$k$-means clustering

```r
km15 = kmeans(x[g==0,],5)
km25 = kmeans(x[g==1,],5)
for(i in 1:6831){
  md = c(mydist(xnew[i,],km15$center[1,]),mydist(xnew[i,],km15$center[2,]),
        mydist(xnew[i,],km15$center[3,]),mydist(xnew[i,],km15$center[4,]),
        mydist(xnew[i,],km15$center[5,]),mydist(xnew[i,],km25$center[1,]),
        mydist(xnew[i,],km25$center[2,]),mydist(xnew[i,],km25$center[3,]),
        mydist(xnew[i,],km25$center[4,]),mydist(xnew[i,],km25$center[5,]))
  mark = which(md == min(md))
  nearest[i] = ifelse(mark <= 5, "blue", "orange")
}
plot(xnew, type="n", xlab = "x1", ylab = "x2",
     main= "kmeans with 5 cluster centers")
points(xnew, col=nearest, pch=".")
points(km25$centers, col="orange", pch=19, cex=2)
points(km15$centers, col="blue", pch=19, cex=2)
points(x, col = ifelse(g==0, "blue","orange"))
```
data(mixture.example)

p.16:
\[ m_{1k} \sim \mathcal{N}_2((1, 0)', I), \ k = 1, \ldots, 10; \ m_{2k} \sim \mathcal{N}_2((0, 1)', I), \ k = 1, \ldots, 10 \]

blueX <- mvrnorm(100, mu = m1[sample(10, 1), ], Sigma = matrix(c(1, 0, 0, 1), ncol=2))
The curse of dimensionality (§2.5)

- “local” in $R^1$ is quite different than local in $R^p$
- Example: each feature variable uniformly distributed on $(0, 1)$.
- want 10% of the sample in $R^1$: need a window of length 0.1.
- want 10% of the sample in $R^p$: need a box with edge length $0.1^{1/10} = 0.80$
- on each axis need a window of length 0.8.
- Figure 2.6
... curse

- Example: $N$ data points uniformly distributed on a unit ball in $\mathbb{R}^p$.
- Distance from the origin to the nearest data point?
- Median: $(1 - 0.5^{1/N})^{1/p} \approx 0.52$ if $p = 10$, $N = 500$. 

![Graph showing the median distance for different values of $N$ and $p$]
Cluster Analysis (§14.3)

- discover groupings among the cases; cases within clusters should be ’close’ and clusters should be ’far apart’
- Figure 14.4
- many (not all) clustering methods use as input an $N \times N$ matrix $D$ of dissimilarities
- require $D_{ij} > 0$, $D_{i'j'} = D_{i'j}$ and $D_{ii} = 0$
- sometimes the data are collected this way (see §14.3.1)
- more often $D$ needs to be constructed from the $N \times p$ data matrix
- often (usually) $D_{ij} = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$, where $d_j(\cdot, \cdot)$ to be chosen, e.g. $(x_{ij} - x_{i'j})^2$, $|x_{ij} - x_{i'j}|$, etc.
- See p 504, 505 for more details on choosing a type of dissimilarity matrix
- this can be done using `dist` or `daisy` (the latter in the R library `cluster`)
... cluster analysis

- dissimilarities for categorical features
- binary: simple matching uses

\[ D_{ij} = \frac{\# \{ (1, 0) \text{ or } (0, 1) \text{ pairs } \}}{p} \]

Jacard coefficient uses

\[ D_{ij} = \frac{\# \{ (1, 0) \text{ or } (0, 1) \text{ pairs } \}}{\# \{ (1, 0), (0, 1) \text{ or } (1, 1) \text{ pairs } \}} \]

- ordered categories – use ranks as continuous data (see eq. (14.23))
- unordered categories – create binary dummy variables and use matching
... cluster analysis

dist(x, method = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski"))

where maximum is $\max_{1 \leq j \leq p}(x_{ij} - x_{i'j})$ and binary is Jacard coefficient.

daisy(x, metric=c("euclidean", "manhattan", "gower"), standardize=F, type=c("ordratio","logratio","asymm","symm"))

(see the help files)

> x = matrix(rnorm(100),nrow=5)
> dim(x)
[1] 5 20
> dist(x)
     1    2    3    4
2 5.493679
3 6.360923 5.652732
4 7.439924 5.885949 7.960187
5 4.437444 3.679995 6.133873 5.936607
Combinatorial algorithms

Suppose number of clusters $K$ is fixed ($K < N$)

$C(i) = k$ if observation $i$ is assigned to cluster $k$

$$T = \frac{1}{2} \sum_{i=1}^{N} \sum_{i' = 1}^{N} D_{ii'}$$

$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i) = k} \left( \sum_{C(i') = k} D_{ii'} + \sum_{C(i') \neq k} D_{ii'} \right)$$

$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i) = k} \sum_{C(i') = k} D_{ii'} + \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i) = k} \sum_{C(i') \neq k} D_{ii'}$$

$$= W(C) + B(C)$$

$W(C)$ is a measure of within cluster dissimilarity

$B(C)$ is a measure of between cluster dissimilarity

$T$ is fixed given the data: minimizing $W(C)$ same as maximizing $B(C)$
K-Means clustering (§14.3.6)

- most algorithms use a ’greedy’ approach by modifying a given clustering to decrease within cluster distance: analogous to forward selection in regression

- $K$-means clustering is (usually) based on Euclidean distance: $D_{ij'} = ||x_i - x_{i'}||^2$, so $x$’s should be centered and scaled (and continuous)

- Use the result

\[
\frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} ||x_i - x_{i'}||^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2
\]

where $N_k$ is the number of observations in cluster $k$ and $\bar{x}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$ is the mean in the $k$th cluster

- The algorithm starts with a current set of clusters, and computes the cluster means. Then assign observations to clusters by finding the cluster whose mean is closest. Recompute the cluster means and continue.
Constructing dissimilarity matrices

\[ \text{dist}(x, \text{method} = \text{c}("euclidean", "maximum", "manhattan", "canberra", "binary")) \]

where \text{maximum} is \( \max_{1 \leq j \leq p}(x_{ij} - x_{i'j}) \) and \text{binary} is Jacard coefficient.

\[ \text{daisy}(x, \text{metric}=\text{c}("euclidean", "manhattan", "gower"), \text{standardize}=F, \text{type}=\text{c}("ordratio","logratio","asymm","symm")) \]

(see the help files)
Hierarchical clustering \[\textcircled{14.3.12}\]

- no specification of number of clusters
- top down = divisive; bottom up = agglomerative
- bottom up: each value is a cluster, cluster the closest pair of points, iterate: find the closest pair of clusters \(C_i\) and \(C_{i'}\), merge them
- need a measure for distance between points and between clusters (the clusters needn’t be vectors)
- **single link** clustering measures the distance between clusters by the minimum distance
  
  \[d(C_1, C_2) = \min_{i \in C_1, i' \in C_2} D_{ii'}\]
  
  - susceptible to ‘chaining’; long strings of points assigned to the same cluster
  - sensitive to outliers
- **complete linkage**
  
  \[d(C_1, C_2) = \max_{i \in C_1, i' \in C_2} D_{ii'}\]
  
- **group average** intermediate between complete and single linkage.
... hierarchical clustering

- easily pictured in a dendogram
- Figs 14.12 and 14.13
- ’look’ is quite different for different linkages
- Implemented in R in `hclust` and `agnes`.  

13 / 16