Linear discriminant analysis

Chapter 12: Discriminant Analysis and Other Linear Classification Models

Why discriminant analysis?

- When the classes are *well-separated*, the parameter estimates for the logistic regression model are surprisingly $\boxed{\text{unstable}}$. Linear discriminant analysis does not suffer from this problem.

data preprocessing

- If *n* is small and the distribution of the predictors is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.

$$y: \begin{cases} 0 \\ 1 \\ 2 \end{cases}$$

Bayes' theorem for classification

k=2, binary case

- Suppose Y can take on K possible distinct values, denoted by C = {1, 2, . . . , K}. Bayes' theorem states that

Given

prior probability

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

$$= \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Bayes' rule

where π_k = Pr(Y = k) is the overall or prior probability of coming from class k. f_k (x) = Pr(X = x | Y = k) is the density for X given that X = x is from class k.

If $Pr(Y=i \mid X=x) > Pr(Y=j \mid X=x)$, we classify Y into i

$$\frac{\pi_i f_i(x)}{\cancel{\sum_{l=1}^{k} \pi_l f_l(x)}} > \frac{\pi_j f_j(x)}{\cancel{\sum_{l=1}^{k} \pi_l f_l(x)}}$$

$$\frac{f_i(x)}{f_j(x)} > \frac{\pi_j}{\pi_i}$$

(In the absence of prior information for Y to be i or j, we often let $\pi_j = \pi_i$)

$$\frac{f_i(x)}{f_j(x)} > 1 \Rightarrow \text{We classify } y \text{ into } i$$

We just need to specify $f_i(x)$ and $f_j(x)$

A two-group classification problem

classify y into 1

- For a two-group classification problem, the rule that minimizes the total probability of misclassification would be to classify X into group 1 if $\pi_1 f_1(x) > \pi_2 f_2(x)$ and into group 2 if the inequality is reversed.
- We assume that $f_k(x)$ is normal (or Gaussian). The Gaussian density has the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right\}$$

where μ_k and σ^2_k are the mean and variance for the class k. For now, we assume that all the $\sigma^2_k = \sigma^2$ are the same.

$\Downarrow$ $\sigma_i = \sigma_j$ for $\forall$ i & j

$$\frac{f_i(x)}{f_j(x)} = \frac{\cancel{\frac{1}{\sqrt{2\pi}\sigma_i}} \exp\left\{-\frac{1}{2\sigma_i^2}(x-\mu_i)^2\right\}}{\cancel{\frac{1}{\sqrt{2\pi}\sigma_j}} \exp\left\{-\frac{1}{2\sigma_j^2}(x-\mu_j)^2\right\}} > 1 \qquad (\sigma_i^2 = \sigma_j^2 = \sigma^2)$$

$$\exp\left\{-\frac{1}{2\sigma^2}\left[(x-\mu_i)^2 - (x-\mu_j)^2\right]\right\} > 1$$

$$-\frac{1}{2\sigma^2}\left[(x-\mu_i)^2 - (x-\mu_j)^2\right] > 0$$

$$(x-\mu_i)^2 - (x-\mu_j)^2 < 0$$

$$\cancel{x^2} - \underline{\underline{2x\mu_i}} + \underline{\mu_i^2} - \cancel{x^2} + \underline{\underline{2x\mu_j}} - \underline{\mu_j^2} < 0$$

$$(\mu_i - \mu_j)(\mu_i + \mu_j) - 2x(\mu_i - \mu_j) < 0$$

$$(\mu_i - \mu_j)(\mu_i + \mu_j - 2x) < 0$$

$$(\mu_i - \mu_j)\left(\frac{\mu_i + \mu_j}{2} - x\right) < 0$$

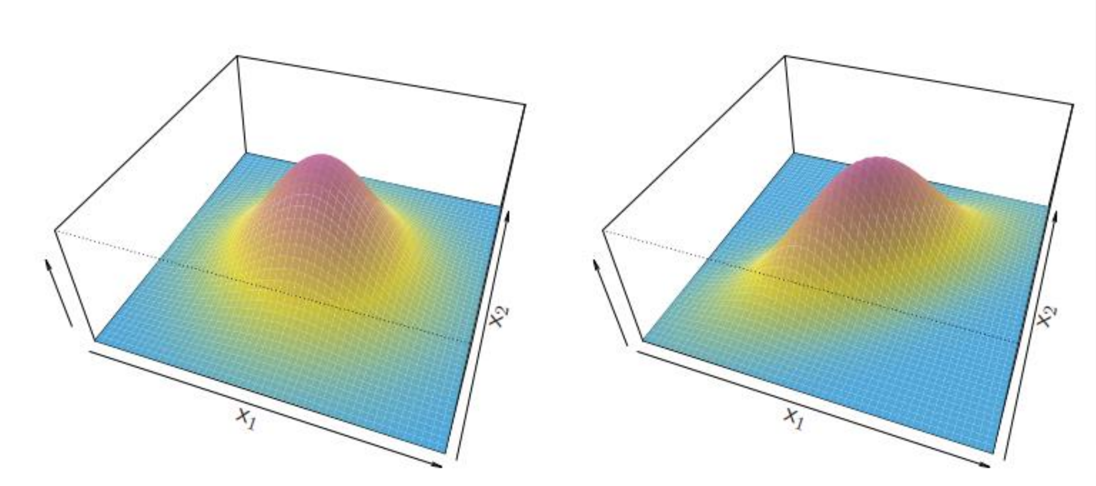
- If $\mu_i - \mu_j > 0$, then $\frac{\mu_i + \mu_j}{2} - x < 0 \Rightarrow \frac{\mu_i + \mu_j}{2} < x$
- If $\mu_i - \mu_j > 0$, then $\frac{\mu_i + \mu_j}{2} - x > 0 \Rightarrow \frac{\mu_i + \mu_j}{2} > x$

Linear discriminant analysis for p > 1

- We model X using multivariate Gaussian

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right\}$$

where μ_k is the mean of X in class k, and $\Sigma = \mathrm{Cov}(X)$ is its covariance matrix.



LDA

```
set.seed(476)
ldaTune <- train(x = as.matrix(Smarket.train[,1:8]),
y = Smarket.train$Direction,
method = "lda",
preProc = c('center', 'scale'),
metric = "ROC",
trControl = ctrl)
ldaTune

### Save the test set results in a data frame
testResults$LDA <- predict(ldaTune, Smarket.test)
```

LDA output

```
> ldaTune
Linear Discriminant Analysis

998 samples
  8 predictor
  2 classes: 'Down', 'Up'

Pre-processing: centered (8), scaled (8)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 750, 750, 750, 750, 750, 750, ...
Resampling results:

  ROC        Sens       Spec
  0.9962972  0.9367213  0.9857143
```

Partial least squares discriminant analysis

Chapter 12: Discriminant Analysis and Other Linear Classification Models

If the predictors are highly correlated, LDA performs worse. To deal with this issue, we consider PLSda

Partial least squares discriminant analysis

- PLSDA can be performed using the *plsr* function within the *pls* package by using a categorical matrix which defines the response categories.
- The caret package contains a function (*plsda*) that can create the appropriate dummy variable PLS model for the data and then post-process the raw model predictions to return class probabilities.
- The syntax is very similar to the regression model code for PLS that we discussed in Chapter 6.

Partial least squares discriminant analysis

```
set.seed(476)
plsdaTune <- train(x = Smarket.train[,1:8],
y = Smarket.train$Direction,
method = "pls",
tuneGrid = expand.grid(.ncomp = 1:5),
trControl = ctrl)

### Save the test set results in a data frame
testResults$plsda <- predict(plsdaTune, Smarket.test)
```

p=8

tunig paramter for the number of components

PLSDA output

```
>  plsdaTune
Partial Least Squares

998 samples
  8 predictor
  2 classes: 'Down', 'Up'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 750, 750, 750, 750, 750, 750, ...
Resampling results across tuning parameters:

  ncomp  ROC        Sens       Spec
  1      0.9939448  0.9245902  0.9720635
  2      0.9968488  0.9331148  0.9869841
  3      0.9974525  0.9409836  0.9885714
  4      0.9966693  0.9413115  0.9860317
  5      0.9963128  0.9367213  0.9853968

ROC was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 3.
```