4.4 Generative Models for Classification

Dr. Lauren Perry

The Need for Alternatives

Why not just use logistic regression?

- If there is a lot of separation between the classes, logistic regression models are surprisingly unstable.
  - (Coefficient estimates can vary significantly given the same data generating process.)
- If the distribution of the predictors X is approx. normal and the sample size is small, these alternatives may be more accurate than logistic regression.
- The methods in this section have more natural extensions to three or more classes.

Idea

- Model the distribution of the predictors X separately for each response class.
- Use Bayes' theorem to work these into estimates for $\mathrm{P}(Y = k|X = x)$.

The Setup

Suppose Y can take on K distinct, unordered values.

- Let π_k represent the overall probability that a randomly chosen observation comes from the kth class.
  - Generally estimated as the proportion of training observations belonging to class k.
- Let $f_k(x) = \mathrm{P}(X|Y = k)$ denote the density function of X for an observation from the kth class.
  - So $f_k(x)$ should be relatively large if there is a high probability that an observation from the kth class has $X \approx x$.
- Then Bayes' Theorem states

$$p_k(x) = \mathrm{P}(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

  - This is the *posterior probability* that an observation belongs to the kth class, given $X = x$.

Goal: estimate $f_k(x)$ to approximate the Bayes' classifier $p_k(x)$.

4.4.1 Linear Discriminant Analysis for One Predictor

We will classify an observation into the category for which $p_k(x) = \mathrm{P}(Y = k|X = x)$ is greatest.

- Assume $f_k(x)$ is normally distributed (Gaussian):

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

  where μ_k and σ_k^2 are the mean and standard deviation parameters for the kth class.
- Also assume $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$ (shared variance term for all classes).

Linear Discriminant Analysis for One Predictor, $p = 1$

Combining the Bayes' Theorem set up with these assumptions, we get

$$p_k(x) = \frac{\frac{\pi_k}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu_k)^2\right]}{\sum_{l=1}^{K} \frac{\pi_l}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu_l)^2\right]}$$

which looks a mess, but it can be shown this is equivalent to assigning the observation to the class for which

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

is largest.

Linear Discriminant Analysis for One Predictor, $p = 1$

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

If $K = 2$ and $\pi_1 = \pi_2$, this classifier assigns an observation to

- class 1 if $2x(\mu_1 - \mu_2) > \mu_1 - \mu_2$.
- class 2 *otherwise*.

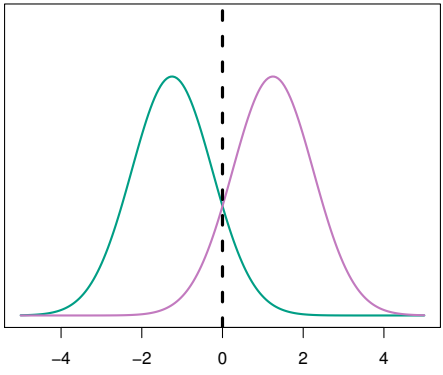
The Bayes' decision boundary is the point for which $\delta_1 = \delta_2$, which in this setting is

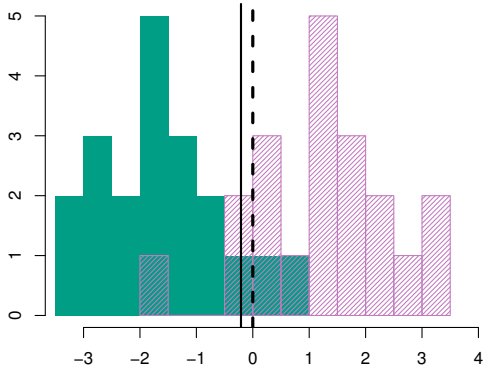
$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

Example

- Consider predictors generated from two normal distributions where $\mu_1 = -1.25$, $\mu_2 = 1.25$, and $\sigma_1 = \sigma_2 = 1$.
- Assume an observation is equally likely to come from either class, i.e., $\pi_1 = \pi_2 = 0.5$.
- Then the (known) Bayes' classifier assigns an observation to class 1 if $x < 0$ and class 2 otherwise.

Example





- 20 observations drawn from each class.
- LDA decision boundary shown as solid vertical line.

Linear Discriminant Analysis for One Predictor, $p = 1$

In practice, we must estimate $\mu_1, \ldots, \mu_K$, $\pi_1, \ldots, \pi_K$, and σ.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{\pi}_k = \frac{n_k}{n}$$

Where n is the number of training observations and n_k is the number of training observations in the kth class.

- $\hat{\sigma}^2$ is a weighted average of sample variances across the K classes.

Linear Discriminant Analysis for One Predictor, $p = 1$

Assign an observation $X = x$ to the class for which

$$\delta_k(x) = x \left(\frac{\hat{\mu}_k}{\hat{\sigma}^2} \right) - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

is largest.

Example: Using Penguin Body Mass to Predict Species

```
data(penguins, package = "palmerpenguins")
mod1 <- lda(species ~ body_mass_g, penguins)
predval <- predict(mod1)$class
species <- penguins$species[!is.na(penguins$species) & !is.na(penguins$body
table(predval, species)
```

```
##             species
## predval      Adelie Chinstrap Gentoo
##   Adelie        140        64     14
##   Chinstrap       0         0      0
##   Gentoo         11         4    109
```

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

```
## [1] 0.7280702
```

4.4.2 Linear Discriminant Analysis for $p > 1$

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

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

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

LDA for $p > 1$

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

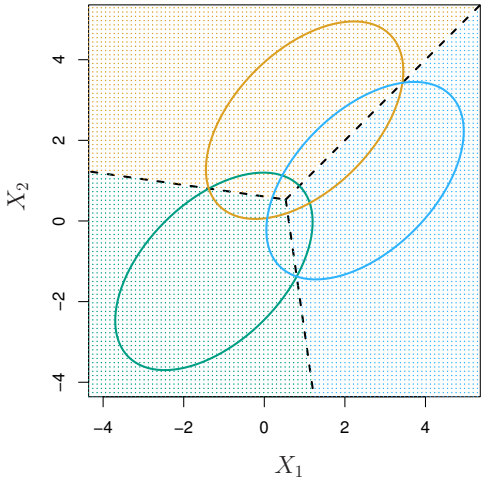
- μ_k is a class-specific vector of (p) means.
- Σ is the covariance matrix, assumed common to all K classes.

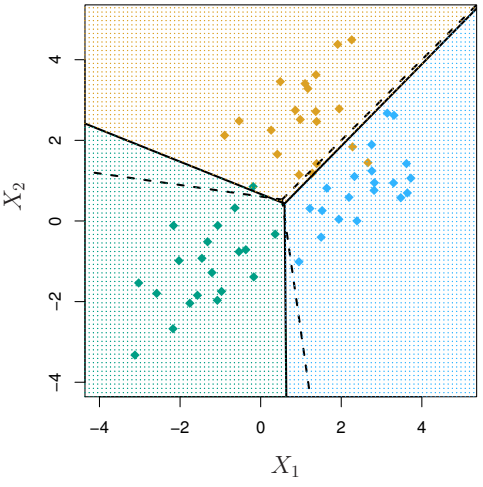
The Bayes classifier assigns an observation $X = x$ to the class for which

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

where π_k is again $P(Y = k)$.

Example: Simulated Data with $p = 2$





LDA in Practice

We estimate the unknown parameters $\mu_1, \ldots, \mu_K$, $\pi_1, \ldots, \pi_K$, and Σ 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

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

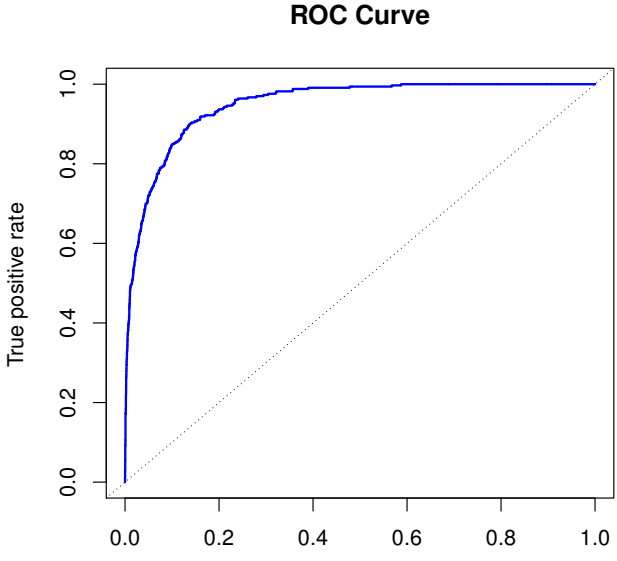
Getting $P(A|B)$ over 0.5 is going to be difficult in this scenario!

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 $1-$ *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.*

4.4.3 Quadrative Discriminant Analysis

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

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

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

QDA vs LDA

- Quadratic discriminant analysis is similar to linear discriminant analysis.
  - We assume predictors from the kth class are of the form

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

- However, QDA allows each class to have its own covariance matrix.
  - Now, assume predictors from the kth class are of the form

4.4.4 Naive Bayes

Recall: Bayes' theorem gives us the expression

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

and we need to estimate $\pi_1, \ldots, \pi_K$ and $f_1(x), \ldots f_K(x)$.

Bayes Classifier

- In practice, estimating $\pi_1, \ldots, \pi_K$ is fairly simple.
- Estimating $f_1(x), \ldots f_K(x)$ is relatively more involved.
  - We can simplify this with strong assumptions about the distribution, e.g., multivariate normal.
    - In this case, we need only to estimate the distributions *parameters*.
  - Naive Bayes' takes a different approach.

Naive Bayes

Assumption: within the kth class, the p predictors are independent.
Mathematically, this means we can write

$$f_k(x) = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)$$