36-490 Spring 2010: Clustering (and things...)

Brian Junker

April 5, 2010

- MASS Book References
- Quick Reprise: Classification With Logistic Regression
- Clustering and Mixture Models
- A Simple Clustering Idea: Mixture of Normals
- Another Way to Think About the Normal Mixture Model
- K-means Clustering
- Distance-Based Clustering
- Hierarchical Clustering

MASS Book References

W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S (Fourth Edition)*, New York: Springer, 2002.

Logistic Regression: Section 7.2, pp. 190ff.

Poisson Regression: Section 7.3, pp. 199ff. (won't seem a lot like Jim's example from last time, but it really is the same thing)

Clustering: Section 11.2, pp. 315ff. (all of Ch 11 is interesting!)

Classification: Chapter 12, pp. 331ff. (extensions and alternatives to logistic regression)

Quick Reprise: Classification With Logistic Regression

Logistic Regression

$$\log \frac{P[Y=1|X_1,X_2,X_3]}{P[Y=0|X_1,X_2,X_3]} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

Use `glm(Y ~ X1 + X2 + X3, data=mydata, family=binomial)` in R.

Polytomous Logistic Regression

$$\begin{aligned}
\log \frac{P[Y=1|X_1,X_2,X_3]}{P[Y=K|X_1,X_2,X_3]} &= \beta_{10} + \beta_{11} X_1 + \beta_{12} X_2 + \beta_{13} X_3, \\
\log \frac{P[Y=2|X_1,X_2,X_3]}{P[Y=K|X_1,X_2,X_3]} &= \beta_{20} + \beta_{21} X_1 + \beta_{22} X_2 + \beta_{23} X_3, \\
&\vdots \\
\log \frac{P[Y=K-1|X_1,X_2,X_3]}{P[Y=K|X_1,X_2,X_3]} &= \beta_{(K-1)0} + \beta_{(K-1)1} X_1 + \beta_{(K-1)2} X_2 + \beta_{(K-1)3} X_3.
\end{aligned}$$

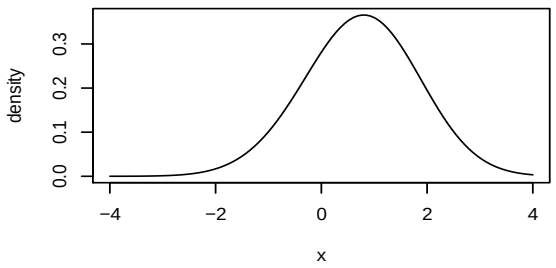
Install `VGAM` from CRAN. Then `library(VGAM)`. Then ... (huh?)

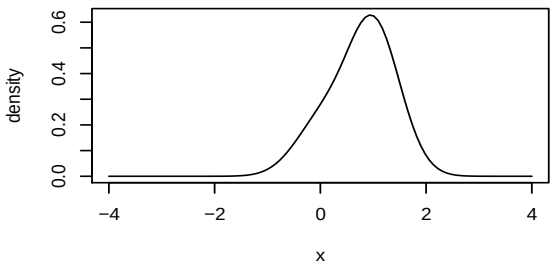
See R notes for VGAM Example. . .

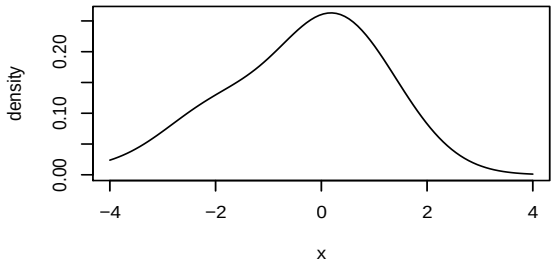
Clustering and Mixture Models

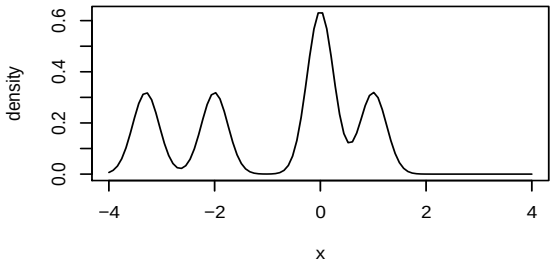
- Cluster analysis (a.k.a. data segmentation) is a basic method of exploratory data analysis (unsupervised learning).
- Grouping by "similarity": observations within a cluster are more similar to each other than observations between clusters.
- Two basic kinds:
  - *Model-Based Clustering* [This time]
    - Mixture-of-Normals
    - Other Mixtures, Latent Class Analysis
  - *Distance-Based Clustering* [Next time]
    - Measures of similarity
    - Proximity matrices
    - Methods for merging or breaking apart clusters (Dendrograms, Heierarchical Clustering. . .)

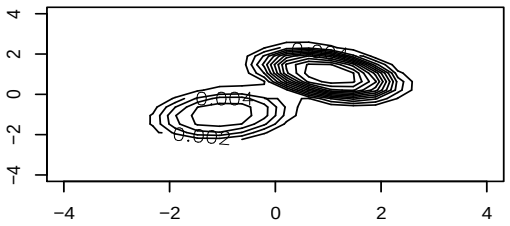
A Simple Clustering Idea: Mixture of Normals

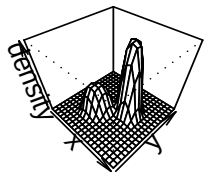






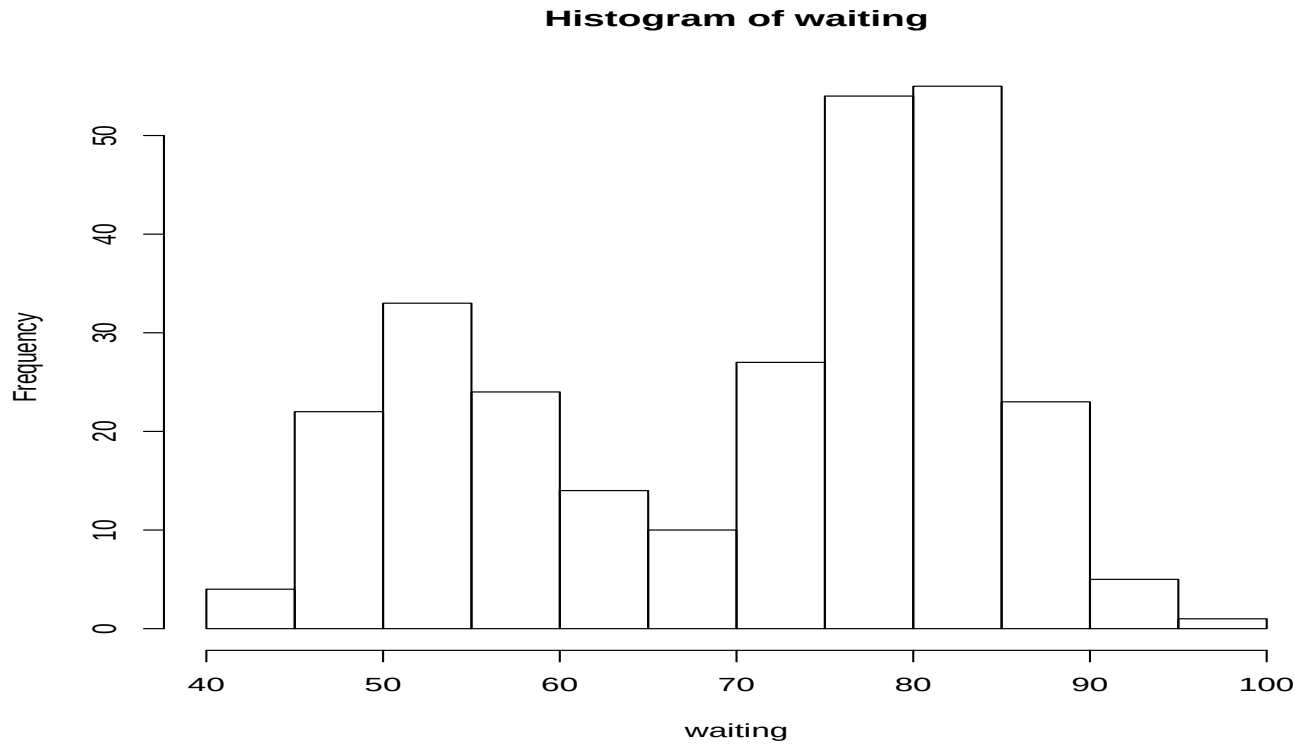






Example: Old Faithful Waiting Times

```
> library(MASS)
> data(faithful)
> attach(faithful)
> hist(waiting)
```



suggests

$$y_i \mid \mu_1, \mu_2, \sigma_1, \sigma_2, q \overset{iid}{\sim} \left\{ q \frac{1}{\sqrt{2\pi}\sigma_1} \exp[-(y_i - \mu_1)^2/2\sigma_1^2] + (1-q) \frac{1}{\sqrt{2\pi}\sigma_2} \exp[-(y_i - \mu_2)^2/2\sigma_2^2] \right\} \quad (*)$$

$i = 1, \ldots, n$ ($n = 272$).

Let $f_1(y_i|\mu_1, \sigma_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp[-(y_i - \mu_1)^2/2\sigma_1^2]$, and $f_2(y_i|\mu_2, \sigma_2) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp[-(y_i - \mu_2)^2/2\sigma_2^2]$. Then the likelihood is

$$g(y|\mu_1, \sigma_1, \mu_2, \sigma_2, q) = \prod_{i=1}^{n} \{q f_1(y_i|\mu_1, \sigma_1) + (1-q) f_2(y_i|\mu_2, \sigma_2)\}.$$

We wish to estimate the parameters μ_1, μ_2, σ_1, σ_2 and q.

`nlm()` in R can *minimize* a function (using Newton's method). For example,

```
> nll <- function(phi) {
+   a  <- phi[1]
+   m1 <- phi[2]
+   s1 <- phi[3]
+   m2 <- phi[4]
+   s2 <- phi[5]
+   return(-sum(log(a*dnorm(y,m1,s1) + (1-a)*dnorm(y,m2,s2))))
+ }
> nlm(nll,c(.25,52,10,82,10))
$minimum
[1] 1034.002
$estimate
[1]  0.3608861 54.6148563  5.8712181 80.0910688  5.8677343
$gradient
[1]  5.903871e-05  1.476694e-06 -6.176736e-06
[4] -2.409947e-06 -2.751431e-06
```

See R notes for more...

Model-based Clustering

$y_1, \ldots, y_n$ are iid according to

$$f(y|\phi) = \sum_{k=1}^{K} \alpha_k \, n_d(y|\mu_k, \Sigma_k)$$

where $\phi = (\alpha_1, \ldots, \alpha_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$, $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$.

- Can be viewed as a kind of density estimation model.
- Can also be viewed as a means of idenfitying clumps or clusters in data.
- Can implement with the `mclust` package (Download from CRAN!, `library(mclust)`, function `Mclust`...

See R notes for Mclust example(s)...

Another Way to Think About the Normal Mixture Model

- $y_1, \ldots, y_n$ are iid according to the density

$$\begin{aligned} f(y_i|\mu_1, \sigma_1, \mu_2, \sigma_2, q) &= q \cdot \tfrac{1}{\sqrt{2\pi}\sigma_1} \exp[-(y_i - \mu_1)^2/2\sigma_1^2] \\ &\quad + (1-q) \cdot \tfrac{1}{\sqrt{2\pi}\sigma_2} \exp[-(y_i - \mu_2)^2/2\sigma_2^2] \\ &= q \cdot n(y|\mu_1, \sigma_1^2) + (1-q) \cdot n(y|\mu_2, \sigma_2^2) \end{aligned}$$

- An alternative to Newton-Raphson (`nlm()` in R):

  Step 1: Decide which "hump" each observation y_i belongs in, depending on whether $n(y_i|\mu_1, \sigma_1^2) > n(y_i|\mu_2, \sigma_2^2)$ or vice-versa.

  Step 2: Estimate $\hat{\mu}_1$ and $\hat{\sigma}_1^2$ as the sample mean and variance of the y's assigned to the first "hump"; and estimate $\hat{\mu}_2$ and $\hat{\sigma}_2^2$ as the sample mean and variance of the y's assigned to the second "hump".

  Iterate these two steps until no more changes.

  This is an "E-M" algorithm (Step 1 puts data points in the **E**xpected cluster, Step 2 calculates **M**aximum likelihood estimates).

  Clearly this would generalize to more than two clusters!

K-means Clustering

Suppose we hypothesize K clusters. Let $z_i = k$ if y_i is in the k^{th} cluster.

For squared Euclidean distance between points $d_{ii'} = D(y_i, y_{i'}) = \|y_i - y_{i'}\|^2$, we can write the *total scatter* as

$$\begin{aligned} T &= \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{K} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{z_i=k}\left(\sum_{z_{i'}=k} d_{ii'} + \sum_{z_{i'}\neq k} d_{ii'}\right) \\ &= W + B \end{aligned}$$

so we can choose z to minimize W or maximize B. To minimize W, note that

$$W = \frac{1}{2}\sum_{k=1}^{K}\sum_{z_i=k}\sum_{z_{i'}=k} d_{ii'} = \sum_{k=1}^{K}\sum_{z_i=k}\|y_i - \bar{y}_k\|^2 \qquad (*)$$

The K-means algorithm is a heuristic for minimizing $(*)$ in two alternating stages:

The K-means Algorithm

Step 1: Given a set of cluster means $\bar{y}_k$, redefine the z_i's by assigning each observation y_i to the closest $\bar{y}_k$.

Step 2: Given a set of cluster assignments $z_1, \ldots, z_n$, find the means $\bar{y}_k = \frac{1}{n_{\text{cluster } k}} \sum_{z_i=k} y_i$; this minimizes $(*)$ for the current z_i's,

Iterate these two steps until no more changes.

Notes:

- This is basically an "abstraction" of the E-M algorithm for normal mixtures: ignore the probability model part and just keep re-assigning points to clusters until you minimize the within-cluster spreads.
- Like E-M and Newton-Raphson it is a *local* algorithm; *Many different choices for the starting means should be tried.* Hartigan and Wong (1979) provide some improvements (ensure no single re-assignment improves $(*)$).
- If $d_{ii'}$ is not squared Euclidean distance, then $\bar{y}_k$ may not be a good representative of the center of a cluster. If we use one of the observations to represent the center, it is called "K-medoids".

Distance-Based Clustering

- *Objects:* $y_1, \ldots, y_n$
- *Attributes:* $y_{i1}, \ldots, y_{id}$
- $n \times n$ *proximity matrix* $D(y_i, y_{i'})$ (distance, (dis-)similarity, etc.)
  - $D(.,.)$ can be basic data (e.g., marketing, political science, perception research: human judgement of similarity/dis-similarity of objects); or
  - $D(y_i, y_{i'})$ can be a direct numerical measure of attributes, e.g.:
    $D(y_i, y_{i'}) = \sum_{\ell=1}^{d} dist(y_{i\ell}, y_{i'\ell})$
    * $dist(y_{i\ell}, y_{i'\ell}) = |y_{i\ell} - y_{i'\ell}|^p$:
  - $D(y_i, y_{i'}) = Corr(y_i, y_{i'}) = \cos(\theta_{y_i, y_{i'}})$, etc.
  - An explicit measure of similarity between categorical variables...
- Usually $D(.,.)$ is chosen to satisfy

  Triangle Inequality: $D(y_1, y_3) \leq D(y_1, y_2) + D(y_2, y_3)$; or
  Ultrametric Inequality: $D(y_1, y_3) \leq \max[D(y_1, y_2), D(y_2, y_3)]$

Hierarchical Clustering

- Produce a nested set of clusterings to choose from.
- Can be *agglomerative* (bottom up, from n individual-observation clusters) or *divisive* (top-down, starting from a single cluster).
- Both approaches can be represented by a *dendrogram* (tree diagram)
- Extend $D(.,.)$ to measure *similarity/proximity between clusters*.
  - In agglomerative clustering, the two most similar clusters are merged at each stage.
  - In divisive clustering the cluster with the greatest within-cluster dis-similarity is split, by first splitting off the most dissimilar observation, and then separating the cluster into two, analogously to K-means/K-medoids. (Kaufman and Rousseeuw, 1990).
- The dendrogram usually indicates along the y-axis the values at which splits take places.

Extending $D(.,.)$ to clusters

Let G and H represent two clusters. The usual approaches to extending $d_{ii'} = D(y_i, y_{i'})$ to G and H are:

- *Single Linkage:* (nearest-neighbor)

$$d_{SL}(G,H) = \min_{i \in G, i' \in H} d_{ii'}$$

- *Complete Linkage:* (farthest-neghbor)

$$d_{CL}(G,H) = \max_{i \in G, i' \in H} d_{ii'}$$

- *Average Linkage:*

$$d_A(G,H) = \frac{1}{N_G}\frac{1}{N_H}\sum_{i \in G}\sum_{i' \in H} d_{ii'}$$

(where N_G and N_H are the number of observations in G and H, respectively).

Pro's and Con's

- If the "natural" clusters in the data are compact and well-separated, then all three approaches produce similar results.
- d_{SL} tends to combine observations linked by a series of close intermediaries ("chaining"). This can produce large-diameter clusters with local, but not global, coherence.

$$diam(G) = \max_{i \in G, i' \in G} d_{ii'}$$

- d_{CL} tends to produce smaller-diameter clusters, but sometimes produces clusters containing observations that are closer to other clusters.
- d_A estimates $E[D(y, y')]$ where y and y' are independent random draws from the cluster densities $f_G(y)$ and $f_H(y')$. On the other hand d_A is sensitive to montonote transformations of $d_{ii'}$; whereas d_{SL} and d_{CL} are not.