

## 4.4.2 Linear Discriminant Analysis for $p > 1$

Dr. Lauren Perry

# Multivariate Normal Distribution

We now assume the predictors  $X = (X_1, X_2, \dots, X_p)$  are drawn from a *multivariate normal distribution*.

$$X \sim N(\mu, \Sigma)$$

- ▶ Each individual predictor follows a one-dimensional normal distribution.
  - ▶ The vector  $\mu$  contains all  $p$  means.
- ▶ Each pair of predictors is allowed to be correlated.
  - ▶ We represent this correlation with a  $p \times p$  covariance matrix  $\Sigma$  that contains each variable's variance and all pairwise covariances.

## LDA for $p > 1$

Assume the observations in the  $k$ th class are drawn from a multivariate normal distribution  $N(\mu_k, \Sigma)$ .

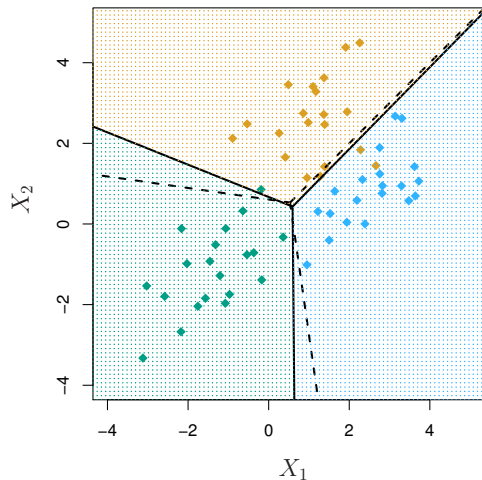
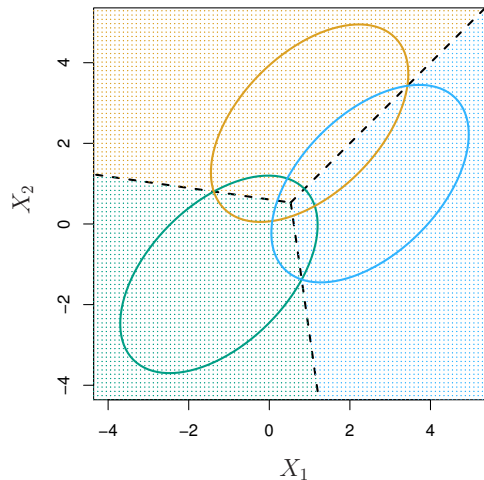
- ▶  $\mu_k$  is a class-specific vector of  $(p)$  means.
- ▶  $\Sigma$  is the covariance matrix, assumed common to all  $K$  classes.

The Bayes classifier assigns an observation  $X = x$  to the class for which

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

where  $\pi_k$  is again  $P(Y = k)$ .

## Example: Simulated Data with $p = 2$



## LDA in Practice

We estimate the unknown parameters  $\mu_1, \dots, \mu_K$ ,  $\pi_1, \dots, \pi_K$ , and  $\Sigma$  similar to how we estimated those used in the one-dimensional case.

The LDA then assigns an observation  $X = x$  to the class for which

$$\hat{\delta}_k(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + \log \hat{\pi}_k$$

is largest.

## Example: Predicting Penguin Species

```
data(penguins, package = "palmerpenguins")
mod1 <- lda(species ~ ., penguins)
predval <- predict(mod1)$class
species <- penguins$species[-unique(which(is.na(penguins), arr.ind=T)[,1])]
table(predval, species)
```

```
##           species
## predval   Adelie Chinstrap Gentoo
##   Adelie      145         0       0
##   Chinstrap    1         68       0
##   Gentoo       0         0      119
```

```
mean(predval == species)
```

```
## [1] 0.996997
```

## Example: Test and Training Data

- ▶ The error rate ( $< 1\%$ ) seems very low, but we're only examining *training* error rate.

```
data(penguins, package = "palmerpenguins")
set.seed(1)

# Remove missing data
pens <- penguins[-unique(which(is.na(penguins), arr.ind=T)[,1]),]
# Use 80% of the data as training data
train.ind <- sample(1:nrow(pens), floor(0.8*nrow(pens)), replace=F)
train.pen <- pens[train.ind, ]
# The rest us test data
test.pen <- pens[-train.ind, ]
```

## Example: Test and Training Data

```
mod2 <- lda(species ~ ., train.pen)
predval <- predict(mod2, test.pen)$class
species <- test.pen$species
table(predval, species)
```

```
##           species
## predval  Adelie Chinstrap Gentoo
##  Adelie      31         1       0
##  Chinstrap   0        10       0
##  Gentoo      0         0      25
```

```
mean(predval == species)
```

```
## [1] 0.9850746
```



## Example: Default Data

```
library(ISLR2)
data(Default)
set.seed(1)

train.ind <- sample(1:nrow(Default), floor(0.8*nrow(Default)), replace=F)
def.train <- Default[train.ind,]
def.test <- Default[-train.ind,]

mod3 <- lda(default ~ ., def.train)
predval <- predict(mod3, def.test)$class
actual <- def.test$default
mean(predval==actual)
```

```
## [1] 0.9705
```

## Example: Default Data Confusion Matrix

```
table(predval, actual)
```

```
##          actual
## predval   No   Yes
##      No 1929   58
##      Yes   1   12
```

- ▶ Overall error is low (approx 3%).
- ▶ But, only 3/3% of those in the training data defaulted, so a model that predicted *no default* would have a very low overall error rate.
- ▶ Also, error rate is very high among people who actually defaulted!
  - ▶ The model correctly identified only 17% (12/70) of the people who defaulted.

## Error Rates

Why does this happen? Consider the two-class case.

- ▶ Bayes classifier - which LDA approximates - has lowest *overall* error rate.
- ▶ The classifier assigns to the posterior probability which is greatest.
- ▶ It will assign to *default* if  $P(\text{default} = \text{Yes} | X = x) > 0.5$ .
  - ▶ But if only 3% of people default, this can be a pretty high threshold to reach!
- ▶ It is also possible to change these assignments, e.g., assign to *default* if  $P(\text{default} = \text{Yes} | X = x) > 0.2$ .
  - ▶ ...but this will come with a trade off in accuracy of assigning people to *not default*.

# Bayes Theorem

In general, Bayes theorem states

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

► Suppose, for example, that  $\pi_1 = P(A) = 0.01$ .

Then Bayes theorem looks something like

$$P(A|B) = \frac{P(B|A) \times 0.01}{P(B)}$$

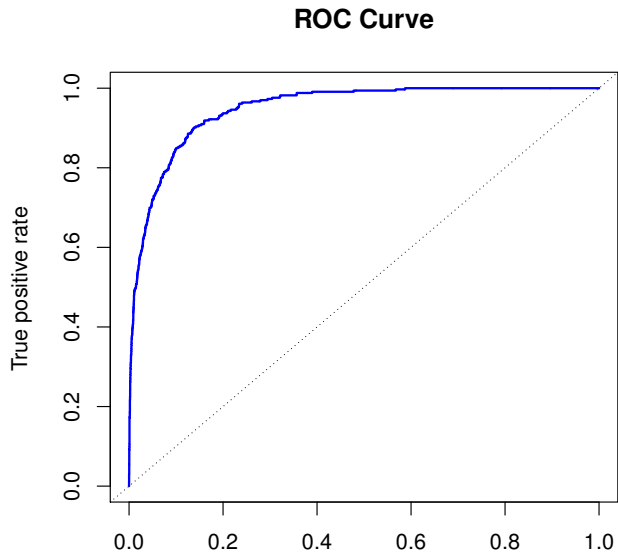
Getting  $P(A|B)$  over 0.5 is going to be difficult in this scenario!

# ROC Curves

Receiver Operating Characteristics curves display the relationship between false positive rate and true positive rate, which vary with different probability thresholds.

- ▶ Classifier performance over all possible thresholds can be summarized by ROC area under the curve (AUC).
- ▶ Ideal ROC curves hug the top left corner.
- ▶ The true positive rate is referred to as *sensitivity*.
- ▶ The false positive rate is  $1 - \text{specificity}$ .
  - ▶ (I.e., specificity is the true negative rate.)

## ROC Curves



# Model Performance and Misclassification

		True class		
		– or Null	+ or Non-null	Total
Predicted class	– or Null	True Neg. (TN)	False Neg. (FN)	N*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P*
	Total	N	P	

**TABLE 4.6.** Possible results when applying a classifier or diagnostic test to a population.

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1–Specificity
True Pos. rate	TP/P	1–Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1–false discovery proportion
Neg. Pred. value	TN/N*	

**TABLE 4.7.** Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.