Notes

- Friday tutorial on R programming
- reminder office hours on 2-3 F; 3-4 R
- The book ”Modern Applied Statistics with S” by Venables and Ripley is very useful. Make sure you have the MASS library available when using R or Splus (in R type `library(MASS)`).
- All the code in the 4th edition of the book is available in a file called ”`scripts`”, in the MASS subdirectory of the R library. On Cquest this is in `/usr/lib/R/library`.
- Undergraduate Summer Research Awards (USRA): see Statistics office SS 6018: application due Feb 18

Likelihood methods

- log-likelihood

$$\ell(\beta) = \Sigma_{i=1}^{N} y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})$$

- Maximum likelihood estimate of β:

$$\frac{\partial \ell(\beta)}{\partial \beta} = 0 \iff \Sigma_{i=1}^{N} y_i x_{ij} = \Sigma p_i(\hat{\beta}) x_{ij}, \quad j = 1, \ldots, p$$

- Fisher information

$$-\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = \Sigma_{i=1}^{N} x_i x_i^T p_i(1 - p_i)$$

- Fitting: use an iteratively reweighted least squares algorithm; equivalent to Newton-Raphson; p.99
- Asymptotics: $\hat{\beta} \xrightarrow{d} N(\beta, \{-\ell''(\hat{\beta})\}^{-1})$

Inference

- Component: $\hat{\beta}_j \approx N(\beta_j, \hat{\sigma}_j) \quad \hat{\sigma}_j^2 = [\{-\ell''(\hat{\beta})\}^{-1}]_{jj}$; gives a t-test (z-test) for each component
- $2\{\ell(\hat{\beta}) - \ell(\beta_j, \tilde{\beta}_{-j})\} \approx \chi^2_{\dim \beta_j}$; in particular for each component get a χ^2_1, or equivalently
- $\mathrm{sign}(\hat{\beta}_j - \beta_j)\sqrt{[2\{\ell(\hat{\beta}) - \ell(\beta_j, \tilde{\beta}_{-j})\}} \approx N(0,1)$
- To compare 2 models $M_0 \subset M$ can use this twice to get $2\{\ell_M(\hat{\beta}) - \ell_{M_0}(\tilde{\beta}_q)\} \approx \chi^2_{p-q}$ which provides a test of the adequacy of M_0
- LHS is the difference in (residual) deviances; analogous to SS in regression
- See Ch. 14 of 302 text, and algorithm on p.99 of HTF. (See Figure 4.12) (See R code)

extensions

- $E(y_i) = p_i$, $\mathrm{var}(y_i) = p_i(1 - p_i)$ under Bernoulli
- Often the model is generalized to allow $\mathrm{var}(y_i) = \phi p_i(1 - p_i)$; called over-dispersion
- Most software provides an estimate of ϕ based on residuals.

- if $y_i \sim Binom(n_i, p_i)$ same model applies
- $E(y_i) = n_i p_i$ and $\mathrm{var}(y_i) = n_i p_i(1 - p_i)$ under Binomial

- Model selection uses a C_p-like criterion called AIC
- In Splus or R, use `glm` to fit logistic regression, `stepAIC` for model selection

Logistic regression (§4.4)

```
> hr <- read.table("heart.data",header=T)
> dim(hr)
[1] 462  11
> hr <- data.frame(hr)
> pairs(hr[2:10],pch=21,bg=c("red","green")[codes(factor(hr$chd))])
> glm(chd~sbp+tobacco+ldl+famhist+obesity+alcohol+age,
+ family=binomial,data=hr)

Call:  glm(formula = chd ~ sbp + tobacco + ldl + famhist + obesity + alcohol + age,
family = binomial, data = hr)

Coefficients:
   (Intercept)             sbp         tobacco             ldl
    -4.1290787       0.0057608       0.0795237       0.1847710
famhistPresent         obesity         alcohol             age
     0.9391330      -0.0345467       0.0006058       0.0425344

Degrees of Freedom: 461 Total (i.e. Null);  454 Residual
Null Deviance:     596.1
Residual Deviance: 483.2  AIC: 499.2
> hr.glm <-.Last.value
> coef(hr.glm)
   (Intercept)            sbp        tobacco            ldl famhistPresent
 -4.1290787150   0.0057608299   0.0795237250   0.1847709867   0.9391330412
       obesity        alcohol            age
 -0.0345466980   0.0006058453   0.0425344469
```

Logistic regression (§4.4)

```
> summary(hr.glm)

Call:
glm(formula = chd ~ sbp + tobacco + ldl + famhist + obesity +
    alcohol + age, family = binomial, data = hr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7517  -0.8379  -0.4552   0.9292   2.4432

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.1290787  0.9606826  -4.298 1.72e-05 ***
sbp             0.0057608  0.0056250   1.024  0.30577
tobacco         0.0795237  0.0261876   3.037  0.00239 **
ldl             0.1847710  0.0573161   3.224  0.00127 **
famhistPresent  0.9391330  0.2243638   4.186 2.84e-05 ***
obesity        -0.0345467  0.0290531  -1.189  0.23440
alcohol         0.0006058  0.0044490   0.136  0.89168
age             0.0425344  0.0101295   4.199 2.68e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 483.17  on 454  degrees of freedom
AIC: 499.17

Number of Fisher Scoring iterations: 3
```

Logistic regression (§4.4)

```
> library(MASS)
> hr.step <- stepAIC(hr.glm,trace=F)
> anova(hr.step)
Analysis of Deviance Table

Model: binomial, link: logit

Response: chd

Terms added sequentially (first to last)


         Df Deviance Resid. Df Resid. Dev
NULL                       461     596.11
tobacco   1    41.46       460     554.65
ldl       1    23.13       459     531.52
famhist   1    24.28       458     507.24
age       1    21.80       457     485.44
> coef(hr.step)
   (Intercept)        tobacco            ldl famhistPresent            age
   -4.20381991     0.08069792     0.16757435     0.92406001     0.04403574
```

Logistic regression (§4.4)

```
> summary(hr.step)

Call:
glm(formula = chd ~ tobacco + ldl + famhist + age, family = binomial,
    data = hr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7559  -0.8632  -0.4546   0.9457   2.4903

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.203820   0.494517  -8.501  < 2e-16 ***
tobacco         0.080698   0.025484   3.167  0.00154 **
ldl             0.167574   0.054092   3.098  0.00195 **
famhistPresent  0.924060   0.222661   4.150 3.32e-05 ***
age             0.044036   0.009696   4.542 5.58e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 485.44  on 457  degrees of freedom
AIC: 495.44

Number of Fisher Scoring iterations: 3
```

Interpretation of coefficients

- e.g. tobacco (measured in kg): coeff= 0.081
- $\mathrm{logit}\{p_i(\beta)\} = \beta^T x_i$; increase in one unit of x_{ij}, say, leads to increase in $\mathrm{logit} p_i$ of 0.081; increase in $p_i/(1-p_i)$ of $\exp(0.081) = 1.084$.
- estimated s.e. 0.026, $\mathrm{logit} p_i \pm 0.026$, $\exp(0.081 + 2 \times .026), \exp(0.081 - 2 \times .026))$ is $(1.03, 1.14)$.
- similarly for age:$\hat{\beta}_j = 0.044$; increased odds 1.045 for 1 year increase
- prediction of new values to class 1 or 0 according as $\hat{p} > (<) 0.5$

generalize to K classes

1. multivariate logistic regression:

$$\log \frac{pr(y = 1 \mid x)}{pr(y = K \mid x)}, \ldots, \log \frac{pr(y = K - 1 \mid x)}{pr(y = K \mid x)}$$

2. impose an ordering (polytomous regression: MASS p. ??? and `polr`)
3. multinomial distribution (related to neural networks: MASS p. ??? and ??)

- $y \in \{1, 2, \ldots, K\}$,
- $f_c(x) = f(x \mid y = c) =$density of x in class c
- Bayes Theorem:

$$\mathrm{pr}(y = c \mid x) = \frac{f(x \mid y = c)\pi_c}{f(x)} \quad c = 1, \ldots, K$$

- associated classification rule: assign a new observation to class c if

$$p(y = c \mid x) > p(y = k \mid x) \quad k \neq c$$

(maximize the posterior probability)

$$x \mid y = k \sim N_p(\mu_k, \Sigma_k)$$

$$\begin{aligned} & p(y = k \mid x) \\ \propto \quad & \pi_k \frac{1}{(\sqrt{2\pi})^p |\Sigma_k|^{1/2}} \exp -\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) \end{aligned}$$

which is maximized by maximizing the log:

$$\max_k \{\log \pi_k - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\}$$

– if we further assume $\Sigma_k = \Sigma$, then

$$\max_k \{\log \pi_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k)\}$$

$$\Leftrightarrow \max_k \{\log \pi_k - \frac{1}{2}(x^T \Sigma^{-1} x - x^T \Sigma^{-1} \mu_k - \mu_k \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k)\}$$

$$\Leftrightarrow \max_k \{\log \pi_k + x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k\}$$

Discriminant analysis (§4.3)

- Procedure: compute
  $\delta_k(x) = \log \pi_k + x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k$
- classify observation x to class c if $\delta_c(x)$ largest (see Figure 4.5, left)
- Estimate unknown parameters π_k, μ_k, Σ:

$$\hat{\pi}_k = \frac{N_k}{N} \quad \hat{\mu}_k = \sum_{y_i = k} \frac{x_i}{N_c}$$

$$\widehat{\Sigma} = \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$$

(see Figure 4.5,right)

- Special case: 2 classes
- Choose class 2 if $\log \hat{\pi}_2 + x^T \hat{\Sigma}^{-1} \hat{\mu}_2 - \frac{1}{2} \hat{\mu}_2^T \hat{\Sigma}^{-1} \hat{\mu}_2 > \log \hat{\pi}_1 + x^T \hat{\Sigma}^{-1} \hat{\mu}_1 - \frac{1}{2} \hat{\mu}_1^T \hat{\Sigma}^{-1} \hat{\mu}_1$,
- $\Leftrightarrow x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} \hat{\mu}_2 \hat{\Sigma}^{-1} \hat{\mu}_2 - \frac{1}{2} \hat{\mu}_1 \hat{\Sigma}^{-1} \hat{\mu}_1 + \log(N_1/N) - \log(N_2/N)$
- Note it is often common to specify $\pi_k = 1/K$ in advance rather than estimating from the data
- If Σ_k not all equal, the discriminant function $\delta_k(x)$ defines a quadratic boundary; see Figure 4.6, left
- An alternative is to augment the original set of features with quadratic terms and use linear discriminant functions; see Figure 4.6, right

Another description of LDA (§4.3.2, 4.3.3):

- Let $W =$ within class covariance matrix ($\hat{\Sigma}$)
- $B =$ between class covariance matrix
- Find $a^T X$ such that $a^T Ba$ is maximized and $a^T Wa$ minimized, i.e.

$$\max_a \frac{a^T Ba}{a^T Wa}$$

- equivalently

$$\max_a a^T Ba \text{ subject to } a^T Wa = 1$$

- Solution a_1, say, is the eigenvector of $W^{-1}B$ corresponding to the largest eigenvalue. This determines a line in R^p.
- continue, finding a_2, orthogonal (with respect to W) to a_1, which is the eigenvector corresponding to the second largest eigenvalue, and so on.

- There are at most $\min(p, K-1)$ positive eigenvalues.
- These eigenvectors are the linear discriminants, also called canonical variates.
- This technique can be useful for visualization of the groups.
- Figure 4.11 shows the 1st two canonical variables for a data set with 10 classes.
- (§4.3.3) write $\hat{\Sigma} = UDU^T$, where $U^T U = I$, D is diagonal (see p.87 for $\hat{\Sigma}$)
- $X^* = D^{-1/2} U^T X$, with $\hat{\Sigma}^* = I$
- classification rule is to choose k if $\hat{\mu}_k^*$ is closest (closest class centroid)
- only needs the K points $\hat{\mu}_k^*$, and the $K-1$ dimension subspace to compute this, since remaining directions are orthogonal (in the X^* space)
- if $K = 3$ can plot the first two variates (cf wine data)
- See p.92, Figures 4.4 and 4.8 (algorithm on p.92 finds the best in order, as described on the previous slide)

Notes

- §4.2 considers *linear* regression of $0, 1$ variable on several inputs (odd from a statistical point of view)
- how to choose between logistic regression and discriminant analysis?
- they give the same classification error on the heart data (is this a coincidence?)
- logistic regression and generalizations to K classes doesn't assume any distribution for the inputs
- discriminant analysis more efficient if the assumed distribution is correct
- warning: in §4.3 x and x_i are $p \times 1$ vectors, and we estimate β_0 and β, the latter a $p \times 1$ vector
- in §4.4 they are $(p+1) \times 1$ with first element equal to 1 and β is $(p+1) \times 1$.

R code for the wine data

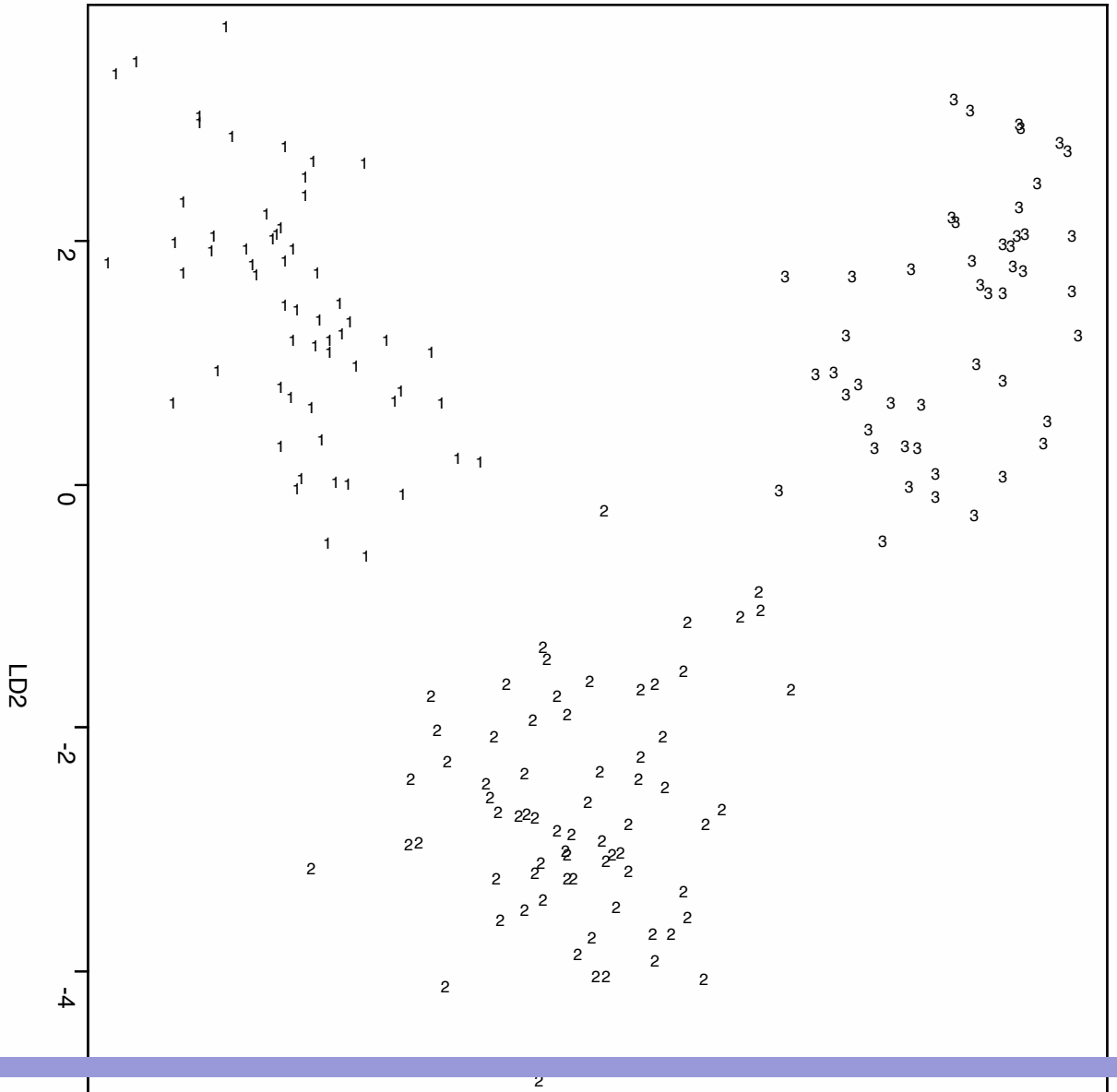
```
> library(MASS)
> wine.lda <- lda(class ~ alcohol + malic + ash + alcil + mag + totphen +
    flav + nonflav + proanth + col + hue + dil + proline, data = wine)
> wine.lda
Call:
lda.formula(class ~ alcohol + malic + ash + alcil + mag + totphen +
    flav + nonflav + proanth + col + hue + dil + proline, data = wine)

Prior probabilities of groups:
        1         2         3
0.3314607 0.3988764 0.2696629

Group means:
   alcohol    malic      ash    alcil      mag  totphen      flav  nonflav
1 13.74475 2.010678 2.455593 17.03729 106.3390 2.840169 2.9823729 0.290000
2 12.27873 1.932676 2.244789 20.23803  94.5493 2.258873 2.0808451 0.363662
3 13.15375 3.333750 2.437083 21.41667  99.3125 1.678750 0.7814583 0.447500
   proanth      col       hue      dil   proline
1 1.899322 5.528305 1.0620339 3.157797 1115.7119
2 1.630282 3.086620 1.0562817 2.785352  519.5070
3 1.153542 7.396250 0.6827083 1.683542  629.8958

Coefficients of linear discriminants:
                  LD1           LD2
alcohol -0.403399781  0.8717930699
malic    0.165254596  0.3053797325
ash     -0.369075256  2.3458497486
alcil    0.154797889 -0.1463807654
mag     -0.002163496 -0.0004627565
totphen  0.618052068 -0.0322128171
flav    -1.661191235 -0.4919980543
nonflav -1.495818440 -1.6309537953
proanth  0.134092628 -0.3070875776
col      0.355055710  0.2532306865
hue     -0.818036073 -1.5156344987
```

R code for the wine data



Separating hyperplanes (§4.5)

- assume two classes only; change notation so that $y = \pm 1$
- use *linear* combinations of inputs to predict y

$$y = \begin{cases} -1 & \text{as } \beta_0 + x^T\beta < 0 \\ +1 & \phantom{\text{as }} \beta_0 + x^T\beta > 0 \end{cases}$$

- misclassification error $D(\beta, \beta_0) = -\Sigma_{i \in \mathcal{M}} y_i(\beta_0 + x_i^T\beta)$ where
- $\mathcal{M} = \{j; y_j(\beta_0 + x_j^T\beta) < 0\}$
- note that $D(\beta) > 0$ and proportional to the 'size' of $\beta_0 + x_i^T\beta$
- Can show that an algorithm to minimize $D(\beta, \beta_0)$ exists and converges the the plane that separates $y = +1$ from $y = -1$ if such a plane exists
- But it will cycle if no such plane exists and be very slow if the 'gap' is small

- Also if one plane exists there is likely many (Figure 4.13)
- The plane that defines the ”largest” gap is defined to be ”best”
- can show that this needs to

$$\min_{\beta_0,\beta} \frac{1}{2}||\beta||^2$$
$$\text{s.t.} \qquad y_i(\beta_0 + x_i^T\beta) \geq 1, \quad i = 1, \ldots N \qquad (4.44)$$

- See Figure 4.15
- the points on the edges (margin) of the gap called support points or support vectors; there are typically many fewer of these than original points
- this is the basis for the development of Support Vector Machines (SVM), more later
- sometimes add features by using basis expansions; to be discussed first in the context of regression (Chapter 5)