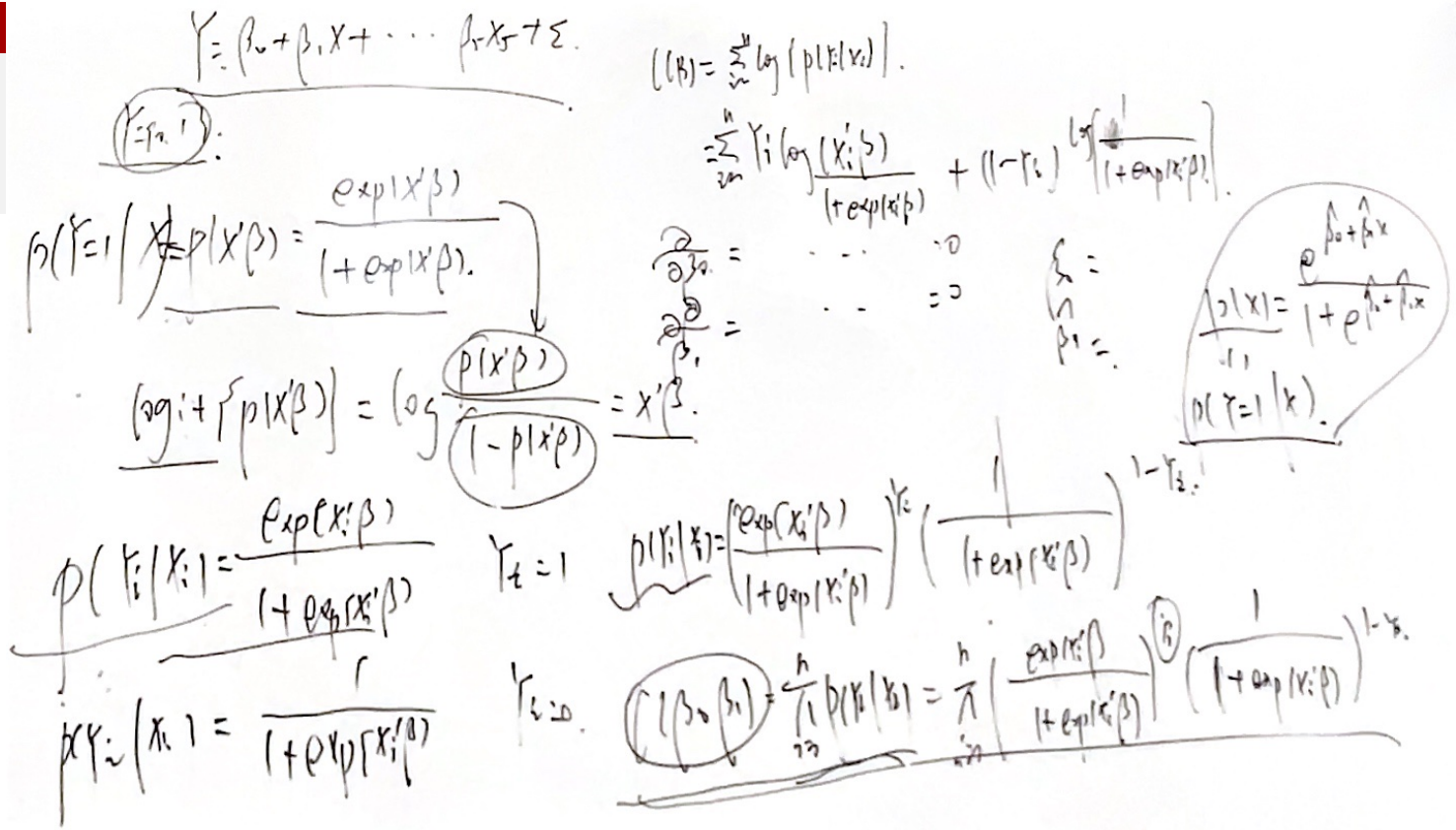
Supervised Learning: Classifications

Zhi LIU

University of Macau

liuzhi@um.edu.mo

Contents



Contents

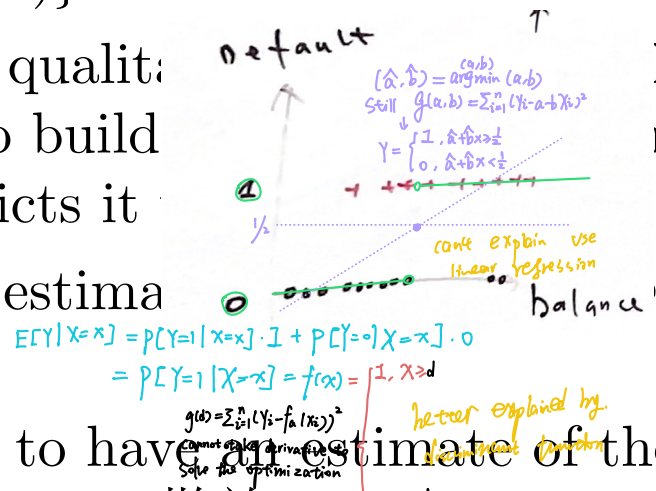
Contents

$Y = f(X) + \varepsilon$

Contents

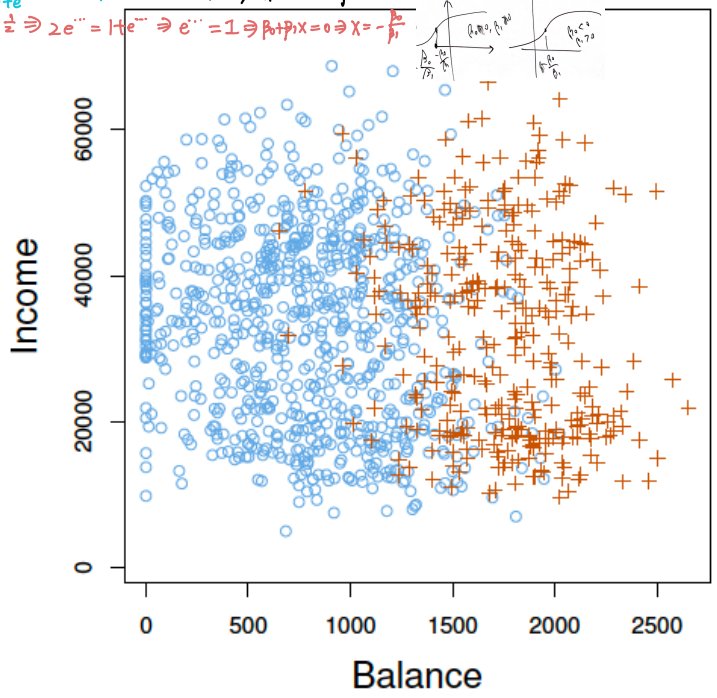
Classifications

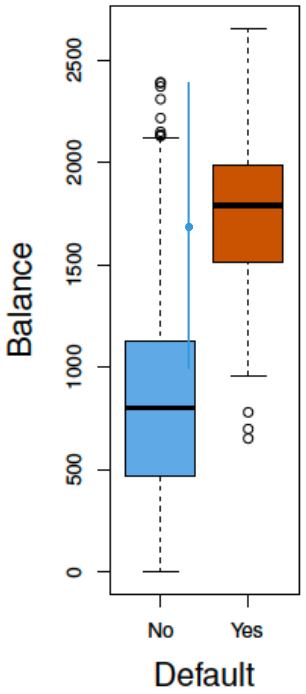
- Qualitative variables values in an unordered set $\mathcal{C}$, such at
  - eye colour *in* {brown, blue, green}
  - email $\in$ {spam, ham (not-spam)}
- Given a feature vector X, and a qualit… king values in set $\mathcal{C}$, the classification task is to build … hat takes as input feature vector X and predicts it …
- Often we are more interested in estima… s that X belongs to each category in $\mathcal{C}$;
- For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent (欺詐), than a classification is fraudulent or not.

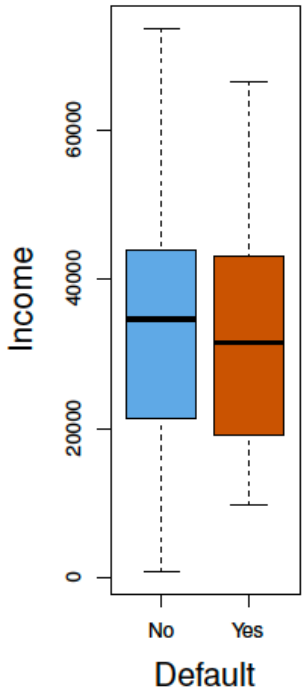


Example: Default Data

We again consider the Default data. We consider if the balance or income is related to the credit card default.







Using Linear Regression?

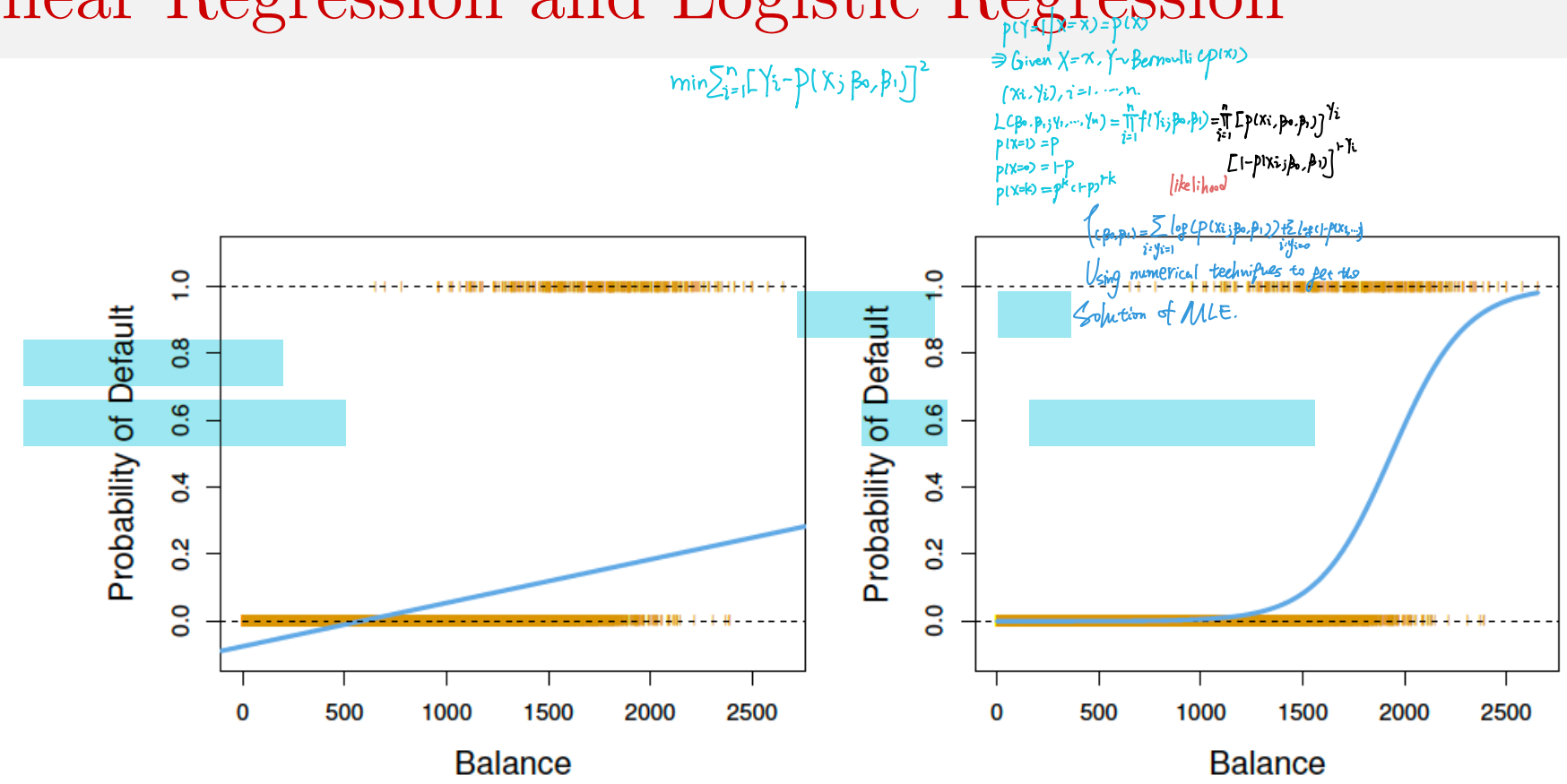
Suppose for the Default classification task that we code

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

Can we simply perform a linear regression of Y on X and classify it as Yes if $\hat{Y} > 0.5$?

- In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to *linear discriminant analysis* which we discuss later;
- Since in the population $E[Y|X = x] = P(Y = 1|X = x)$, we might think that regression is perfect for this task;
- However, linear regression might produce probabilities less than zero or bigger than one (R);
- *Logistic regression* is more appropriate.

Linear Regression and Logistic Regression



- The orange marks indicate the response Y, either 0 or 1;
- Linear regression does not estimate $P(Y = 1|X)$ well;
- Logistic regression seems well suited to the task.

More than Two Categories

- Now suppose we have a response variable with three possible values. A patient presents the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if stroke (中風)} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure (癲癇).} \end{cases}$$

- This coding suggests an ordering, and in fact implies that the difference between *stroke* and *drug overdose* is the same as between *drug overdose* and *epileptic seizure*.
- Linear regression is not appropriate here.
- *Multi-class logistic regression* or *Discriminant analysis* are more appropriate.

Logistic Regression

- We write $p(X) = P(Y = 1|X)$ for short and consider using *balance* to predict *default*. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

- It is easy to see that no matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.
- A bit of rearrangement gives

$$\log(\frac{p(X)}{1 - p(X)}) = \beta_0 + \beta_1 X.$$

- This monotone transformation is called the *log odds* or *logit transformation* *of* $p(X)$.
- Logistic regression ensures that our estimate of $p(X)$ lies between 0 and 1!

```r
library(MASS)
library(sparsediscrim)
#Load data
data=Default
y=as.numeric(data$default=="Yes")
x1=data$student
x2=data$balance
x3=data$income
#Sigma=cov_pool(cbind(x2),y)
#Inv=solve(Sigma)
#Create plot
plot(x2,x3,col=as.factor(y),xlab="balance",ylab="income")
```

```
#########################Use the gradient descent#####################################
#Predictor variables
X1<-as.matrix(cbind(x2))
X2<-as.matrix(cbind(x3))
#Add ones to X
X <- cbind(rep(1,nrow(X1)),X1)
#Response variable
Y <- as.matrix(y)
#Sigmoid function
sigmoid<-function(z)
{
g<-1/(1+exp(-z))
return(g)
}
#Cost Function
cost <- function(beta)
{
m <- nrow(X)
g <- sigmoid(X%*%beta)
J <- (1/m)*sum((-Y*log(g))-((1-Y)*log(1-g)))
return(J)
}
#Intial beta
initial_beta <- rep(0,2)
#Cost at inital beta
cost(initial_beta)
```

Environment is empty

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(MASS)
> library(sparsediscrim)
> #Load data
> data=Default
Error: object 'Default' not found
> library(MASS)
> library(sparsediscrim)
> #Load data
> data=Default
Error: object 'Default' not found
```

```r
}
#Cost Function
cost <- function(beta)
{
m <- nrow(X)
g <- sigmoid(X%*%beta)
J <- (1/m)*sum((-Y*log(g))-((1-Y)*log(1-g)))
return(J)
}
#Intial beta
initial_beta <- rep(0,2)
#Cost at inital beta
cost(initial_beta)
# Derive beta using gradient descent using optim function
beta_optim <- optim(par=initial_beta,fn=cost)
#set beta
beta <- beta_optim$par
#cost at optimal value of the beta
beta_optim$value
#lines(X[,2],-(beta[1]+beta[2]*X[,2])/beta[3])
# probability of admission for student
prob <- sigmoid(t(c(1,500))%*%beta)
##########################################################################################
##############################Use the glm command#########################################
##########################################################################################
```

Environment is empty

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(MASS)
> library(sparsediscrim)
> #Load data
> data=Default
Error: object 'Default' not found
> library(MASS)
> library(sparsediscrim)
> #Load data
> data=Default
Error: object 'Default' not found
```

Estimating Logistic Regression: Maximum Likelihood

- With the training data $\{(x_i, y_i), i = 1, \cdots, n\}$, we use the maximum likelihood to estimate the parameters:

$$\ell(\beta_0, \beta_1) = \prod_{i=1}^{n} [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i} = \Big(\prod_{i:y_i=1} p(x_i) \Big) \times \Big(\prod_{i:y_i=0} [1 - p(x_i)] \Big),$$

- This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood the observed data,
- Most statistical package can fit linear logistic regression model by maximum likelihood.

Estimating Logistic Regression: Maximum Likelihood

We study the Default data using *glm* function in R:

	Coefficient	*SE*	*Z-statistic*	$p-value$
$Intercept(\hat{\beta}_0)$	-10.65	0.3612	-29.5	< 0.0001
$balance(\hat{\beta}_1)$	0.0055	0.0002	24.9	< 0.0001

R command:

```
library(ISLR2)
library(ISLR)
data=Default
y=as.numeric(data$default=="Yes")
x=data$balance
fit=glm(y~x,family=binomial)
summary(fit)
```

Making Predictions

What is the estimated probability of *default* for someone with a balance of \$1000?

$$\hat{p}(1000) = \frac{e^{\hat{\beta}_0 + 1000 \cdot \hat{\beta}_1}}{1 + e^{\hat{\beta}_0 + 1000 \cdot \hat{\beta}_1}} = 0.0006.$$

With a balance of \$2000, we have

$$\hat{p}(2000) = \frac{e^{\hat{\beta}_0 + 2000 \cdot \hat{\beta}_1}}{1 + e^{\hat{\beta}_0 + 2000 \cdot \hat{\beta}_1}} = 0.586.$$

```
newdata=data.frame(x=c(1000,2000))
prob=predict(fit,newdata,type = "response")
```

Student as Predictor

Now we use *student* as the predictor, we obtain:

	Coefficient	*SE*	*Z-statistic*	*p-value*
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student	0.4049	0.115	-3.52	0.0004

$$\hat{P}(default = Yes|student = Yes) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\hat{P}(default = Yes|student = No) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Logistic Regression with Several Variables

We consider predict the default with *income*, *student*, *balance* together.

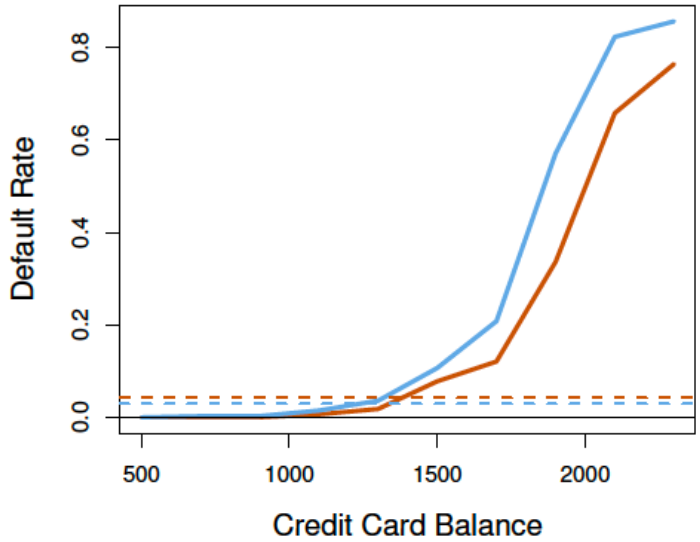
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X + \cdots + \beta_p X_p,$$

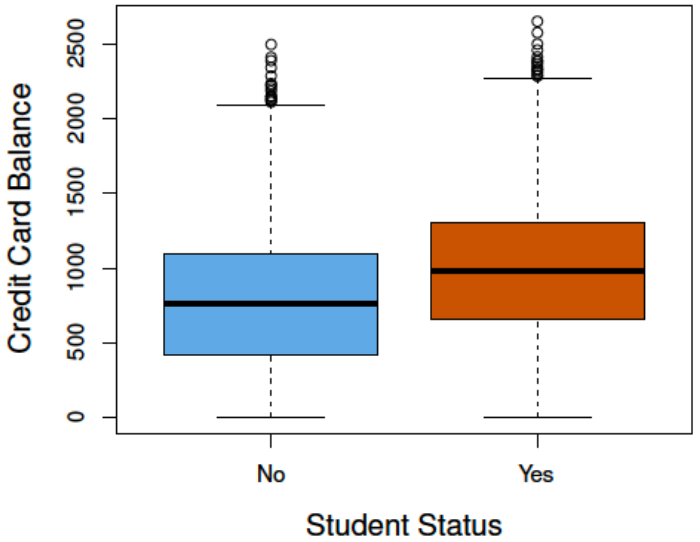
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

	Coefficient	*S.E.*	*Z-statistic*	*p-value*
Intercept	-10.869	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.003	0.0082	0.37	0.7115
student[*Yes*]	-0.6468	0.2362	-2.74	0.0062

Confounding

- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students,
- But for each level of balance, students default is less than non-student,
- Multiple logistic regression can tease this out.





Logistic Regression: Smarket Data

We use Smarket data to examine the method.

```
> names(Smarket)
[1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"      "Volume"    "Today"
[9] "Direction"
> dim(Smarket)
[1] 1250    9
> summary(Smarket)
      Year           Lag1                Lag2                Lag3                Lag4         
 Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000  
 1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000   1st Qu.:-0.640000  
 Median :2003   Median : 0.039000   Median : 0.039000   Median : 0.038500   Median : 0.038500  
 Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716   Mean   : 0.001636  
 3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750  
 Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000  
      Lag5              Volume            Today           Direction 
 Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000   Down:602  
 1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500   Up  :648  
 Median : 0.03850   Median :1.4229   Median : 0.038500             
 Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138             
 3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750             
 Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000             
```

This data set consists of percentage returns for the 500 stock index over 1250 days, from 2001 to 2005.

Logistic Regression: Smarket Data

For each date, it recorded the percentage returns for the 5 previous reading days, Lag1 through Lag5.
The Volume recorded is the number of shares traded on the previous day. The Today recorded is the percentage return on the data in question . And the Direction is whether the market was Up or Down on the date.

```
> head(Smarket)
  Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
1 2001  0.381 -0.192 -2.624 -1.055  5.010 1.1913  0.959        Up
2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032        Up
3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623      Down
4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614        Up
5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213        Up
6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392        Up
```

Fitting results

fit generalized linear models.

```
> glm.fits = glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket, family=binomial)
> summary(glm.fits)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6

Number of Fisher Scoring iterations: 3
```

better with logistic regression model

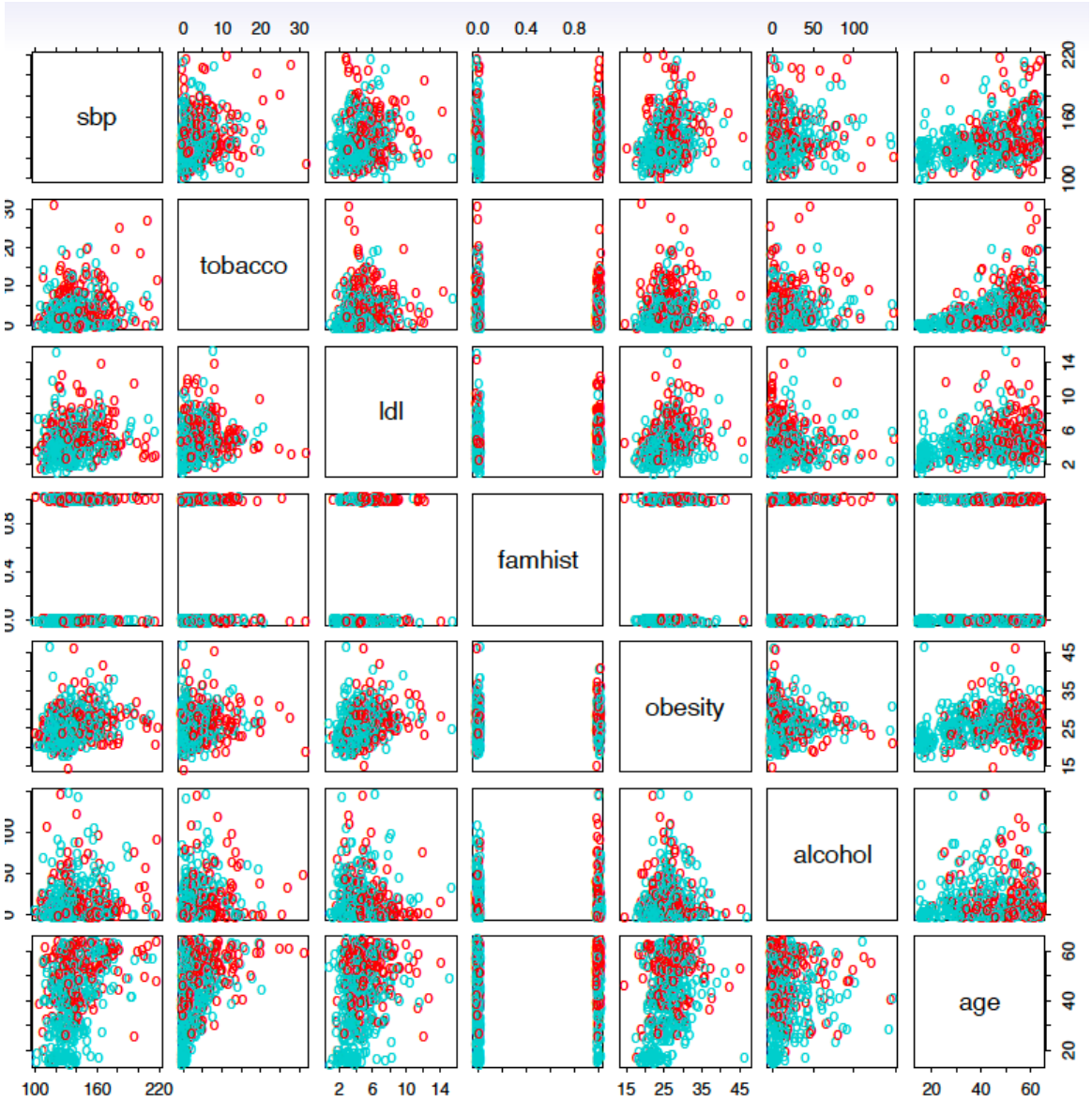
$$p(up|X_1, X_2, \cdots, X_6) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1 + \cdots + \hat{\beta}_6 X_6}}{1+e^{\hat{\beta}_0+\cdots+\hat{\beta}_6 X_6}}$$

useful information.

Logistic Regression: South African Heart Disease

- 160 cases of myocardial infarction (MI, 心肌梗塞) and 302 controls (all male in age range 15-64), from Western Cape, South Africa in early 80s,
- Measurements on seven predictor (risk factors), shown in scatterplot matrix,
- Goal is to identify relative strengths and directions of risk factors,
- This was part of an intervention study aimed at educating the public on healthier diets.

Plots



Scatterplot matrix of the *South African Heart Disease* data. The response is colour coded - The cases (MI) are red, the controls turquoise. *famhist* is a binary variable, with 1 indicating family history of MI.

command

```
library(ElemStatLearn)
attach(SAheart)
heartfit=glm(chd~.,data=SAheart,family=binomial)
summary(heartfit)
```

Logistic Regression with more than two classes

- So far we have discussed logistic regression with two classes. It can be generalised to more than two classes. Suppose that we have $k = 1, \cdots, K$ classes, we may select one class, say Kth class, to serve as a baseline,

$$P(Y = k|X) = \frac{e^{\beta_{k,0}+\beta_{k,1}X_1+\cdots+\beta_{k,p}X_p}}{1 + \sum_{k=1}^{K-1} e^{\beta_{k,0}+\beta_{k,1}X_1+\cdots+\beta_{k,p}X_p}},$$

and

$$P(Y = K|X) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_{k,0}+\beta_{k,1}X_1+\cdots+\beta_{k,p}X_p}}.$$

It is not hard to show that for $k = 1, \cdots, K-1$,

$$\log\left(\frac{P(Y = k|X)}{P(Y = K|X)}\right) = \beta_{k,0} + \beta_{k,1}X_1 + \cdots + \beta_{k,p}X_p.$$

- Multi-class logistic regression is also referred to as *multi-class regression*.

$$p(y=1 \mid x=x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

log odd:

$$\log \frac{p(y=1 \mid x=x)}{p(y=0 \mid x=x)} = \beta_0+\beta_1 x.$$

For the case of $Y \in \{c_1, \cdots c_k\}$ with $k>2$,

$$\log \frac{p(y=1 \mid x=x)}{p(y=k \mid x=x)} = \beta_{0,1}+\beta_{1,1} x$$

$$\vdots$$

$$\log \frac{p(y=k-1 \mid x=x)}{p(y=k \mid x=x)} = \beta_{0,k-1}+\beta_{1,k-1} x$$

parameters: $(\beta_{0,1}, \beta_{1,1}, \cdots \beta_{0,k-1}, \beta_{1,k-1}), 2\times(k-1)$

if p variables

we have $(p+1)\cdot(k-1)$ parameters.

$$p(y=1 \mid x=x) = e^{\beta_{0,1}+\beta_{1,1} x} \cdot p(y=k \mid x=x)$$

$$\vdots$$

$$p(y=k-1 \mid x=x) = e^{\beta_{0,k-1}+\beta_{1,k-1} x} p(y=k \mid x=x).$$

$$1 - p(y=k \mid x=x) = p(y=k \mid x=x) \cdot \sum_{j=1}^{k-1} e^{\beta_{0,j}+\beta_{1,j} x}$$

$$\Rightarrow p(y=k \mid x=x) = \frac{1}{1+\sum_{j=1}^{k-1} e^{\beta_{0,j}+\beta_{1,j} x}} = p_k$$

Hence;

$$p(y=j \mid x=x) = \frac{e^{\beta_{0,j}+\beta_{1,j} x}}{1+\sum_{j=1}^{k-1} e^{\beta_{0,j}+\beta_{1,j} x}} \quad \text{for } j=1, \cdots, k-1.$$

Then the likelihood is: $= p_j$

$$p(x) = \prod_{j=1}^{k} p_j^{I\{y=j\}}$$

Logistic Regression with more than two classes

Define $p_k(X) = P(Y = k|X)$. Then, we can maximize the likelihood, which is

$$L(\beta_{k,0}, \cdots, \beta_{k,p}, k = 1, \cdots, K-1) = \prod_{i=1}^{n} \Big(\prod_{k=1}^{K} [p_k(X_i)]^{\mathbf{1}_{\{Y_i = k\}}} \Big).$$

We classify the data x into $\hat{k}$th class, such that

$$\hat{k} = \arg\max_k \{\hat{p}_k(x)\}.$$

Softmax coding

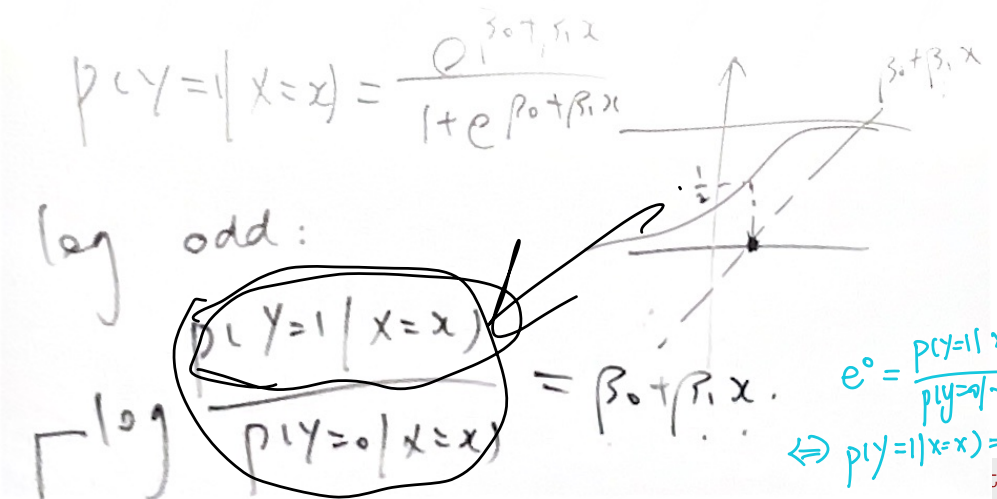
- We may also try the softmax coding in the multi-class logistic regression (package *glmnet*). It has the symmetric form

$$P(Y = k|X) = \frac{e^{\beta_{k,0}+\beta_{k,1}X_1+\cdots+\beta_{k,p}X_p}}{\sum_{k=1}^{K} e^{\beta_{k,0}+\beta_{k,1}X_1+\cdots+\beta_{k,p}X_p}},$$

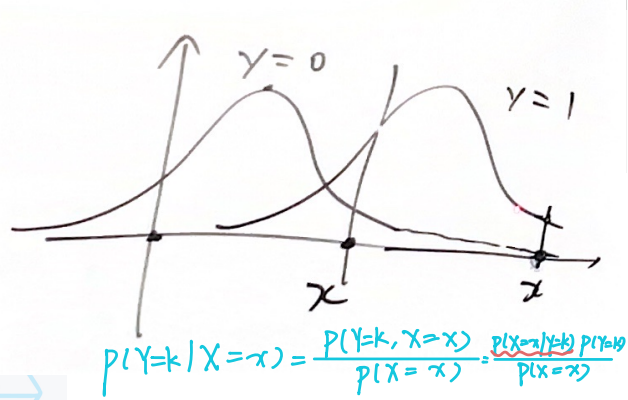
for $k = 1, \cdots, K$.

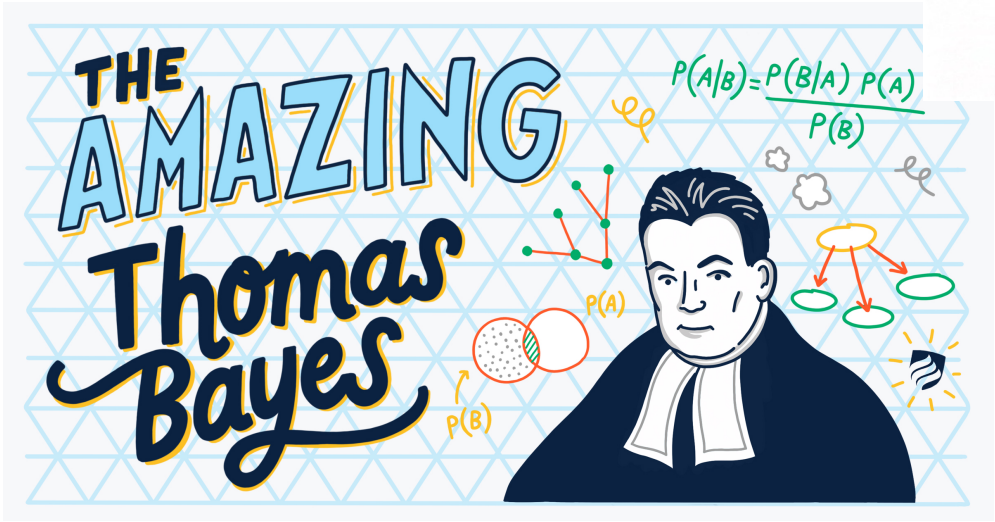
Discriminant Analysis

- Here the approach is to model the distribution of X in each of the classes separately, and then use *Bayes Theorem* to flip things around and obtain $P(Y = k|X)$,
- When we use normal distributions for each class, this leads to linear or quadratic discriminant analysis,
- However, this approach is quite general, and other distributions can be used as well, we will focus on normal distribution.



Bayes Theorem for Classification





1701-1761, English statistician, philosopher

Bayes Theorem:

$$= \frac{f_k(x)\,\pi_k}{f(x)} = \frac{f_k(x)\,\pi_k}{\sum_{j=1}^{K} f_j(x)\,\pi_j}$$

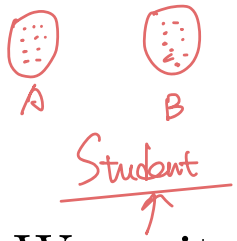
where π_k: marginal dist. of y
$f_k(x)$: conditional dist. of $x|y=k$

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

$$= \frac{P(X = x|Y = k) \cdot P(Y = k)}{\sum_{k=1}^{K} P(X = x|Y = k) \cdot P(Y = k)}.$$

Bayes Theorem for Classification

let $P(X=x|Y=k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$ LDA good choice of normal



already know which one is better and apply logistic regression. $P(Y=1|X=x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$

- We write this for discriminant analysis:

$$P(Y=k|X=x) = \frac{\pi_k f_k(x)}{\sum_{k=1}^{K} \pi_k f_k(x)},$$

where $f_k(x) = P(X=x|Y=k)$ is the *density* for X in class k. We will use normal densities for these, separately in each class,

- $\pi_k = P(Y=k)$ is the marginal or *prior* probability for class k.

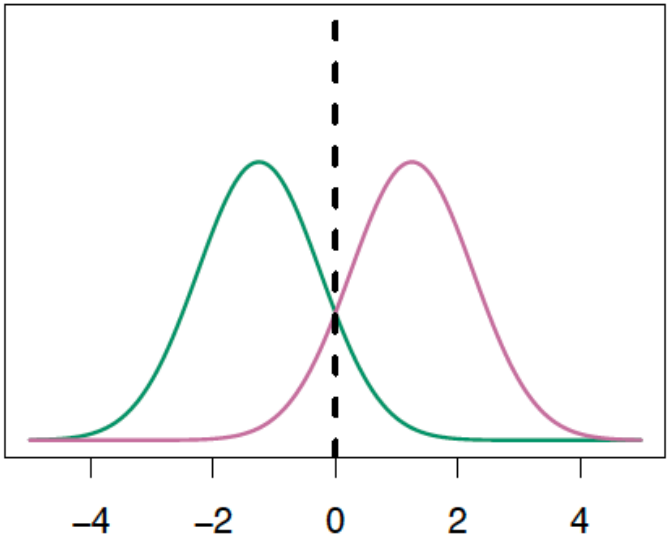
Classify to the Highest Density

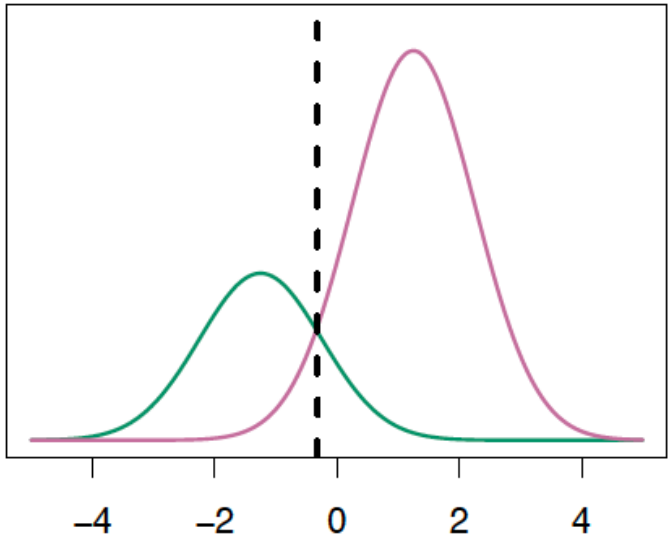
$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$

when dist. is continuous, the first choice must be normal distribution.

We classify a new point according to which density is highest.

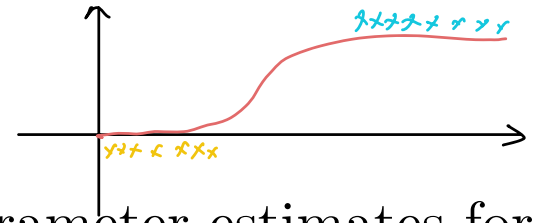
π_1=.5, π_2=.5 μ_k depends on k and is different, but σ^2 is common for all classes. π_1=.3, π_2=.7





When the priors are different, we take them into account as well, and compare $\pi_k f_k(x)$. On the right, we favour the pink class - the decision boundary has shifted to the left.

Why Discriminant Analysis?



- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem,
- If n is small and distribution of the predictors X 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.

Linear Discriminant Analysis when $p = 1$

- The Gaussian density has the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}},$$

where μ_k is the mean, and σ_k^2 the variance. We will assume that all the $\sigma_k = \sigma$ are the same (in the linear discriminant analysis).

- Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = P(Y = k|X = x)$:

$$p_k(x) = \frac{\pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}}{\sum_{k=1}^{K} \left\{ \pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}} \right\}}.$$

We happily see that there are simplifications and cancellations.

$$p(y=1|x=x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

In particular, if $k=2$:

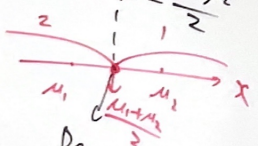
$$\log P_x(1) = \log C + \log \pi_1 - \frac{\mu_1^2}{2\sigma^2} + \frac{\mu_1}{\sigma^2}x$$

$$\log P_x(2) = \log C + \log \pi_2 - \frac{\mu_2^2}{2\sigma^2} + \frac{\mu_2}{\sigma^2}x$$

If $\pi_1 = \pi_2$ ($p(y=1)=p(y=2)$)

$$\delta_x = \log P_x(1) - \log P_x(2) = \frac{1}{2\sigma^2}(\mu_2-\mu_1)x + \frac{1}{2\sigma^2}(\mu_1^2-\mu_2^2)$$

$$\delta(x) = 0 \Rightarrow x = \frac{\mu_1+\mu_2}{2} - \frac{1}{2\sigma^2}(\mu_1+\mu_2)\frac{\sigma^2}{(\mu_2-\mu_1)} = \frac{x-\frac{1}{2}(\mu_1+\mu_2)}{(\mu_2-\mu_1)}$$



Decision boundary.

Hence,

$$P_x(k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}} \cdot \pi_k = \frac{e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}\pi_k}{\sum_{j=1}^{k} e^{-\frac{(x-\mu_j)^2}{2\sigma^2}}\pi_j} = \frac{e^{-\frac{x^2-2\mu_k x+\mu_k^2}{2\sigma^2}}\pi_k}{\sum_{j=1}^{k} e^{-\frac{x^2-2\mu_j x+\mu_j^2}{2\sigma^2}}\pi_j}$$

does not depend on k

$$= C \cdot e^{\frac{2\mu_k x-\mu_k^2}{2\sigma^2}}\pi_k = C \cdot e^{\frac{\mu_k}{\sigma^2}x - \frac{\mu_k^2}{2\sigma^2}}\cdot \pi_k$$

$$\Rightarrow \log P_x(k) = \log C + \log \pi_k - \frac{\mu_k^2}{2\sigma^2} + \frac{\mu_k}{\sigma^2}x$$

Then:

$$\hat{k} = \underset{k}{\operatorname{argmax}} \left\{ \left(\log \pi_k - \frac{\mu_k^2}{2\sigma^2}\right) + \frac{\mu_k}{\sigma^2}x \right\}$$

Linear function $\delta^k(x)$

$$\kappa = \frac{\mu_1+\mu_2}{2}$$

$$p(y=k|x=x) = \frac{p(x=x|y=k)\,p(y=k)}{p(x=x)}$$

Then

$$P_x(k) = p(y=k|x=x) = \frac{f_k(x)\pi_k}{\sum_{j=1}^{k} f_j(x)\pi_j}$$

$$= \frac{f_k(x)\pi_k}{f(x)}$$

$$= \frac{f_k(x)\pi_k}{\sum_{j=1}^{k} f_j(x)\pi_j}$$

where: π_k : margin dist. of y

$f_k(x)$: conditional dist. of $x|y=k$

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$$ LDA (Linear discriminant Analysis)

σ^2 is common for all classes.

In particular, if $k=2$:

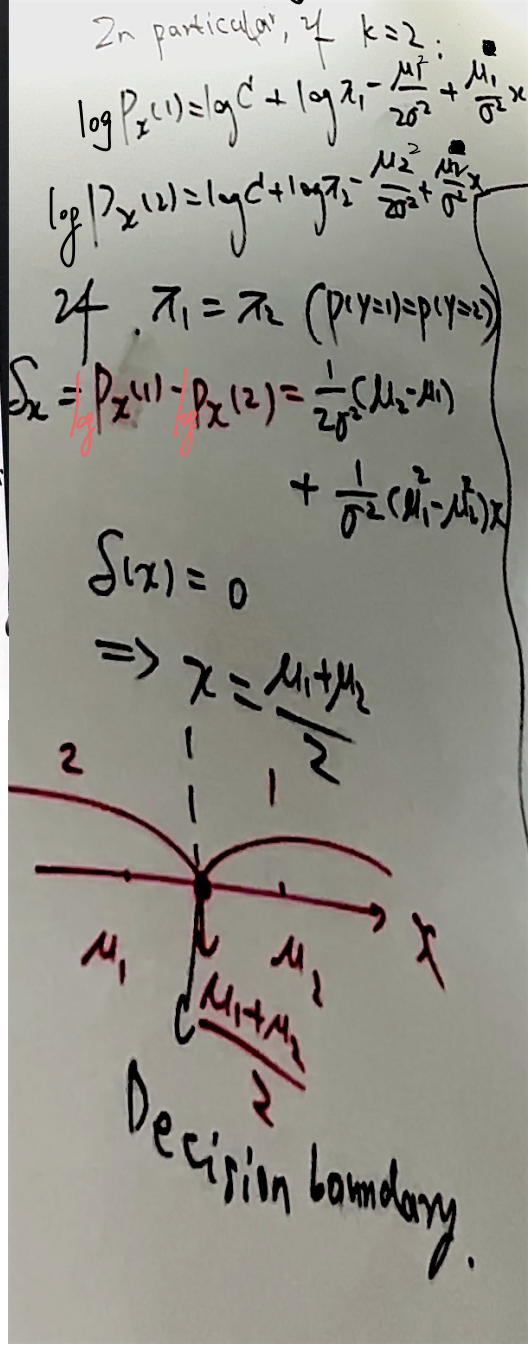
$$\log P_x(1) = \log C + \log \pi_1 - \frac{\mu_1^2}{2\sigma^2} + \frac{\mu_1}{\sigma^2}x$$

$$\log P_x(2) = \log C + \log \pi_2 - \frac{\mu_2^2}{2\sigma^2} + \frac{\mu_2}{\sigma^2}x$$

If $\pi_1 = \pi_2$ ($p(y=1)=p(y=2)$)

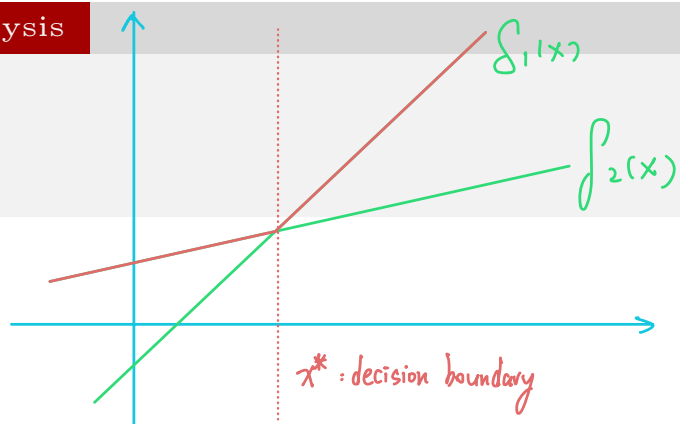
$$\delta_x = \log P_x(1) - \log P_x(2) = \frac{1}{2\sigma^2}(\mu_2-\mu_1)x + \frac{1}{2\sigma^2}(\mu_1^2-\mu_2^2)$$

$$\delta(x) = 0 \Rightarrow x = \frac{\mu_1+\mu_2}{2}$



Decision boundary.

Discriminant Functions



- Assume $\sigma_k^2 \equiv \sigma^2$. To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest. Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest *discriminant score*:

$$\delta_k(x) = \left\{ \log(\pi_k) - \frac{\mu_k^2}{2\sigma^2} \right\} + \frac{\mu_k}{\sigma^2} x,$$

  note that $\delta_k(x)$ is a *linear* function of x,

- If there are $K = 2$ classes and $\pi_1 = \pi_2 = 0.5$, then one can see that the decision boundary is at

$$x = \frac{\mu_1 + \mu_2}{2}.$$

$p=1$

$p=0$ decision boundary

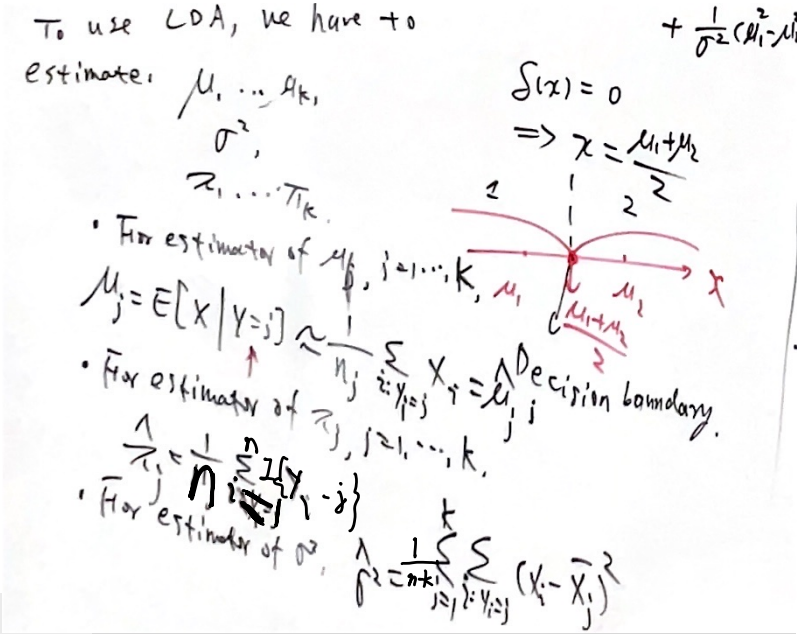
$\log(\pi_1) = \log(\pi_2)$

Estimating the Parameters

Typically we do not know these parameters, we just have a training data. In that case we simple estimate the parameters and plug them into the rule. We estimate the parameter by:

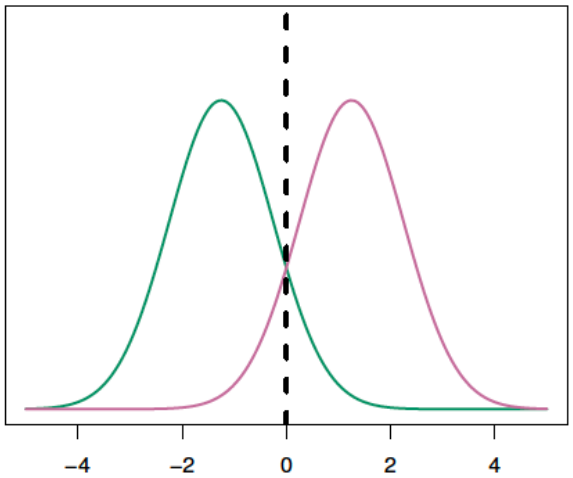
$$\begin{aligned}
\hat{\pi}_k &= \frac{n_k}{n}, \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i, \\
\hat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2, \\
&= \sum_{k=1}^{K} \frac{n_k - 1}{n - K} \hat{\sigma}_k^2,
\end{aligned}$$

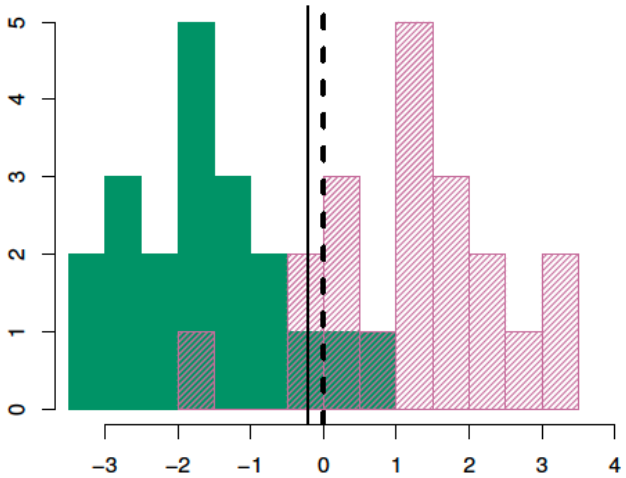
where the $\hat{\sigma}^2$ is a weighted sum.



Linear Discriminant Analysis: Example

- Example with $\mu_1 = -1.5, \mu_2 = 1.5, \pi_1 = \pi_2 = 0.5$ and $\sigma^2 = 1$,





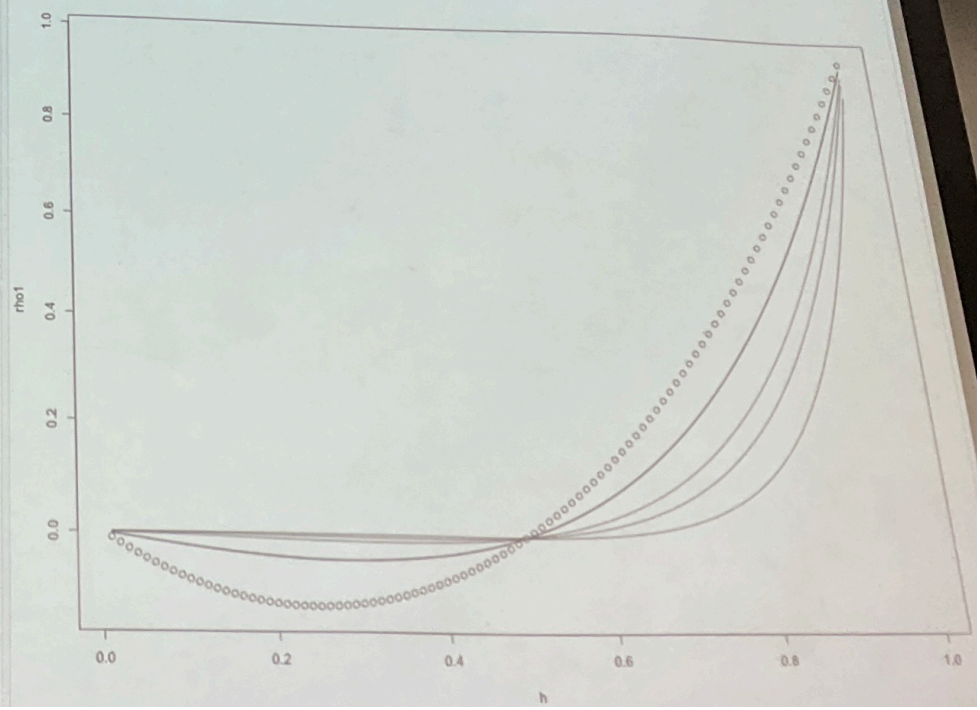
R command

RStudio

```
data1=rnorm(500,-1.5,1)
y1=rep(0,500)
y2=rep(1,500)
data2=rnorm(500,1.5,1)
data=c(data1,data2)
y=c(y1,y2)
hist(data1,xlim=c(-5,5),ylim=c(0,400),col="red")
hist(data2,add=T,col=rgb(0,1,0,0.5))
abline(v=0,col="red",lty=2)
abline(v=(mean(data1)+mean(data2))/2,col="black")
```

data	num [1:1000] -1.551 -3.011 0.253 0.191 -2.588 ...
data1	num [1:500] -1.551 -3.011 0.253 0.191 -2.588 ...
data2	num [1:500] 1.182 1.233 1.309 1.1 0.384 ...
h	num [1:99] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
k1	1
k2	2
k3	5

```
> rho4=h*(2*h-1)*k4^(2*h-2);
> rho5=h*(2*h-1)*k5^(2*h-2);
> plot(h,rho1)
> lines(h,rho2)
> lines(h,rho2)
> lines(h,rho3)
> lines(h,rho4)
> lines(h,rho5)
> data1=rnorm(500,-1.5,1)
> data2=rnorm(500,1.5,1)
> y1=rep(0,500)
> y2=rep(1,500)
> data2=rnorm(500,1.5,1)
> data=c(data1,data2)
```



Linear Discriminant Analysis when $p > 1$

- The multivariate normal distribution has density

$$f(x) = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)},$$

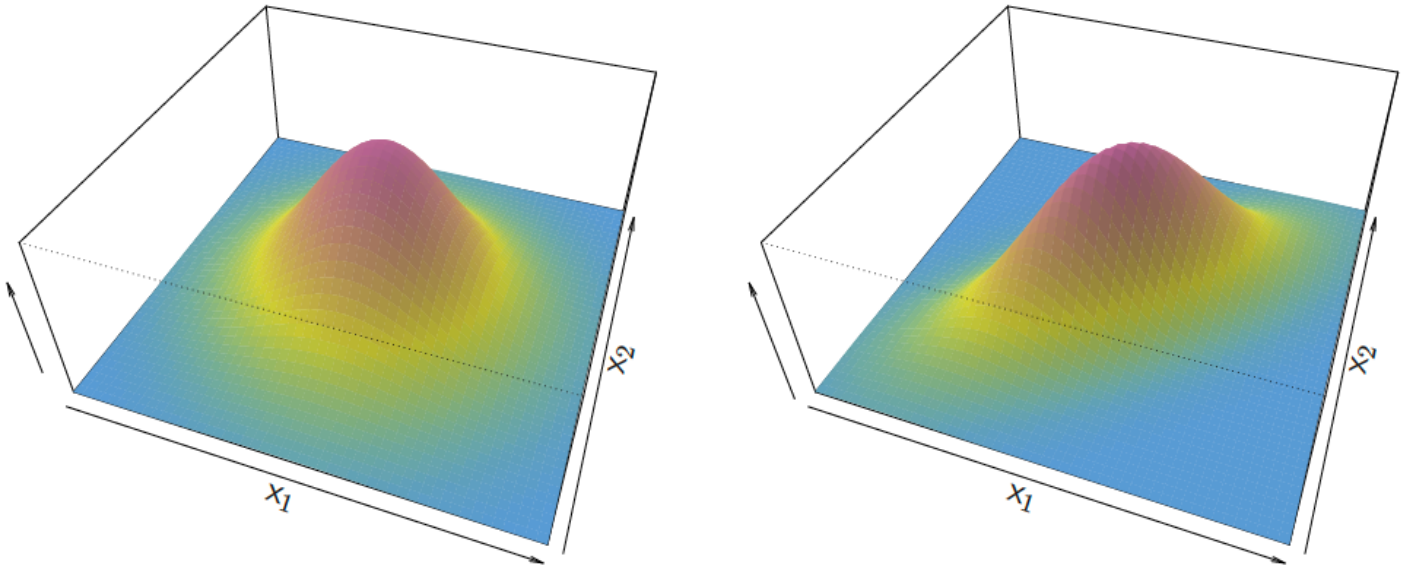
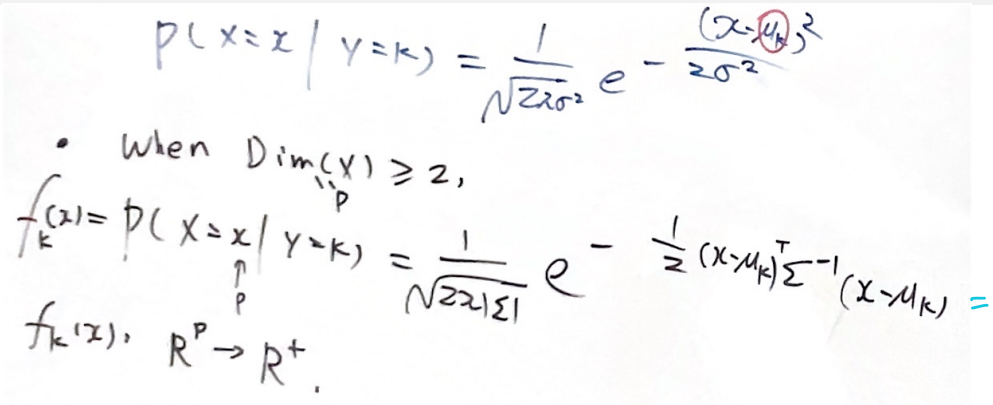


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

Linear Discriminant Analysis when $p > 1$



$= \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_k - \mu_k^T\Sigma^{-1}x + \mu_k^T\mu_k)}$

The discriminant function is

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

it is a linear function, despite its complex form.

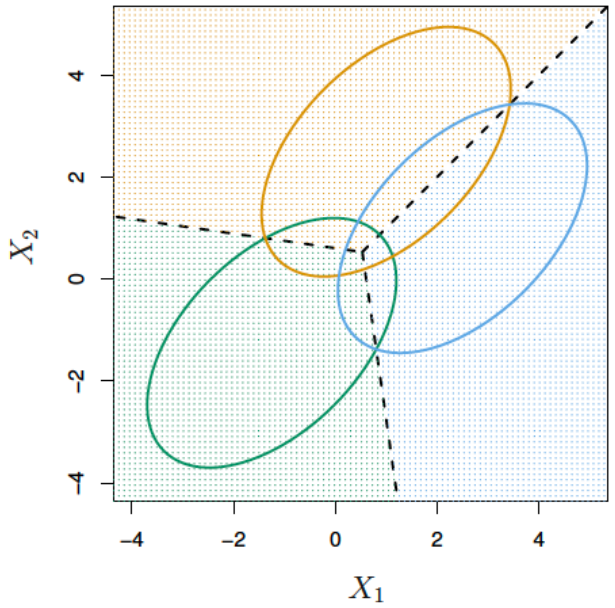
$$\delta_k(x) = \left\{ \log(\pi_k) - \frac{\mu_k^2}{2\sigma^2} \right\} + \frac{\mu_k}{\sigma^2} x$$

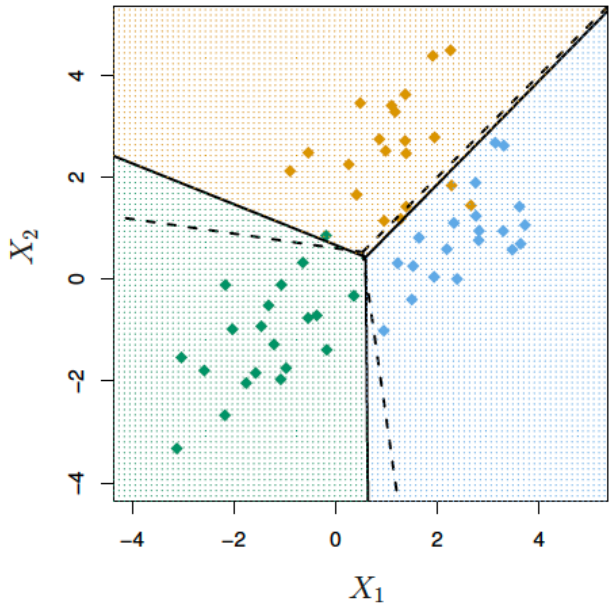
Linear Discriminant Analysis: Illustration

Consider an example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.

- Left: Ellipses that contain 95% of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries;
- Right: 20 observations were generated from each class ($\pi_k = 1/3$), and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.

p=2 p=1 decision boundary





$$\frac{p(X=x \mid Y=k)\, p(Y=k)}{\sum_{j=1}^{K} p(X=x \mid Y=j)\, p(Y=j)} = p(Y=k \mid X=x)$$

LDA

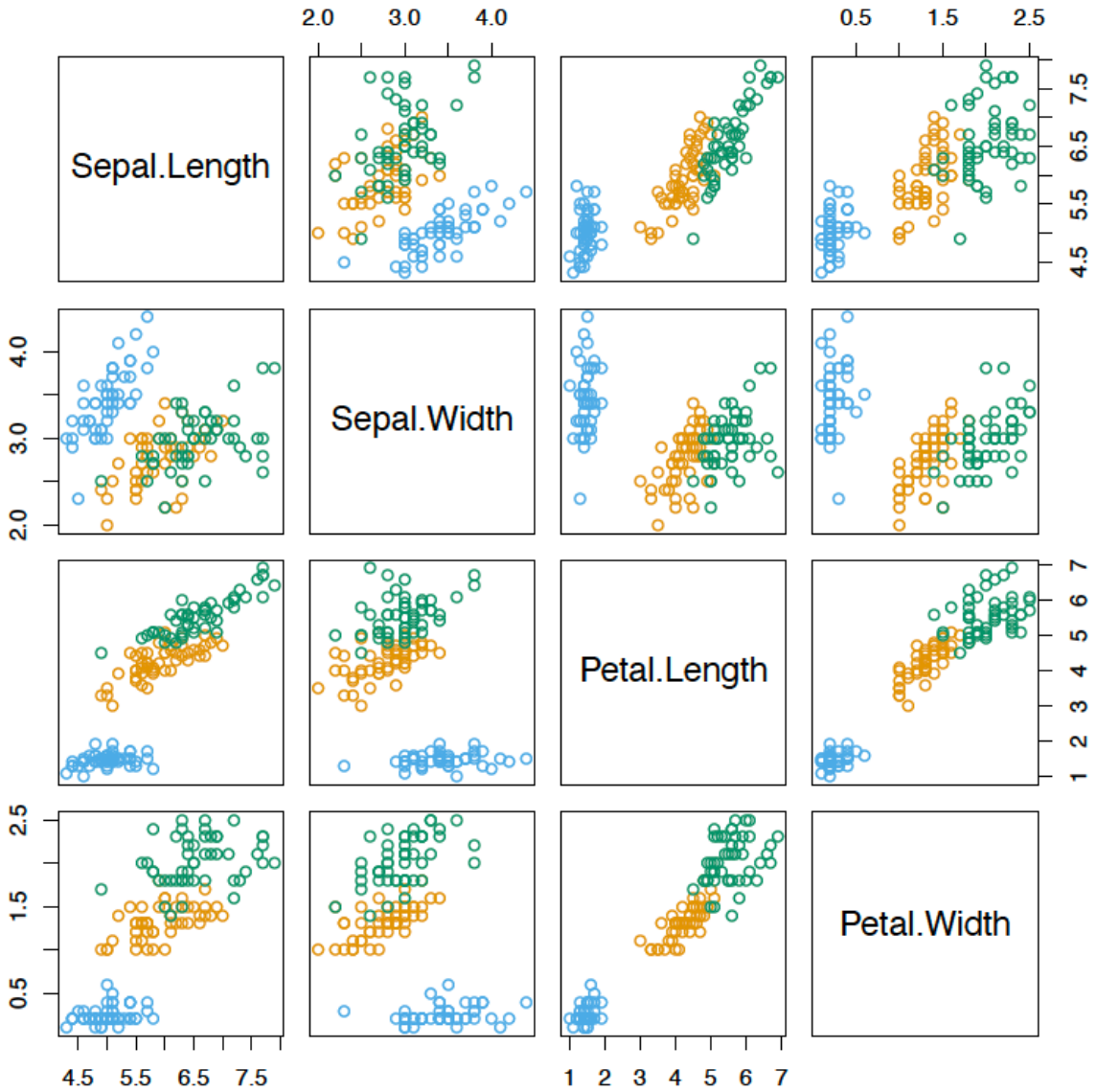
LR

$$p(X=x \mid Y=k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}$$

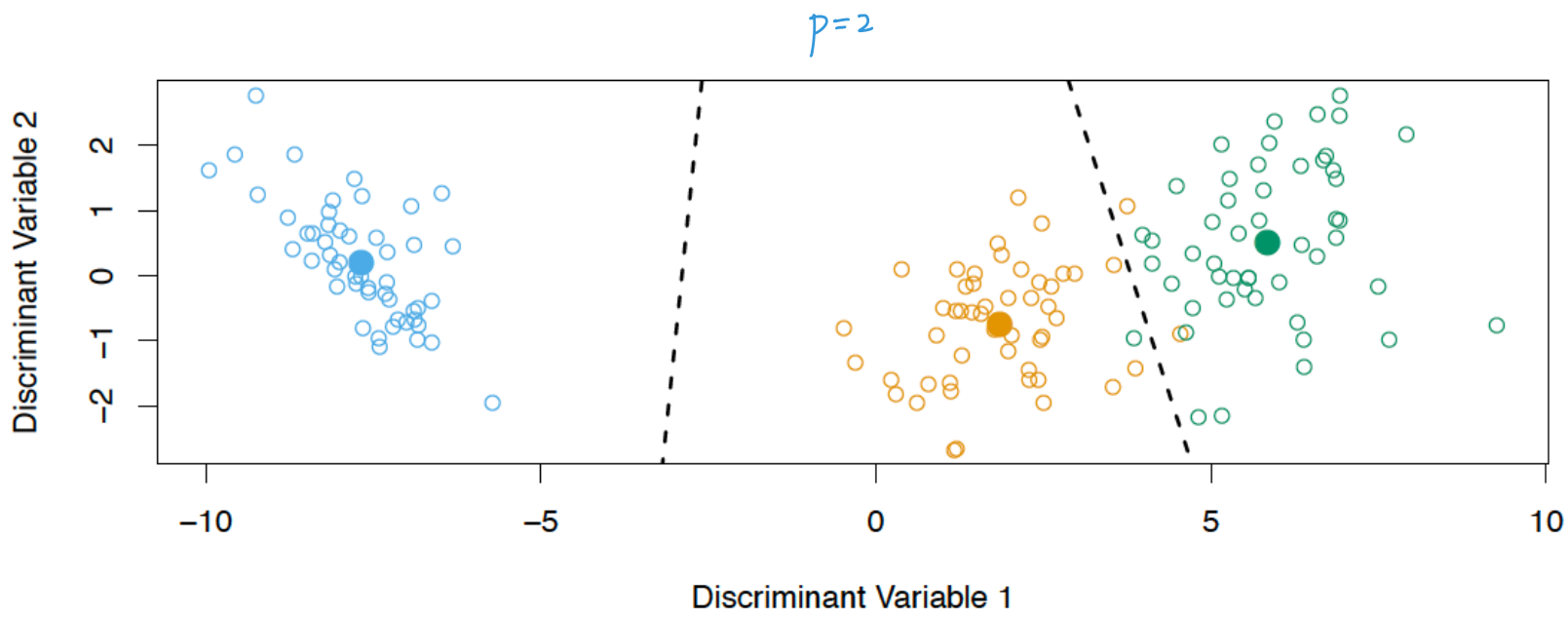
probability density function

Fisher's Iris Data

- 4 variables:
- 3 species
- 50 samples
- Sepal.Length and Sepal.Width are mostly uncorrelated
- LDA classifies all but 3 of the 150 training samples correctly.

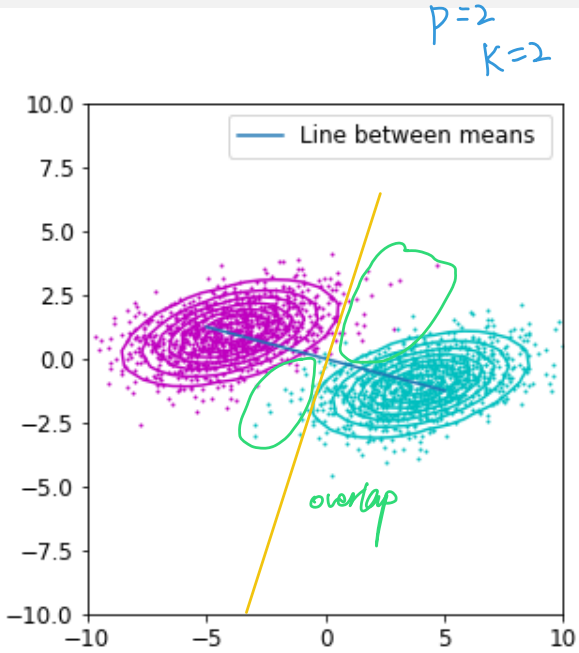


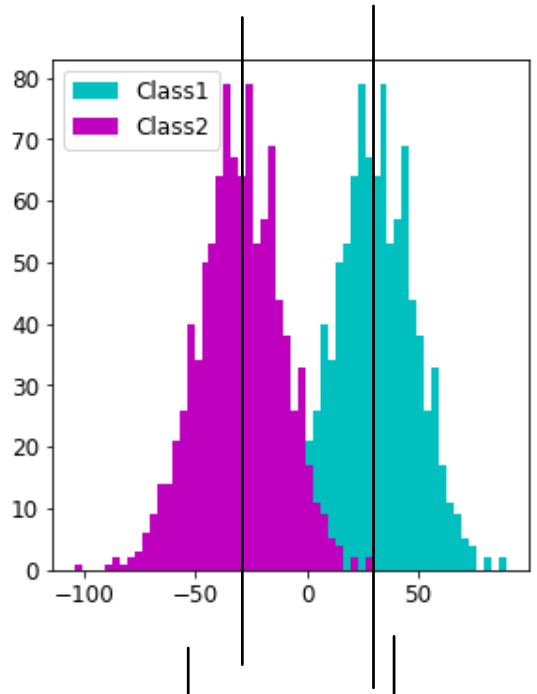
Fisher's Discriminant Plot

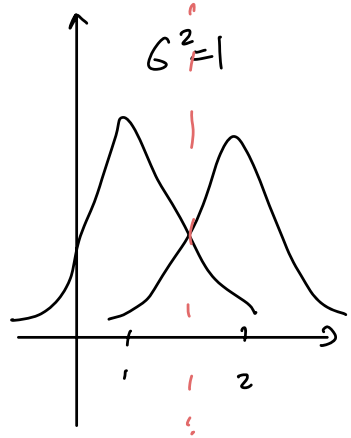


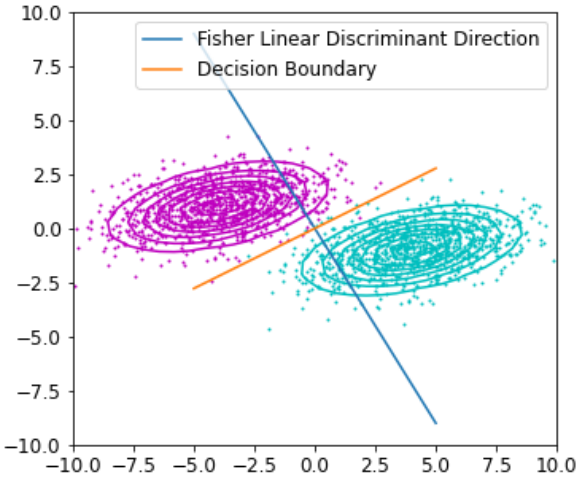
- When there are K classes, linear discriminant analysis can be viewed exactly in a $K-1$ dimensional plot, because it essentially classifies to the closest centroid, and they span a $K-1$ dimensional plane.

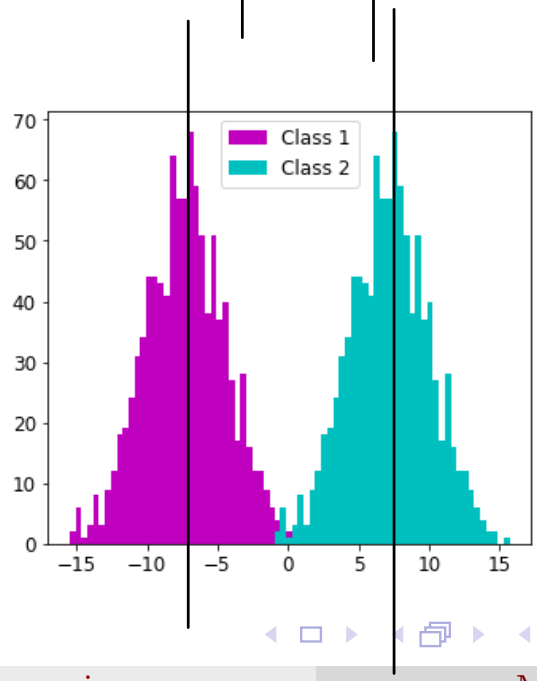
Fisher's Discriminant Plot

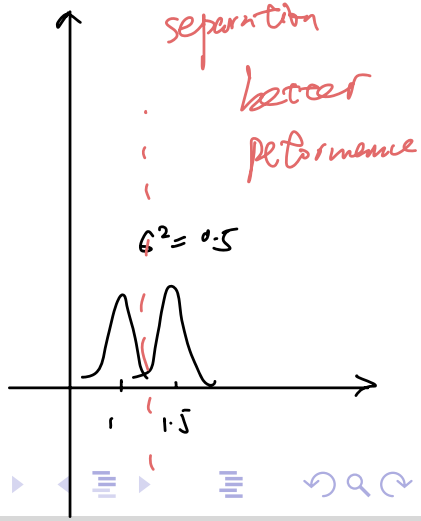




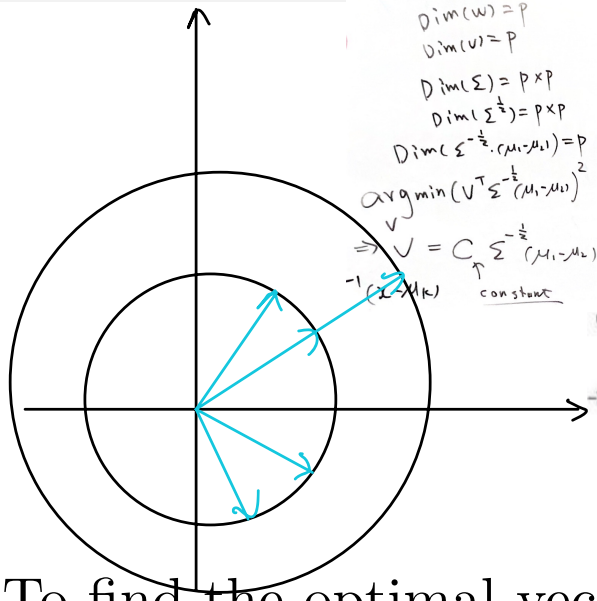


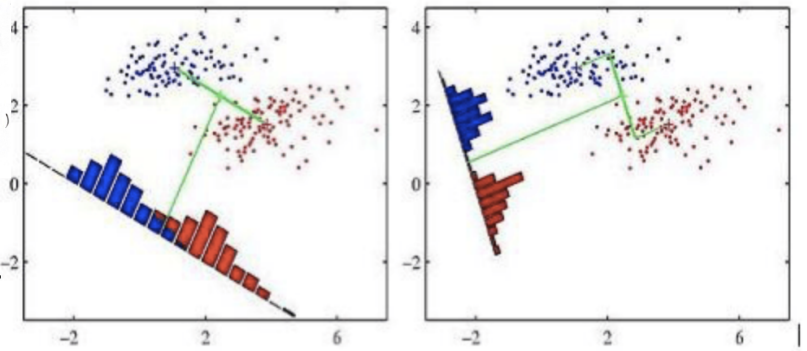


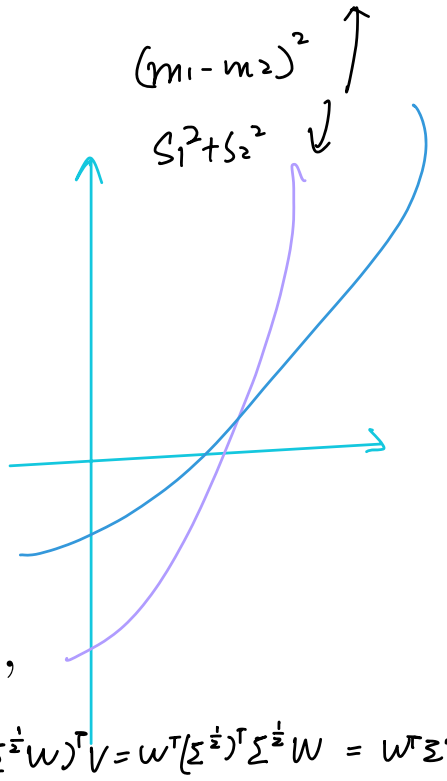




Fisher's Discriminant Plot







To find the optimal vector, we maximize

$$f(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{(\mathbf{w}^\top(\mu_1 - \mu_2))^2}{\mathbf{w}^\top \Sigma \mathbf{w}},$$

where $\Sigma = \sum_{i:Y_i=0}(x_i - \mu_1)^2 + \sum_{i:Y_i=1}(x_i - \mu_2)^2$. Let $\mathbf{v} = \Sigma^{\frac{1}{2}}\mathbf{w}$, then we have $f(\mathbf{w}) = g(\mathbf{v}) = \frac{[\mathbf{v}^\top \Sigma^{-\frac{1}{2}}(\mu_1 - \mu_2)]^2}{\mathbf{v}^\top \mathbf{v}}$. The *length* of $\mathbf{v}$ does not change the value of the function, hence we assume $\mathbf{v}^\top \mathbf{v} = 1$, the solution is $\mathbf{v} \sim \Sigma^{-\frac{1}{2}}(\mu_1 - \mu_2)$, that is $\mathbf{w} \sim \Sigma^{-1}(\mu_1 - \mu_2)$.

From Discriminant Function to Probabilities

- Once we have estimates $\hat{\delta}_k(x)$, we can turn these into estimates for class probabilities:

$$\hat{P}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{k=1}^{K} e^{\hat{\delta}_k(x)}},$$

- So classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\hat{P}(Y = k|X = x)$ is largest,
- When $K = 2$, we classify to class 2 if $\hat{P}(Y = 2|X = x) \geq 0.5$, else to class 1.

Linear Discriminant Analysis on Default Data

Use Balance and Student to classify the Default (R command).

		True		
		No	*Yes*	*Total*
Predicted	*No*	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

error rate $= \frac{23+252}{10000} = 2.75\%$!

Some caveats!

- This is training error, and we may be overfitting, but it is not a big concern here since $n = 10000$ and $p = 2$!
- If we classified to the prior - always to class *No* in this case - we would make only 3.33% errors! $\frac{252+81}{10000} = 3.33\%$
- Of the true *No*'s, we make $23/9667 = 0.2\%$ errors; of the true *Yes*'s, we make $252/333 = 75.7\%$ errors!

Types of Errors

T = Negative, C = Positive

False positive rate: The fraction of negative examples that are classified as positive - 0.2% in example;

T = Positive, C = Negative

False Negative rate: The fraction of positive examples that are classified as negative - 75.7%.

We produced this table by classifying to class *Yes* if

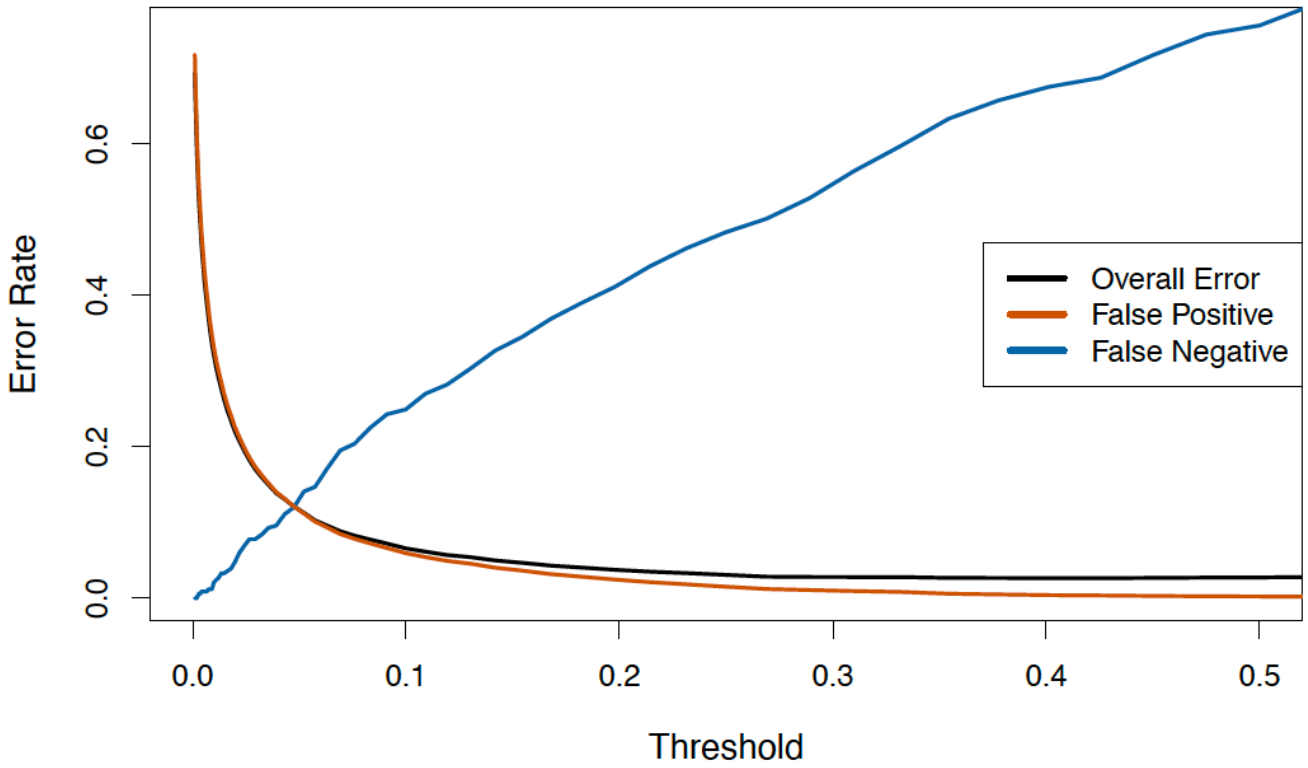
$$\hat{P}(Default = Yes|Balance, Student) \geq 0.5.$$

We can change the two error rates by changing the threshold from 0.5 to some other values in [0, 1]:

$$\hat{P}(Default = Yes|Balance, Student) \geq threshold,$$

and vary the *threshold*!

Varying the Threshold



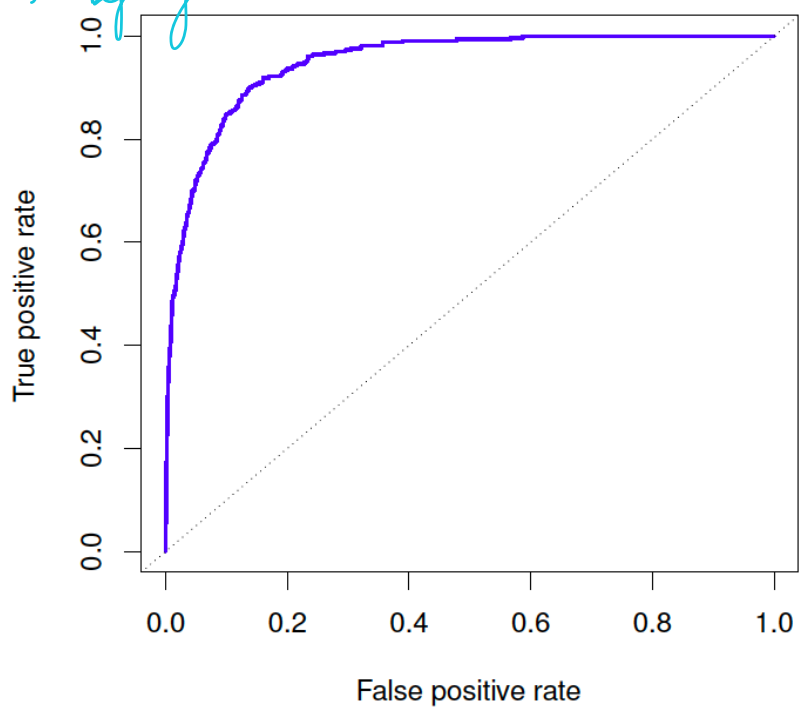
In order to reduce the false negative rate, we may want to reduce threshold to 0.1 or less.

ROC Curve (Receiver Operating Characteristic Curve)

- The ROC plot displays both false positive rate (FPR) and true positive rate (TPR) simultaneously,

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}.$$

		True
Predicted	*No*	*Yes*
No	9644(*TN*)	252(*FN*)
Yes	23(*FP*)	81(*TP*)
Total	9667	333



Other forms of Discriminant Analysis

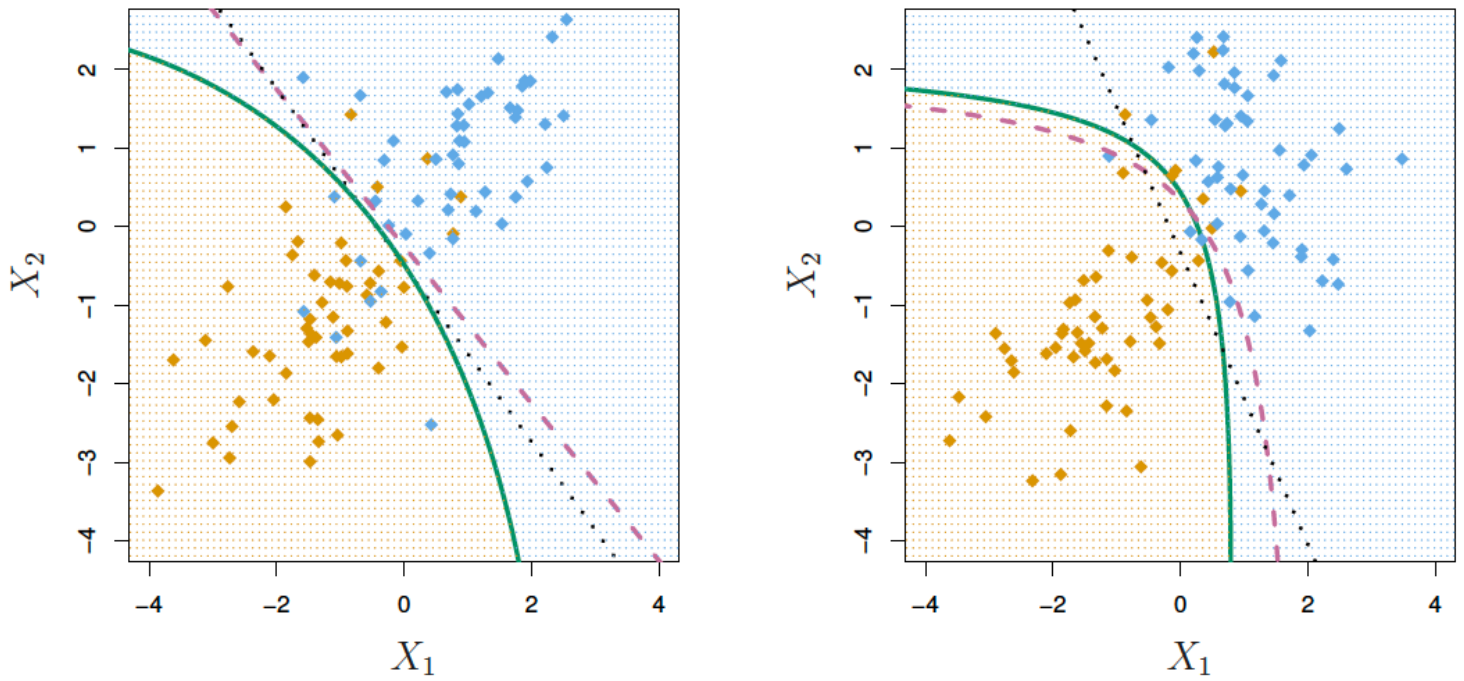
Recall

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{k=1}^{K} \pi_k f_k(x)}.$$

When $f_k(x)$ are Gaussian densities, with the same covariance matrix Σ in each class, this leads to linear discriminant analysis. By altering the forms for $f_k(x)$, we get different classifiers.

- With Gaussians but different Σ_k in each class, we get *quadratic discriminant analysis*,
- With $f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$ (conditional independence model) in each class we get *naive Bayes*. For Gaussian this means the Σ_k are diagonal,
- Many other forms, by proposing specified density models for $f_k(x)$, including nonparametric approaches.

Quadratic Discriminant Analysis



- For the quadratic discriminant analysis, the discriminant function is

$$\delta_k(x) = \left\{ \log(\pi_k) - \frac{1}{2}\log|\Sigma_k| \right\} - \frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k).$$

- Because Σ_k are different, the quadratic terms matter!

Naive Bayes

- Assume features are independent in each class.
- Useful when p is large, and so multivariate methods like QDA and even LDA break down.
- Gaussian naive Bayes assumes each Σ_k are diagonal:

$$\begin{aligned}\delta_k(x) &\approx \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right] \\ &= -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j-\mu_{kj})^2}{\sigma_{kj}^2} + \log(\sigma_{kj}^2)\right] + \log(\pi_k).\end{aligned}$$

- Can use for *mixed* feature vectors. If X_j is qualitative, replace $f_{kj}(x_j)$ with probability mass function (histogram) over discrete categories,
- Despite strong assumption, naive Bayes often produces good classification results.

Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x_1 + \cdots + c_p x_p.$$

So it have the same form as logistic regression. The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on $P(Y|X)$ (known as *discriminative learning*),
- LDA uses the full likelihood based on $P(X, Y)$ (known as *generative learning*),
- Despite these differences, in practice the results are often very similar.

Summary

- Logistic regression is very popular for classification, especially when $K = 2$,
- LDA is useful when n is small, or the classes are well separated, and Gaussian assumption are reasonable. Also when $K > 2$,
- Naive Bayes is useful when p is very large.

The End