.man <- cutree(milk.hclust.man, 3)
> xtabs(~milk.cut+milk.cut.man)
```

or the Minkowski, with $\lambda = 5$ (R calls this `p`)

```
> milk.dist.min5 <- dist(milk, method = "minkowski", p = 5)
> milk.hclust.min5 <- hclust(milk.dist.min5)
> plot(milk.hclust.min5)
> milk.cut.min5 <- cutree(milk.hclust.min5, 3)
> xtabs(~milk.cut+milk.cut.min5)
```

Also, one may wish to consider different clustering strategies, in this case we consider Ward's method (based on the Euclidean distance):

```
> milk.hclust.ward <- hclust(milk.dist, method = "ward")
> plot(milk.hclust.ward)
> milk.cut.ward <- cutree(milk.hclust.ward, 3)
> xtabs(~milk.cut+milk.cut.ward)
```

You should actually find for these data, whatever you do yields quite a stable three cluster "solution".

3 Visualising the solution in terms of the data

We can use some standard exploratory data analysis techniques, only in this case it is possible to "label" the rows in terms of the cluster membership we have proposed:

```
> parcoord(milk, col = milk.cut, lty = milk.cut,
+   main = "milk data, four group clustering")
> pairs(milk,
+          lower.panel = function(x, y){ points(x, y,
+          pch = milk.cut, col = milk.cut)},
+          main = "Scatterplot for milk data")
```

4 k-means clustering

k-means clustering assumes we have some idea of k before we start. In this example, we seem to be quite sure that $k = 3$, that won't necessarily be the case with other problems. The `kmeans` object contains other information about the solution, such as the centroids, but the cluster assignments are easily extracted and compared with the solutions we found earlier.

```
> milk.kmeans <- kmeans(milk, centers = 3)
> milk.kmeans$cluster
> xtabs(~milk.cut+milk.kmeans$cluster)
```

Do note that k-means clustering is applied to the *data* and *not* to the distance matrix!!!!!!

5 Measures of fit

This is a massive area. We consider just one possibility amongst many based on the cophenetic distance, a measure of the distance between the level at which two points are merged. Measuring the correlation between this distance and the original distance matrix tells us something about how well a given dendrogram represents a given distance matrix.

```
> d.retained <- cophenetic(milk.hclust)
> cor(milk.dist, d.retained)
```

(Too) many more possibilities are given in `cluster.stats()` in `library(fpc)`. One of the more common ones is the Rand statistic, which can be used to compare two cluster solutions. Again, we supply the original distance matrix, and then the cluster solution we have chosen and an alternative. You should find with the milk data that the Rand statistic = 1, as these solutions agree perfectly. The helpfile for this function gives you references to the literature if you want to know what all the other statsitics are measuring.

```
> library(fpc)
> cluster.stats(milk.dist, milk.cut, alt = milk.cut.man)
```