Chapter 4

Classification

4.1 Motivation and Overview

- When we predict a categorical or qualitative variable Y this is called classification analysis. The standard classifiers considered in this chapter are
  - logistic regression
  - linear discriminant analysis
  - quadratic discriminant analysis
  - K-nearest neighbors

Example 1. *The following simulated data set contains information on ten thousand customers. The aim here is to predict which customers will default on their credit card debt. Describe the distribution of the default variable and its relationship to some other variables.*

Code

```
> library(ISLR)
> help(Default)
> plot(Default, pch = ".")
> dim(Default)
[1] 10000     4
> Default[1:3,]
  default student   balance   income
1      No      No  729.5265 44361.63
2      No     Yes  817.1804 12106.13
3      No      No 1073.5492 31767.14
> summary(Default)
 default    student       balance           income
 No :9667   No :7056   Min.   :   0.0   Min.   :  772
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
                       Median : 823.6   Median :34553
                       Mean   : 835.4   Mean   :33517
                       3rd Qu.:1166.3   3rd Qu.:43808
                       Max.   :2654.3   Max.   :73554
```

4.2 Logistic Regression

Generalized linear models assume $g(E(Y|X)) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$ where $g(\cdot)$ is some monotone function and Y is a member of the exponential family (Binomial, Poisson, Normal, Exponential, ...)

4.2.1 The Logistic Model

Definition 1. *If $Y \sim Bernoulli(p(X))$, where $p(X) = \Pr(Y = 1|X)$, the simple logistic regression model assumes that the log odds, also called the logit, is linear in the predictor X. Specifically,*

- $logit(p(X)) = \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$
- *This gives* $\frac{p(X)}{1-p(X)} = \exp\{\beta_0 + \beta_1 X\}$ *or*
- $p(X) = \frac{\exp\{\beta_0+\beta_1 X\}}{1+\exp\{\beta_0+\beta_1 X\}}$

Interpretations:

- Sketch a few curves of $p(X)$ and determine how β_0 and β_1 describe the nature of the relationship between X and Y.

```
> logit<-function(b0=0,b1=1, xlim =NULL){
+ curve(exp(b0 + b1*x)/(1+exp(b0 + b1*x)), ylab = "p(x)",
+  add = TRUE, xlim=xlim, lwd=3)}
>
> logit(b0=0,b1=1, xlim = c(-10,10))
Warning message:
In curve(exp(b0 + b1 * x)/(1 + exp(b0 + b1 * x)), ylab = "p(x)",  :
  'add' will be ignored as there is no existing plot
> logit(b0=5, b1=1)
> logit(b0 = -5, b1=1)
> logit(b0=0, b1=-1)
> logit(b0=0, b1= -.5)
```

```
> logit(b0=0, b1 = -2)
> logit(b0=0, b1 = -.2)
```

- How can you interpret β_1, e^{β_1} and the event that $logit(p(X)) > 0$?

- Why is the solution to $0 = logit(p(X)) = \beta_0 + \beta_1 X$ in terms of X, say $X = -\beta_0/\beta_1$ important? Verify that $p(X) = 1 - p(X)$ when $X = -\beta_0/\beta_1$

4.2.2 Maximum Likelihood Estimation

- Note that if $\Pr(Y_i = 1) = p$ then we can write the probability mass function for y_i by $f(y_i|p) = \Pr(Y_i = y_i|p) = p^{y_i}[1-p]^{1-y_i}$.

- For $y_1, y_2, ..., y_n$ independent Bernoulli(p), the joint probability mass function is
$$f(\mathbf{y}|p) = \prod_i f(y_i|p) = \prod_i p^{y_i}[1-p]^{1-y_i}.$$

- The likelihood of p is just the probability mass function $l(p) = \prod_i p^{y_i}[1-p]^{1-y_i}$, but is called the likelihood to emphasize our viewpoint that $\mathbf{y}$ is fixed/observed and p is the argument varying in the function. Find $l(p)$ for $y_1 = y_2 = 1$ and $y_3 = 0$, plot $l(p)$ vs p, and identify the maximum of $l(p)$.

- We can also maximize the log of the likelihood $\log l(p)$, and more generally verify that $\log l(p)$ is maximized when $p = \bar{y}$ using calculus arguments (Stat students should verify this).

- For the simple logistic regression model, we plug in $p(x_i)$ in for p and the likelihood is written $l(\beta_0, \beta_1) = \prod_i p(x_i)^{y_i}[1 - p(x_i)]^{1-y_i}$. Can you find closed form expression for the maximums here?

- MLE's are values of parameters that maximize the likelihood. They and their standard errors are based on first and second derivatives of the log likelihood. Large sample results let us use usual formula for confidence intervals and hypothesis tests. Maximization is done numerically using software.

4.2.3 Examples

- Identify the model fit below, maximum likelihood estimates, $p(X)$ and interpret the p-value and the nature of the relationship between the predictor and the response, and the number 1936.987.

Code

```
> attach(Default)
> Inc.model<-glm(default~balance, family = "binomial")
> summary(Inc.model)$coef
                 Estimate   Std. Error   z value      Pr(>|z|)
(Intercept) -10.651330614 0.3611573721 -29.49221 3.623124e-191
balance       0.005498917 0.0002203702  24.95309 1.976602e-137
> co<-coef(Inc.model)
> plot(balance, I(default == "Yes"))
> curve(exp(co[1] + co[2]*x)/(1 + exp(co[1] + co[2]*x)), add = TRUE, lwd =3)
> abline(h = 0.5, v = -co[1]/co[2])
> -co[1]/co[2]
(Intercept)
   1936.987
```

- Interpret the probabilities below and verify that $logit(p) < 0 \iff p < 1/2 \iff X < 1936.987$ for the X values below.

Code

```
> exp(co[1] + co[2]*1000)/(1 + exp(co[1] + co[2]*1000))
(Intercept)
0.005752145
> predict(Inc.model,  newdata = data.frame(balance = c(1000,2000)), type = "response")
          1           2
0.005752145 0.585769370
```

```
> predict(Inc.model, newdata = data.frame(balance = c(1000,2000)))
         1          2
-5.1524137  0.3465032
> exp(-5.1524)/(1 + exp(-5.1524))
[1] 0.005752223
```

- Below we fit a logistic regression model for predicting default with student status. Write out the model and interpret the p-value and $e^{\hat{\beta}_1}$.

Code

```
> student.model<-glm(default~student, family = "binomial", data = Default)
> summary(student.model)$coef
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
studentYes   0.40489    0.11502    3.52 0.000431 ***
> exp(.40489)
[1] 1.499138
```

- Interpret the probabilities below and identify logit(p) and p

Code

```
> predict(student.model,
+  newdata = data.frame(student = c("Yes","No")), type = "response")
         1          2
0.04313859 0.02919501
> predict(student.model,
+  newdata = data.frame(student = c("Yes","No")))
        1         2
-3.099241 -3.504128
> exp(-3.099241)/(1+exp(-3.099241))
[1] 0.04313857
> exp(-3.50412)/(1 + exp(-3.50412))
[1] 0.02919523
```

4.2.4 Multiple Logistic Regression

- For X_1, ..., X_p predictors we have

$$logit(p(X)) = \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

  where note that X_j could represent an interaction term, indicator variable, or continuous variable.

- Write out the model below and identify each curve on the plot.

Code

```
> bal.stud.model<-glm(default~ balance + student,
+  family = "binomial",data = Default)
> summary(bal.stud.model)$coef
                 Estimate  Std. Error    z value      Pr(>|z|)
(Intercept) -10.749495878 0.369191361 -29.116326 2.230782e-186
balance       0.005738104 0.000231847  24.749526 3.136911e-135
studentYes   -0.714877620 0.147519010  -4.846003  1.259734e-06
> plot(balance, I(default == "Yes"), col = student)
> co<-coef(bal.stud.model)
> # recall the logit function for plotting
> args(logit)
function (b0 = 0, b1 = 1, xlim = NULL)
NULL
> logit(b0 = co[1], b1=co[2])
> logit(b0 =co[1] + co[3], b1 = co[2])
```

- Observe that now the coefficient for student status is negative. This is because balance and student status are dependent. Do you find this confounding (pun intended)?

4.2.5 Classification with Logistic Regression

- Basic rule: Classify Y into group 1 (ie. $\hat{Y} = 1$) using X if $p(X) = \Pr(Y = 1|X) > 1/2$ or if $logit(p(X)) > 0$.

- More general rule: Classify Y into group 1 using X if $p(X) > c$ or if $logit(p(X)) > logit(c)$.

- Write out the classification rule for the model below.

Code

```
> coef(bal.stud.model)
  (Intercept)       balance    studentYes
-10.749495878   0.005738104  -0.714877620
> boxplot(predict(bal.stud.model)~default)
> abline(h = log(.5/(1-.5)), lwd =3)
> abline(h = log(.2/.8), lty=2, col =2, lwd =3)
> log(.2/.8)
[1] -1.386294
```

4.3 Error Rates for Classification Rules

4.3.1 Training Error Rate

- Compute the training error rate $\frac{1}{n}\sum_i (y_i \neq \hat{y}_i)$ for the balance student model above.

Code

```
> predictions<-rep("No",10000)
> predictions[predict(bal.stud.model)>0]<-"Yes"
> data.frame(predict(bal.stud.model), predictions, default)[c(1:3,9952),]
     predict.bal.stud.model. predictions default
1                  -6.563397          No      No
2                  -6.775307          No      No
3                  -4.589359          No      No
9952               -2.052789          No     Yes
> table(predictions, default)
           default
predictions   No  Yes
        No  9628  228
        Yes   39  105
> round(prop.table(table(predictions, default), 2),3)
           default
predictions    No   Yes
        No  0.996 0.685
        Yes 0.004 0.315
> mean(predictions!=default)
[1] 0.0267
```

- How would the error rate compare with a rule that just automatically predicted that a person wouldn't default?

```
> predictions<-rep("No",10000)
> mean(predictions!=default)
[1] 0.0333
```

- Should we use a different cutoff for classification than $logit(p(x)) > 0$ above?

4.3.2 Confusion Matrix: False Positives and True Positives

Definition 2. *The sensitivity, specificity, false positive rate, false discovery proportion, false discovery rate, and many other terms are used to summarize the performance of a classifier. The receiver operating characteristic (ROC) curve, plots the false positive rate (FPR = FP/N) on the x axis and the true positive rate (TPR = TP/P) on the Y axis.*

		TRUTH		
		-	+	
CLASSIFICATION	-	TN	FN	N*
	+	FP	TP	P*
		N	P	

- Assuming that a default is a "positive", what are the TPR and FPR for the rules below.

Code

```
> predictions<-rep("No",10000)
> predictions[predict(bal.stud.model)>log(.2/(.8))]<-"Yes"
> table(predictions, default)
           default
predictions   No  Yes
        No  9391  130
        Yes  276  203
> round(prop.table(table(predictions,default),2),2)
           default
predictions   No  Yes
        No  0.97 0.39
        Yes 0.03 0.61
> predictions<-rep("No",10000)
> predictions[predict(bal.stud.model)>log(.05/(.95))]<-"Yes"
```

```
> table(predictions, default)
           default
predictions   No  Yes
        No  8704   51
        Yes  963  282
> round(prop.table(table(predictions,default),2),2)
           default
predictions   No  Yes
        No  0.90 0.15
        Yes 0.10 0.85
```

- Describe how you would construct the ROC curve and determine what the area under the curve tells you.

- Critical thinking: Suppose you are the bank and will issue a credit card to an individual if the are predicted to not default based on the last classification rule. The row percentages from the rule above are below. Is the rule profitable if the bank earns on average \$100 credit cards issued that don't default but looses \$5,000 on average per default?

```
> round(prop.table(table(predictions,default),1),2)
           default
predictions   No  Yes
        No  0.99 0.01
        Yes 0.77 0.23
```

- Why might you not want to base rules on training error rates? Will error rates for these rules, when applied to test data, tend to be higher or lower?

4.3.3 Estimating Test Error Rates

- Cross validation can be used to get a better estimate of the test error rate.

Code

```
> set.seed(1)
> train.index<-sample(1:10000, 9000)
> train.index[1:5]
[1] 2656 3721 5728 9080 2017
> #detach(Default)
> Default.test<-Default[-train.index,]
> dim(Default.test)
[1] 1000    4
> Default.train<-Default[train.index,]
> dim(Default.train)
[1] 9000    4
>
> model<-glm(default~balance + student, family = "binomial", data = Default.train)
> predictions<-rep("No", 1000)
> predictions[predict(model, newdata = Default.test)>0]<-"Yes"
> table(predictions,Default.test$default)

predictions  No Yes
        No  966  19
        Yes   5  10
> mean(predictions!=Default.test$default)
[1] 0.024
```

- Are the error rates similar? Would you expect them to be similar for this simple model and large sample size?

- Is the rule which gives a credit card to an individual if the probability of default

is less 0.05 still profitable if the bank still earns $100 on non-defaults and looses $5000 on defaults? How could you adjust the rule in practice?

Code

```
> predictions<-rep("No", 1000)
> predictions[predict(model, newdata = Default.test)>log(.05/.95)]<-"Yes"
> table(predictions,Default.test$default)

predictions  No Yes
        No  866   4
        Yes 105  25
> round(prop.table(table(predictions,Default.test$default),1),3)

predictions    No   Yes
        No  0.995 0.005
        Yes 0.808 0.192
```

4.4 Bayes' Rule

Recall for a partition $A_1, A_2, ..., A_K$ and event B Bayes rule gives

$$\Pr(A_k|B) = \frac{\Pr(B|A_k)\Pr(A_k)}{\Pr(B|A_1)\Pr(A_1) + ... + \Pr(B|A_K)\Pr(A_K)}$$

- $\Pr(A_k)$ called a prior probability.
- $\Pr(A_k|B)$ is called a posterior probability.

Example 2. *Suppose that 5% of the population has a disease. The probability that a person tests positive given they have the disease is 0.9 and the probability that they test positive given they do not have the disease is 0.1. Use Bayes rule to compute the probability that a person has the disease given that they test positive.*

4.5 Linear Discriminant Analysis

4.5.1 Univariate LDA classifier

- Model for Univariate LDA: Suppose that Y is a categorical random variable such that $\Pr(Y = k) = \pi_k$, for $k = 1, 2, ..., K$, and that $X|Y = k$ has normal density with mean μ_k and variance σ^2.

- Write out and plot $f(x) = \pi_1 f_1(x) + \pi_2 f_2(x)$ for some different μ_k, σ and π_k values.

- Use Bayes' rule to compute the posterior probability $p_k(x) = \Pr(Y = k|X = x)$.

- The Bayes classifier classifies x into group k if $p_k(x)$ is the largest. Observe $p_k(x)$ is maximized if $\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$ is maximized, so the classifier is linear in x.

  - Can you verify that $p_k(x) = \dfrac{\pi_k \exp\left\{x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2}\right\}}{\sum_j \pi_j \exp\left\{x\frac{\mu_j}{\sigma^2} - \frac{\mu_j^2}{2\sigma^2}\right\}}$?

  - Why in finding the largest $p_1(x), ..., p_K(x)$ do you only need to compute $\delta_1(x), ..., \delta_K(x)$?

  - What happens to the classifier if we increase prior π_k?

- For $K = 2$, what is the value of x such that the linear discriminant function $L(x) \equiv \delta_2(x) - \delta_1(x) = 0$ when $\pi_2 = \pi_1$? This is called a Bayes Decision boundary

- We can write $L(x) = \delta_2(x) - \delta_1(x) = \beta_0 + \beta_1 x$ hence the terminology *linear discriminant analysis*. What are β_0, β_1?

- In practice we have data x_{ik} the i'th unit from the kth class, for $i = 1, 2, ..., n_k$ and $k = 1, 2, ..., K$ and we can plug in estimates of parameters $\hat{\mu}_k$, $\hat{\sigma}^2$, and *possibly* $\hat{\pi}_k$. The resulting method is called a linear discriminant analysis (LDA). In general, plugging in estimates of priors in Bayes methods is called Empirical Bayes.

- Example: First determine if the balance is different across populations under the normal model using the training data. We might also use the whole data set in this step or skip this step.

Code

```
> summary(lm(balance~default, data = Default.train))

Call:
lm(formula = balance ~ default, data = Default.train)

Residuals:
     Min       1Q   Median       3Q      Max
-1092.34  -333.92    -0.95   319.41  1589.18

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  801.826      4.858  165.04   <2e-16 ***
defaultYes   942.909     26.434   35.67   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 453 on 8998 degrees of freedom
Multiple R-squared:  0.1239,    Adjusted R-squared:  0.1238
F-statistic:  1272 on 1 and 8998 DF,  p-value: < 2.2e-16


> t.test(balance~default, var.equal = T, data = Default.train)

        Two Sample t-test

data:  balance by default
t = -35.67, df = 8998, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -994.7267 -891.0920
sample estimates:
 mean in group No mean in group Yes
         801.8258         1744.7351
```

- Identify parameter estimates, the estimated decision boundary, and estimated coefficient in $L(x)$ and write out the estimated formula for $p_2(x) = \Pr(Default|x)$ (note that $\hat{\sigma}^2 = 453^2$ above)

Code

```
> # Univariate LDA
> library(MASS)
> lda.fit<-lda(default~balance, data = Default.train)
> lda.fit
Call:
lda(default ~ balance, data = Default.train)

Prior probabilities of groups:
        No        Yes
0.96622222 0.03377778

Group means:
      balance
No   801.8258
Yes 1744.7351

Coefficients of linear discriminants:
                LD1
balance 0.002207272
```

- Identify the linear discriminant scores $L(x_i) = \delta_2(x_i) - \delta_1(x_i)$ for each i in the test data, and estimated posterior probabilities $p_k(x_i)$. Note R takes $\pi_1 = \pi_2$ in computing $L(x_i)$ but uses $\hat{\pi}_k$ in computing $p_k(x_i)$ unless otherwise specified. Classifications are based on $p_k(x_i)$.

Code

```
> plot(lda.fit)
> predictions<-predict(lda.fit, newdata = Default.test)
> names(predictions)
[1] "class"     "posterior" "x"
> cbind(predictions$x, predictions$posterior, predictions$class, Default.test$default)
         LD1        No         Yes
10 -1.840148 0.9998993 0.000100735 1 1
55 -1.840148 0.9998993 0.000100735 1 1
56  1.017160 0.9628925 0.037107508 1 1
64  1.428061 0.9168983 0.083101730 1 1
73  1.245963 0.9415813 0.058418745 1 1
> table(predictions$class, Default.test$default)

       No Yes
  No  968  21
  Yes   3   8
> round(prop.table(table(pred=predictions$class, def=Default.test$default),2),3)
     def
pred     No   Yes
  No  0.997 0.724
  Yes 0.003 0.276
```

4.5.2 Multivariate LDA

Multivariate Normal Distribution

- Suppose that $X = (X_1, ..., X_p)^T$ is a random vector with

  $E(X) = \mu =$

  $Cov(X) = \mathbf{\Sigma} = (\sigma_{ij}) =$

- We write $X \sim MVN(\mu, \mathbf{\Sigma})$ to denote a multivariate normal distribution. The MVN density is

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

  where $|\mathbf{\Sigma}|$ is the determinant.

- Properties of MVN distribution

- For data $x_1, x_2, ..., x_n$, unbiased estimates for the mean and covariance matrix are estimated (these are also MLE's except the MLE of $\mathbf{\Sigma}$ divides by n)

  $\hat{\mu} = \bar{x} =$

  $\hat{\mathbf{\Sigma}} = \mathbf{S} =$

- The quantity $\Delta = (x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)$ is called the Mahalanobis distance from x to μ.

- $z = \mathbf{\Sigma}^{-1/2}(x-\mu) \sim MVN(0, \boldsymbol{I})$

- $z^T z = \Delta$ has a χ^2_p distribution.

- The relationship between X_i and X_j is linear, if it exists. If $\sigma_{ij} = 0$ and X has a MVN distribution, then X_i and X_j are independent.

- Sketch some different bivariate normal densities and determine how the covariance matrix influences the Mahalanobis distances.

The LDA classifier

- Suppose that Y is a categorical random variable such that $\Pr(Y = k) = \pi_k$, for $k = 1, 2, ..., K$, and that $X|Y = k \sim MVN(\mu_k, \mathbf{\Sigma})$. Use Bayes' rule to compute the posterior probability $p_k(x) = \Pr(Y = k|X = x)$

- The Bayes classifier classifies x into category k if $p_k(x)$ is larger than $p_l(x)$ for $l \neq k$. We can again ignore the denominator, take log, and maximize

  $\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log(\pi_k)$

- Can you relate $p_k(x)$ to terms of $\delta_k(x)$s and π_ks?

- A Bayes decision boundary between populations 1 and 2 is the solution to the $L(x) \equiv \delta_2(x) - \delta_1(x) = 0$.

- What is the expression for $L(x) = \delta_2(x) - \delta_1(x)$? Is it linear?

- Suppose that $x = (x_1, x_2)^T$ so $p = 2$ and suppose that $f_1(x)$ and $f_2(x)$ have bivariate normal densities with mean vectors $(0, 0)^T$ and $(1, 1)^T$ respectively. Draw the Bayes decision boundary with the two bivariate normal densities superimposed. What does increasing π_1 do to the boundary?

- In practice we have data $x_{ik} = (x_{ik1}, ..., x_{ikp})^T$ the i'th unit from the kth class, for $i = 1, 2, ..., n_k$ and $k = 1, 2, ..., K$ and we can estimate parameters $\hat{\mu}_k$, $\hat{\Sigma}$, and $\hat{\pi}_k$.

- The linear discriminant function, sometimes still denoted $L(x)$, plugs in estimates of parameters in $L(x) = \delta_2(x) - \delta_1(x)$.

- We can first test the null hypothesis $H_0 : \mu_1 = \mu_2$ using Hotelling T^2 statistic $T^2 \propto (\bar{x}_1 - \bar{x}_2)^T \mathbf{S}_p^{-1} (\bar{x}_1 - \bar{x}_2)$, which can be transformed to an F statistic to get a p-value. This step might not be implemented in practice, or could be applied to training data.

Code

```
> summary(manova(cbind(Default$income,Default$balance) ~ Default$default))
                  Df  Pillai approx F num Df den Df    Pr(>F)
Default$default    1 0.12373   705.78      2   9997 < 2.2e-16 ***
Residuals       9998
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Identify parameter estimates below and the linear discriminant function. Interpret the coefficients, and relate the estimated decision boundary to the means of each group.

Code

```
> plot(income, balance, pch = as.numeric(default), col = as.numeric(default))
> lda.fit2<-lda(default~balance + income, data = Default.train)
Call:
lda(default ~ balance + income, data = Default.train)
```

```
Prior probabilities of groups:
        No        Yes
0.96633333 0.03366667

Group means:
     balance   income
No   806.278 33577.54
Yes 1746.123 32091.40

Coefficients of linear discriminants:
                 LD1
balance 2.224156e-03
income  7.764002e-06
> table(pred = predictions2$class, def = Default.test$default)
     def
pred   No Yes
  No  969  22
  Yes   2   7
```

- Notice that income did little to improve the LDA classifier, which is why its coefficient is small, i.e. $L(x)$ isn't affected much by increasing income as compared to balance.

4.6 Quadratic Discriminant Analysis

4.6.1 The QDA classifier

- Model: Suppose that Y is a categorical random variable such that $\Pr(Y = k) = \pi_k$, for $k = 1, 2, ..., K$, and that $X|Y = k \sim MVN(\mu_k, \mathbf{\Sigma}_k)$. What's the difference between this model and the LDA model?

- Use Bayes' rule to compute the posterior probability

  $p_k(x) = \Pr(Y = k|X = x) =$

- The Bayes classifier classifies x into category k if $p_k(x)$ is larger than $p_j(x)$ for $j \neq k$. Why can we ignore the denominator in the classifier?

- Can you verify that maximizing $p_k(x)$ is equivalent to maximizing the quadratic function

  $\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log(|\mathbf{\Sigma_k}|) + \log(\pi_k)$

- Sketch a few different pairs of bivariate normal densities with differen covariances and sketch the decision boundary, i.e. where $\delta_1(x) = \delta_2(x)$.

- Suppose that $K = 3$ and $p = 5$. How many parameters are estimated for an LDA vs. a QDA?

4.6.2 Application

- A QDA is run on the Default data below. Identify parameter estimates and the test error rate.

Code

```
>
> qda.fit<-qda(default~balance + income, data = Default.train)
> qda.fit
Call:
qda(default ~ balance + income, data = Default.train)

Prior probabilities of groups:
        No        Yes
0.96622222 0.03377778

Group means:
      balance   income
No   801.8258 33602.14
Yes 1744.7351 32138.64
> predictions<-predict(qda.fit, newdata = Default.test)
> names(predictions)
[1] "class"     "posterior"
> table(predictions$class, Default.test$default)

       No Yes
  No  968  19
  Yes   3  10
```

4.7 K Nearest Neighbors

4.7.1 Method

- Recall to implement the K-NN method to predict Y with X

  1. Estimate $\Pr(Y = j|X = x) = p_j(x) \approx \frac{1}{K}\sum_{i\in N(x)} I(y_i = j)$

     - $N(x)$ indexes the K nearest neighbors of x as defined by $d(x, x_i) = ||x - x_i|| = \sqrt{(x_1 - x_{i1})^2 + ...(x_p - x_{ip})^2}$

  2. Classify Y as class j if $p_j(x)$ is the largest.

- Considerations

  - For $K \geq 2$ we can have ties.

  - The Euclidean distances aren't standardized, so data should be standardized so that each variable has mean 0 and variance 1. Can you think of some other distances that could be applied?

4.7.2 Example

- Get the estimated test error rate for $k = 3$ and $k = 5$.

Code

```
> library(class)
> train.X<- scale(Default.train[,3:4])
> test.X<-scale(Default.test[,3:4])
> y.train<-Default.train[,1]
> set.seed(1)
>
> predictions<-knn(train.X, test.X,y.train,k=3)
> table(predictions, default = Default.test$default)
           default
predictions  No Yes
        No  961  21
        Yes  10   8
>
> predictions<-knn(train.X, test.X,y.train,k=5)
> table(predictions, default = Default.test$default)
           default
predictions  No Yes
        No  963  20
        Yes   8   9
```

4.8 Comments

- All methods made use of an estimate of $\Pr(Y = j|X) = p_j(X)$.
  - Logistic regression directly assumed Y is a Bernoulli random variable with $p(X) = \Pr(Y = 1|X)$ and got MLE's for regression parameters.

  - LDA and QDA used Bayes' theorem and assumed $X|Y = k$ has a MVN density $f_k(x)$, and plugged in MLE for MVN parameters.

  - KNN estimated $p(x)$ as the proportion of K nearest neighbors from class j.

- In order from simple to complex the methods are

```
LDA ----> logistic regression ----> QDA -------> KNN
```

- Which classifier works best depends on the degree to which the modeling assumptions are satisfied, n, and p and whether interpretability is important.

- LDA vs. logistic regression

  - Both are linear in the predictors and interpretable.

  - LDA assumes predictors are MVN while Logistic regression allows for any kind (including categorical) predictors.

  - MLE's can fail for logistic regression when groups are highly separated, while LDA estimates are simple and stable.

  - For $K > 2$, estimation of parameters for $p_k(x)$ is more computationally complex in logistic regression. There's no real additional issue for $K > 2$ for other methods considered here.

- LDA vs. QDA

  - We can't any longer interpret coefficients for QDA.

  - Both assume MVN distribution for the predictors, but QDA doesn't require equal covariances.

- QDA vs. KNN
  - KNN makes no assumption about the distribution of the predictors, but still requires specification of K.

- Before performing an LDA recall we could test $H_0 : \mu_1 = \mu_2$ with Hotellings T^2 statistic
$$T^2 = \frac{n_1 n_2}{n_1 + n_2}(\bar{x}_1 - \bar{x}_2)^T \mathbf{S}^{-1}(\bar{x}_1 - \bar{x}_2)$$
to determine if the two populations are actually discriminable with an LDA.
  - Other common MANOVA tests are Pillai's, Wilk's, and Roy's. They are all identical when $K = 2$.
  - There are also tests for the equal covariance assumptions as well as multivariate normality.

Code
```
> summary(manova(cbind(Default$income,Default$balance) ~ Default$default))
                  Df  Pillai approx F num Df den Df    Pr(>F)
Default$default    1 0.12373   705.78      2   9997 < 2.2e-16 ***
Residuals       9998
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- LDA, QDA, and KNN all easily extend to the $K > 2$ setting. The only difference is that, for say $K = 3$, we have 2 Bayes decision boundaries and 3 posterior probabilities.

Code
```
> data(iris)
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> lda.fit<-lda(Species~Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
+ data = iris)
> lda.fit
Call:
lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
```

```
    data = iris)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785

Proportion of trace:
   LD1    LD2
0.9912 0.0088
> preds<-predict(lda.fit)
> cbind(preds$posterior, preds$x)[1:5,]
  setosa   versicolor    virginica      LD1        LD2
1      1 3.896358e-22 2.611168e-42 8.061800  0.3004206
2      1 7.217970e-18 5.042143e-37 7.128688 -0.7866604
3      1 1.463849e-19 4.675932e-39 7.489828 -0.2653845
4      1 1.268536e-16 3.566610e-35 6.813201 -0.6706311
5      1 1.637387e-22 1.082605e-42 8.132309  0.5144625
> plot(preds$x, pch = as.numeric(iris$Species))
```