Section 8: Classification with R

STA 141A – Fundamentals of Statistical Data Science

Instructor: Akira Horiguchi

Fall Quarter 2025 (Sep 24 – Dec 12)
MWF, 9:00 AM – 9:50 AM, TLC 1215
University of California, Davis

Overview

Based on Chapter 4 of ISL book James et al. (2021).

- For more R code examples, see R Markdown files in `https://www.statlearning.com/resources-second-edition`

1. Why not linear regression?

2. Logistic regression
   - Binary classification
   - Multinomial logistic regression

3. Alternatives to logistic regression
   - Linear discriminant analysis for $p = 1$
   - Naive Bayes

4. Errors in classification

5. Comparison of classification methods

Example (two categories)

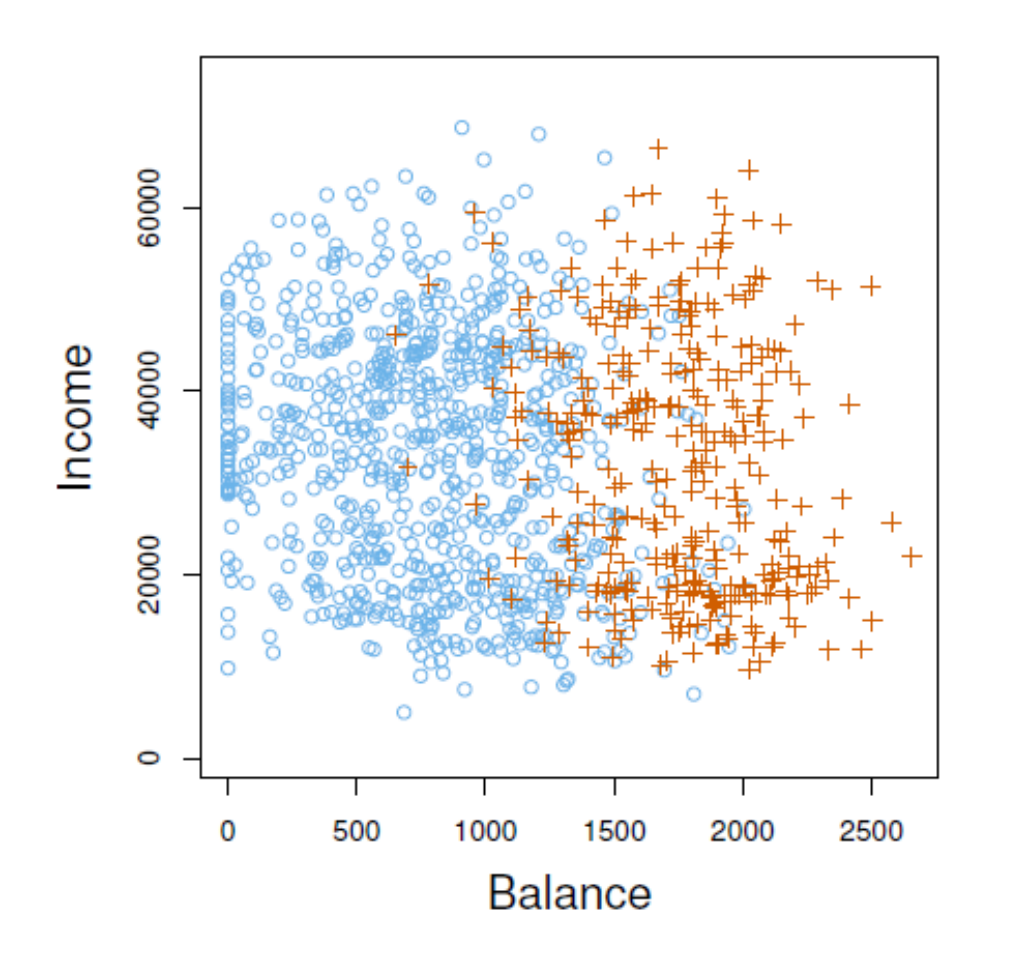


Figure 1: Image by James et al. (2021), based on the `Default` data set in R. The annual incomes and monthly credit card balances of a number of individuals, where the individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.

Examples

What are the predictors and responses in each example?

1. A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of these medical conditions does the person have based on the symptoms given?

2. An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.

3. On the basis of DNA sequence data for a number of patients with and without a given disease, one would like to figure out which DNA mutations are disease-causing and which are not.

The concept

Classification: the task of predicting *qualitative*/*categorical* responses

- Each response y_i is one of finitely many predetermined categories.
- *Classifying* an observation: assigning/predicting that observation to a certain category/class.
- In contrast, regression deals with "continuous" numeric response values.

As in regression, in the classification setting

- We have a set of training observations $(x_1, y_1), \ldots, (x_n, y_n)$ that we can use to build a classifier.
- We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

Why not linear regression?

No natural ordering

In example 1 above, a person arrives at the emergency room with a set of symptoms. We would like to treat the person based on three reasonable medical conditions: `Appendicitis`, `Food poisoning`, `Gastritis`.

- We could code each medical condition Y as:

$$Y = \begin{cases} 1, & \text{if } \texttt{Appendicitis}\,, \\ 2, & \text{if } \texttt{Food poisoning}, \\ 3, & \text{if } \texttt{Gastritis}. \end{cases}$$

  This coding implies an ordering on the outcomes, insisting that the difference between `Appendicitis` and `Food poisoning` is the same as the difference between `Food poisoning` and `Gastritis`.

- We could also code:

$$Y = \begin{cases} 1, & \text{if } \texttt{Gastritis}, \\ 2, & \text{if } \texttt{Appendicitis}, \\ 3, & \text{if } \texttt{Food poisoning}. \end{cases}$$

  Equally reasonable, but would lead to very different predictions on test observations.

Natural ordering

What if categories had a natural ordering, such as `mild`, `moderate`, and `severe`?

- Issue: the distance between *ordinal* categories is generally unknown.
- In general there is no natural way to convert a qualitative response variable with *more than two levels* into a quantitative response that is ready for linear regression.

Only two levels

Can we use linear regression for a *binary* (two levels) response?

- In the Default data set, the two response values can be coded as

$$Y = \begin{cases} 1, & \text{if } \texttt{Default}, \\ 0, & \text{if } \texttt{Not default}. \end{cases}$$

- We could then fit a linear regression to this binary response:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{Balance} + \hat{\beta}_2 \times \texttt{Income}$$

  and then predict Default if $\hat{Y} > 0.5$ and Not default otherwise.
- What if we also want to estimate e.g.,

$$P\big(\texttt{Default} \mid \texttt{Balance} = 4000, \texttt{Income} = 80000\big)$$

  i.e., the probability of defaulting given certain values of Balance and Income?
- Issue: $\hat{Y}$ can be smaller than zero or larger than one.

Only two levels

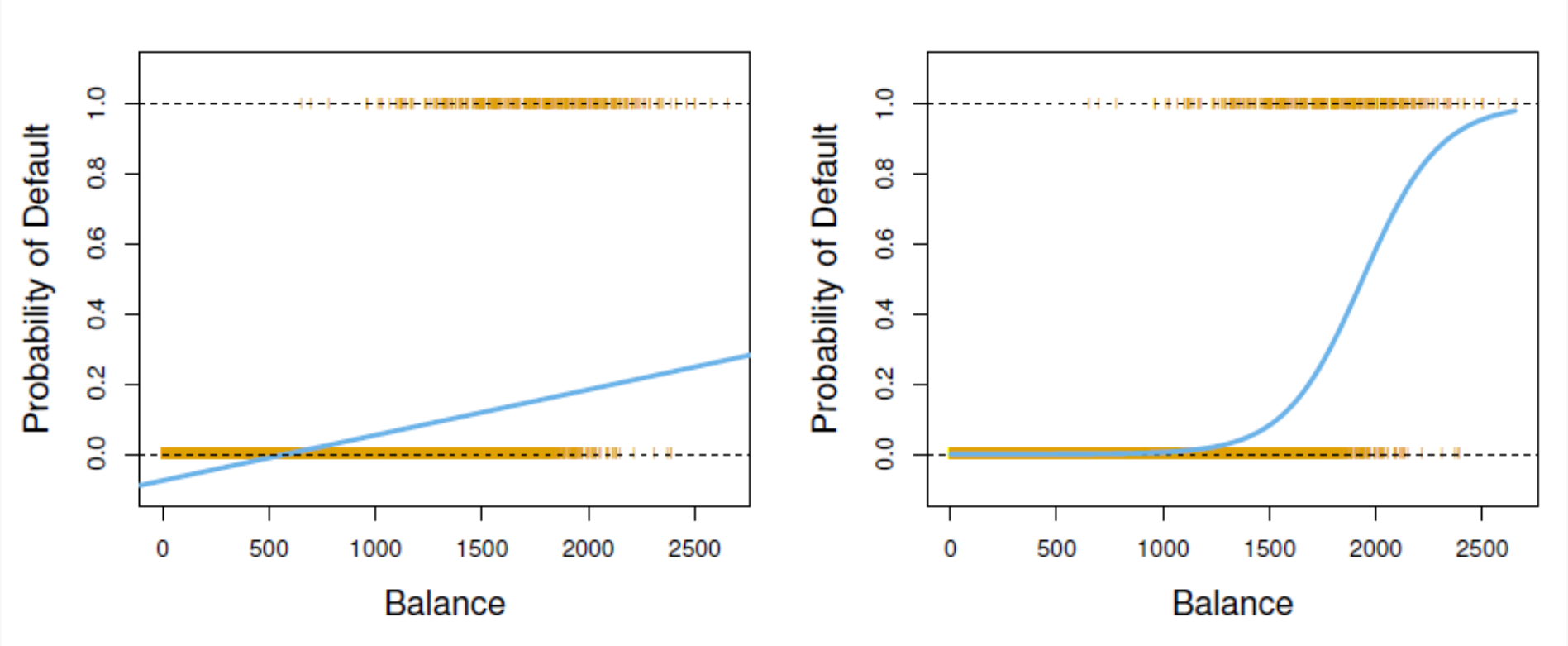


Figure 2: Image by James et al. (2021), based on the `Default` data set in R. Left: The estimated probability of default using *linear regression*, where the orange ticks indicate the values "0" for `No`, and "1" for `Yes`. Right: Predicted probabilities of default using *logistic regression*, where all probabilities lie between 0 and 1.

- Let's explore logistic regression.

Log odds

Let $P(A)$ be the *probability* that event A occurs. Then $P(A) \in [0, 1]$. Map to $(-\infty, \infty)$?

- The *log odds* of A occurring is defined as

$$\log\left(\frac{P(A)}{1 - P(A)}\right) \tag{1}$$

which can be a value in $\mathbb{R}$. (For this course, assume $\log$ is the natural logarithm, i.e., log with base e.) We will also write (1) as $\mathit{logit}\Big(P(A)\Big)$.

Logistic regression

Logistic regression

Binary classification

Binary classification

Each response belongs to one of two classes, coded as 0 and 1 (e.g., `No` and `Yes`).

- Classification: compute/estimate conditional prob. $P(Y = k|X)$ for each class k.
- If only two classes, we only need $P(Y = 1|X)$. (Why?)

$P(Y=0|X) = 1 - P(Y=1|X)$

Suppose we have computed $P(Y = 1|X)$ for a given value of predictor X. What class should be assigned to X?

- A default decision rule for predictor value X is to assign:

$$\begin{cases} 1 & \text{if } P(Y = 1|X) > 0.5; \\ 0 & \text{if } P(Y = 1|X) \leq 0.5. \end{cases}$$

- Consequences might not be symmetric. E.g., in court, is it worse give a `guilty` verdict to an innocent person, or give a `not guilty` verdict to guilty person? May want to change the decision rule to assign:

$$\begin{cases} \texttt{guilty} & \text{if } P(Y = 1|X) > 0.8; \\ \texttt{not guilty} & \text{if } P(Y = 1|X) \leq 0.8. \end{cases}$$

Logistic regression

Logistic regression models the conditional probability $P(Y = 1|X)$.

- Convert $p(X) = P(Y = 1|X)$ to log odds, then use linear regression on log odds:

$$\text{logit}\Big(p(X)\Big) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

- The conditional probabilities are then

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}. \tag{2}$$

Interpretation:

- Increasing X_1 by one unit changes $p(X)$ by

$$p(X_1 + 1, X_2, X_3, \ldots, X_p) \; - \; p(X_1, X_2, X_3, \ldots, X_p)$$

  which depends on all $p - 1$ coefficient values and the current predictor values.

- Increasing X_1 by one unit changes the log odds $\text{logit}\Big(p(X)\Big)$ by

$$\text{logit}\Big(p(X_1 + 1, X_2, X_3, \ldots, X_p)\Big) - \text{logit}\Big(p(X_1, X_2, X_3, \ldots, X_p)\Big)$$

$$[\beta_0 + \beta_1(X_1+1) + \beta_2 X_2 \cdots + \beta_p X_p] - [\beta_0 + \beta_1 X_1 + \beta_2 X_2 \cdots + \beta_p X_p]$$

  which is just β_1.

Logistic regression

Estimating the regression coefficients:

- Usually use the method of *maximum likelihood*.
- Details outside scope of this class; we will just use R to compute these estimates.

Making predictions from estimators $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$

To assign 0 or 1 to X, we can use estimated log odds or estimated $p(X)$.

- We can estimate log odds at X by

$$\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p. \tag{3}$$

- Alternatively, we can estimate the conditional probability $p(X) = P(Y = 1|X)$ by

$$\hat{p}(X) := \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p}}.$$

- If this estimated probability is over a certain pre-defined threshold (e.g. 0.25), then we would assign $X = x$ to class 1.

Example

If $\hat{\beta}_0 = -9.9$ and $\hat{\beta}_1 = 0.005$, we predict the probabilities of `default` for individuals with balance $X = \$1,000$ and $X = \$2,000$ by

$$\hat{p}(X = 1,000) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-9.9 + 0.005 \cdot 1,000}}{1 + e^{-9.9 + 0.005 \cdot 1,000}} \approx 0.007, \quad < 0.25$$

$$\hat{p}(X = 2,000) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-9.9 + 0.005 \cdot 2,000}}{1 + e^{-9.9 + 0.005 \cdot 2,000}} \approx 0.525. \quad > 0.25$$

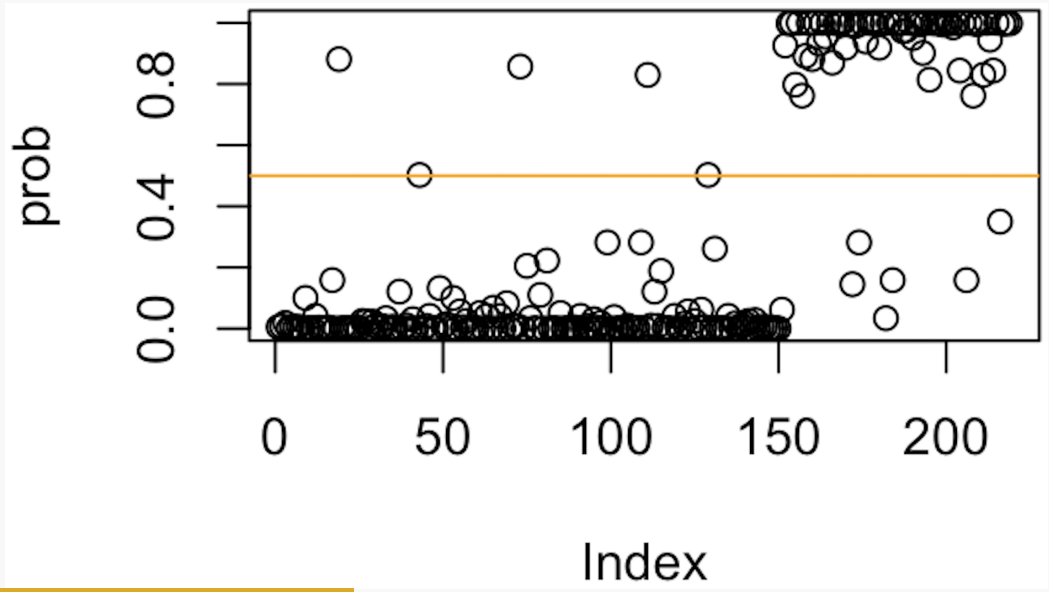
glm( )

We use glm( ) for logistic regression ('glm' stands for *general linear model*).

- Must specify which variables are used, data set, and type of response.
- Must put family=binomial to specify a binary response.

```
library(palmerpenguins)
# We work with only Adelie and Chinstrap species (we exclude Gentoo).
peng_binary <- na.omit(penguins[penguins$species != 'Gentoo', ])
logreg <- glm(species ~ bill_length_mm, data=peng_binary, family=binomial)
prob <- predict(logreg, type='response')  # 'link' also possible
predicted <- ifelse(prob<.5, 'Adelie', 'Chinstrap')

plot(prob)
abline(a=0.5, b=0, col='orange')
```



Logistic regression

Multinomial logistic regression

Multinomial logistic regression

We sometimes wish to classify *a response variable that has more than two classes.*

- Extend the two-class logistic regression approach to the setting of $K > 2$ classes.
- We will need separate regression coefficients for each of the first $K - 1$ classes. Hence, for any $x \in \mathbb{R}^p$, define

$$\alpha_l(x) := \beta_{l,0} + \beta_{l,1}x_1 + \cdots + \beta_{l,p}x_p \quad \text{for any } l = 1, \ldots, K-1. \tag{4}$$

E.g., consider conditions `Appendicitis`, `Food poisoning`, and `Gastritis`. For $j = 1, \ldots, p$, consider $x_j =$`Severity of symptom` j (e.g. how much does head hurt? how nauseated?).

- For `Appendicitis`, we want to define $\beta_{\texttt{Appendicitis},j}$ for $j = 0, 1, \ldots, p$.
- For `Food poisoning`, we want to define $\beta_{\texttt{Food poisoning},j}$ for $j = 0, 1, \ldots, p$.
- We could define similarly for `Gastritis`, but we will see that we won't need to.

Multinomial logistic regression

For convenience, copy-and-paste Eq. (4) here:

$$\alpha_l(x) := \beta_{l,0} + \beta_{l,1}x_1 + \cdots + \beta_{l,p}x_p \qquad \text{for any } l = 1, \ldots, K-1.$$

Multinomial logistic regression model: without loss of generality

1. Select a class to serve as the baseline; WLOG, select the Kth class for this role.
2. Replace the model (2) with the model

$$P(Y = k \mid X = x) = \begin{cases} \dfrac{e^{\alpha_k(x)}}{1+\sum_{l=1}^{K-1} e^{\alpha_l(x)}} & \text{for } k = 1, \ldots, K-1, \\ \dfrac{1}{1+\sum_{l=1}^{K-1} e^{\alpha_l(x)}} & \text{for } k = K. \end{cases}$$

For $k = 1, \ldots, K-1$, we have

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = K \mid X = x)}\right) = \alpha_k(x)$$

which is linear in the predictors.

Interpretation

Consider classifying ER visits into `Appendicitis`, `Food poisoning`, `Gastritis`.

- Suppose we set `Appendicitis` as the baseline.
- If X_j increases by one unit, then

$$\log\left(\frac{P(Y = \texttt{Food poisoning} \mid X = x)}{P(Y = \texttt{Appendicitis} \mid X = x)}\right)$$

  increases by $\beta_{\texttt{Food poisoning},j}$.
- If X_j increases by one unit, then

$$P(Y = \texttt{Food poisoning} \mid X = x)$$

  increases by a complicated function of all $p - 1$ coefficient values and the current predictor values.

Code walkthrough

ISLR2 textbook doesn't have code walkthrough for multinomial logistic regression, so you can find one here:
`https://www.r-bloggers.com/2020/05/multinomial-logistic-regression-with-r/`

Alternative coding: softmax coding

In the *softmax coding* (used extensively in some areas of machine learning), rather than selecting a baseline class, we treat all K classes symmetrically:

$$P(Y = k \mid X = x) = \frac{e^{\alpha_k(x)}}{\sum_{l=1}^{K} e^{\alpha_l(x)}} \qquad \text{for } k = 1, \ldots, K$$

- Thus, we estimate coefficients for all K classes (rather than for just $K - 1$ classes).
- The log odds ratio between the kth and lth classes equals

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = l \mid X = x)}\right) = \alpha_k(x) - \alpha_l(x)$$
$$= (\beta_{k,0} - \beta_{l,0}) + (\beta_{k,1} - \beta_{l,1})x_1 + \cdots + (\beta_{k,p} - \beta_{l,p})x_p.$$

Example interpretation: if X_j increases by one unit, then

$$\log\left(\frac{P(Y = \texttt{Food poisoning} \mid X = x)}{P(Y = \texttt{Appendicitis} \mid X = x)}\right)$$

increases by $(\beta_{\texttt{Food poisoning},j} - \beta_{\texttt{Appendicitis},j})$.

End of 11/10 lecture

Alternatives to logistic regression

Motivation

Recall: logistic regression directly models $P(Y = k|X = x)$ for binary responses by using the logistic function.

Pros:

- We can see impact of e.g., a unit increase in X_j on log odds.
- Assumptions are relatively loose: independence of errors, a linear relationship between the logit and independent variables, and absence of severe multicollinearity among independent variables.

Some issues:

- When there are big differences between two classes, the parameter estimates in the logistic regression model are unstable.
- If the sample size is small, other approaches can be more accurate than logistic regression.

Using Bayes' theorem

We can instead use *Bayes' theorem* to obtain the *posterior* probability $P(Y = k \mid X = x)$:

$$p_k(x) := P(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}, \tag{5}$$

- $\pi_k := P(Y = k)$ is the overall or *prior* probability that a randomly chosen observation comes from the kth class.
  - Here π is just a variable name — not the same as $\pi = 3.14159\ldots$
- f_k is the predictor's PMF/PDF given that the response is from the kth class.
  - PMF case: $f_k(x) = P(X = x \mid Y = k)$.

We typically don't know either π_k or f_k.

- Can estimate π_k by the proportion of observed elements in the kth class.
  - E.g. if there are 3, 2, 5 elements in the classes 1, 2, 3, respectively, then the estimated probabilities are $\hat{\pi}_1 = \frac{3}{10}, \hat{\pi}_2 = \frac{2}{10}, \hat{\pi}_3 = \frac{5}{10}$.
- Estimating f_k is more challenging — typically requires a huge amount of data unless *strong simplifying assumptions* are made.
  - Bias-variance tradeoff: the assumptions will reduce variance (hopefully by a lot) at the cost of introducing some bias.
  - Two approaches will be discussed in the next few slides.

Alternatives to logistic regression

Linear discriminant analysis for $p = 1$

Assumptions of f_k in LDA for $p = 1$

The *linear discriminant analysis (LDA)* approach estimates f_k by assuming:

1. f_k is a normal/Gaussian PDF, i.e. for all x holds

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right\}. \tag{6}$$

   - This class will focus on $p = 1$.
   - But if $p > 1$, replace μ_k with p-tuple and replace σ_k with $p \times p$ covariance matrix Σ_k.

2. Same variance parameter across all K classes: $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$.
   - The *quadratic discriminant analysis (QDA)* relaxes this *equivariance* assumption, but we won't discuss QDA in this class.

Bayes decision boundary

With these assumptions, we plug the PDF (6) into the posterior probability (5) to get

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu_k)^2\right\}}{\sum_{\ell=1}^{K} \pi_\ell \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu_\ell)^2\right\}}. \tag{7}$$

- Recall: *Bayes classifier* assigns observation with predictor x to class

$$\underset{k\in\{1,2,\dots,K\}}{\arg\max}\ p_k(x) = \underset{k}{\arg\max}\ \log\left\{p_k(x)\right\} = \underset{k}{\arg\max}\ \log\left\{\pi_k \exp\left\{-\frac{1}{2\sigma^2}(x-\mu_k)^2\right\}\right\} =$$

- This class is equivalent (take the log of (7)) to the class $\underset{k\in\{1,2,\dots,K\}}{\arg\max}\ \delta_k(x)$ where

$$\delta_k(x) := x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k). \tag{8}$$

  Here δ_k is called a *discriminant function*.
- If $K = 2$, classifier assigns x to class 1 if $\delta_1(x) > \delta_2(x)$, to class 2 otherwise.
- If $p = 1$, the *Bayes decision boundary* is the value x for which $\delta_1(x) = \delta_2(x)$
  - What does this inequality $\delta_1(x) > \delta_2(x)$ and boundary simplify to if $\pi_1 = \pi_2$?

$$\delta_1(x) > \delta_2(x)$$

$$\Leftrightarrow\ x\frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + \log(\pi_1) > x\frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} + \log(\pi_2)$$

$$\Leftrightarrow\ x > \frac{\mu_1+\mu_2}{2}$$

boundary is $x = \frac{\mu_1+\mu_2}{2}$

Example for decision boundary

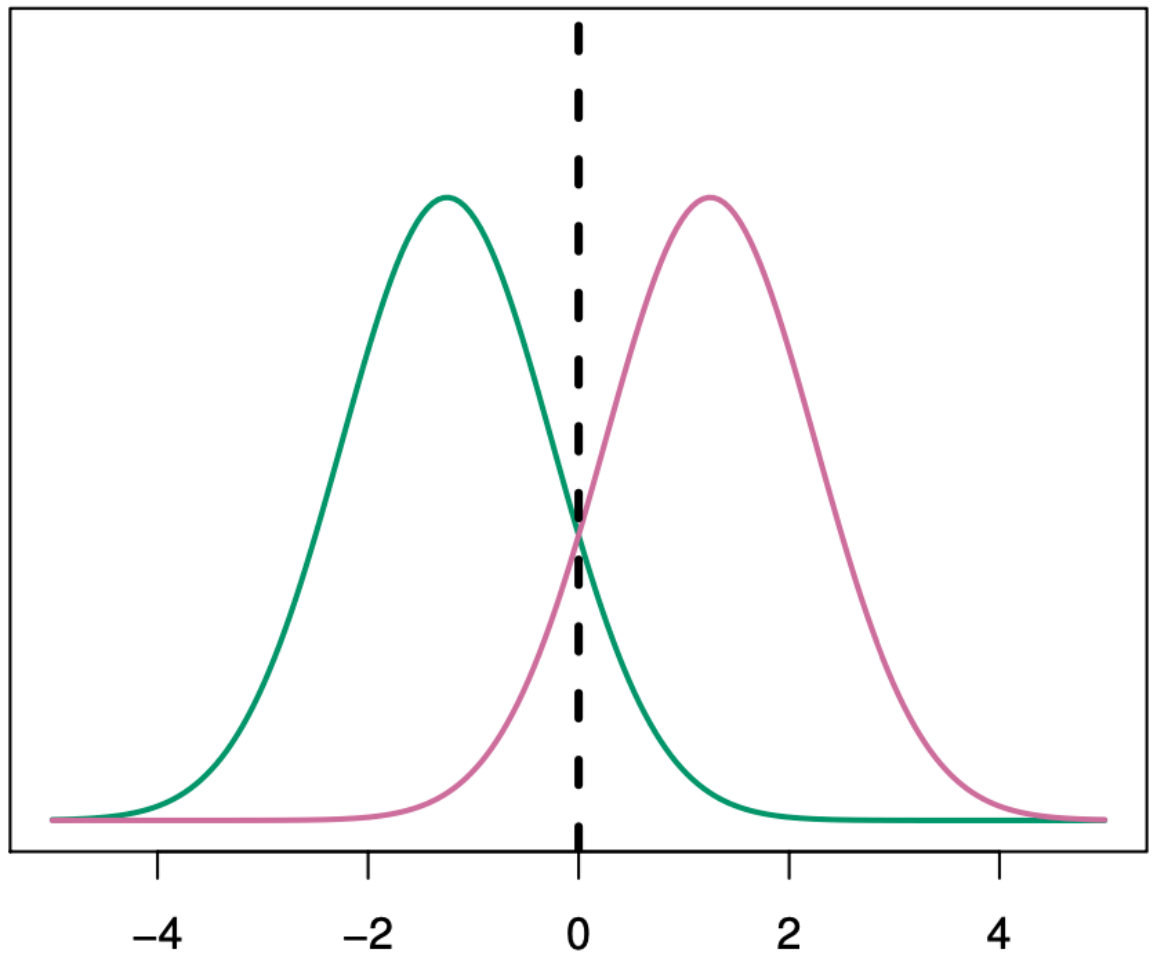


Figure 3: Image by James et al. (2021). Two PDFs of normal distributions with means $\mu_1 = -1.25$ and $\mu_2 = 1.25$, respectively, and variance $\sigma^2 = 1$. The dashed vertical line represents the Bayes decision boundary, so we assign the observation to class 1 if x is left of the line, and to class 2 otherwise.

The LDA method for $p = 1$

In the plot above, we can calculate the Bayes classifier because we know values for all parameters $\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \sigma^2$.

- In practice, we must estimate these parameters to apply the Bayes classifier.
- Consider the estimates

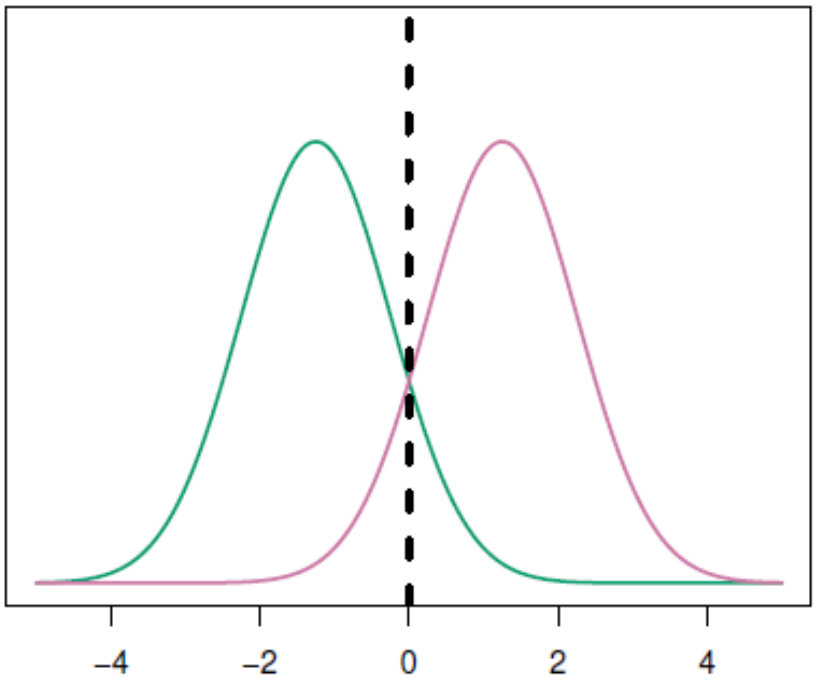
$$\hat{\pi}_k = \frac{n_k}{n}, \qquad \hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i, \qquad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2,$$

  - $\hat{\pi}_k$ — proportion of all training observations from kth class.
  - $\hat{\mu}_k$ — average of all training observations from kth class.
  - $\hat{\sigma}^2$ — weighted average of sample variances for each class.
- Plug these estimates into (8) to get the *LDA discriminant function*

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k), \tag{9}$$

  which is linear in x, hence the "linear" in LDA.

Example for decision boundary



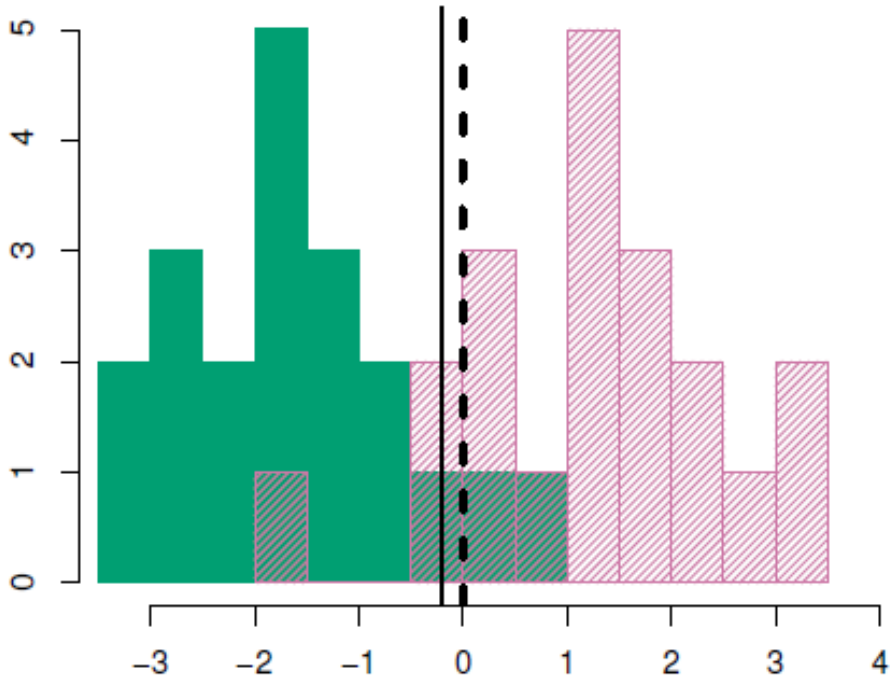


Figure 4: Image by James et al. (2021). Left: Two PDFs of normal distributions with means $\mu_1 = -1.25$ and $\mu_2 = 1.25$, respectively, and variance $\sigma^2 = 1$. The dashed vertical line represents the Bayes decision boundary, so we assign the observation to class 1 if $x < 0$ and class 2 otherwise. Right: 20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is shown as a dashed vertical line, and the solid vertical line represents the LDA decision boundary estimated from the training data.

Calculation example

For $K = 2$ classes (class "0" and "1"), we have the five data points

$$(9, 1), (8, 0), (6, 0), (7, 1), (4, 0)$$

(x_1, y_1) (x_2, y_2) ... (x_5, y_5)

and want to calculate the LDA discrimination function (9) for $k = 0$ and $k = 1$.

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

$\hat{\pi}_0 = \frac{n_0}{n} = \frac{3}{5}$ $\hat{\pi}_1 = \frac{n_1}{n} = \frac{2}{5}$ $\hat{\mu}_0 = \frac{1}{3}[8+6+4] = 6$ $\hat{\mu}_1 = \frac{1}{2}[9+7] = 8$

$$\hat{\sigma}^2 = \frac{1}{3}\left[\underbrace{\{(8-6)^2 + (6-6)^2 + (4-6)^2\}}_{\text{class } 0} + \underbrace{\{(9-8)^2 + (7-8)^2\}}_{\text{class } 1}\right] = \frac{1}{3}[8+2] = \frac{10}{3}$$

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k).$$

$$\hat{\delta}_0(x) = x \cdot \frac{6}{10/3} - \frac{6^2}{2 \cdot 10/3} + \log\left(\frac{3}{5}\right)$$

$$\hat{\delta}_1(x) = x \cdot \frac{8}{10/3} - \frac{8^2}{2 \cdot 10/3} + \log\left(\frac{2}{5}\right)$$

End of 11/12 lecture

Alternatives to logistic regression

Naive Bayes

Naive Bayes classifier

The *naive Bayes classifier* assumes for each class $k = 1, \ldots, K$ that:

Within the kth class, the p predictors are independent.

Mathematically, this means that for each class k:

$$f_k(x) = f_{k,1}(x_1) \times f_{k,2}(x_2) \times \cdots \times f_{k,p}(x_p) \tag{10}$$

where f_{kj} is the PDF/PMF of the jth predictor for observations in the kth class.

- Plug (10) into (5) to get the posterior probability

$$p_k(x) = P(Y = k|X = x) = \frac{f_{k,1}(x_1) \cdot f_{k,2}(x_2) \cdots f_{k,p}(x_p) \cdot \pi_k}{\sum_{\ell=1}^{K} f_{\ell,1}(x_1) \cdot f_{\ell,2}(x_2) \cdots f_{\ell,p}(x_p) \cdot \pi_\ell}. \tag{11}$$

- The *naive Bayes classifier* will assign an observation with predictor x to the class that maximizes posterior probability (11).
- The independence assumption, though often unrealistic, produces decent results, especially when n is too small to effectively estimate the joint distribution f_k.

Estimation approach

Estimating the posterior probability (11) requires estimating the univariate density functions $f_{k,j}$ for all classes $k = 1, \ldots, K$ and all predictors $j = 1, \ldots, p$. Some options:

- If X_j is quantitative, we can assume $X_j | Y = k \sim \mathcal{N}(\mu_{k,j}, \sigma^2_{k,j})$ (as in LDA) or use a nonparametric approach.
- If X_j is qualitative, we could count the proportion of training observations for the jth predictor corresponding to each class k.
  - E.g. suppose we want to predict whether a student studies more than 10 hours per week based on their major ([redacted]). We survey 100 people:

	Math major	Art major	Poli Sci major	
Study $>$ 10 hr /wk	20	25	5	20+25+5 = 50
Study $\leq$ 10 hr /wk	15	25	10	15+25+10 = 50

We use proportions (i.e., divide each cell by row sum) to estimate the "true" PMFs $f_{k,j}$ (each ~~column~~ row is a PMF):

	Math major	Art major	Poli Sci major
Study $>$ 10 hr /wk	0.4 [redacted] 20/50	0.5 [redacted] 25/50	0.1 [redacted] 5/50
Study $\leq$ 10 hr /wk	0.3 [redacted] 15/50	0.5 [redacted] 25/50	0.2 [redacted] 10/50

Also see Fig 4.10 in ISLR2 textbook

Errors in classification

Confusion matrix

In classification, observations can be assigned to the wrong class.

- In binary classification, two mistakes are: *false positives* and *false negatives*.
- Examples: not default vs default, cancer vs no cancer, spam vs not spam.
- A *confusion matrix* displays both error types.

		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	

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*	

Figure 5: Tables by James et al. (2021). A confusion matrix compares the LDA predictions to the true default statuses for the 10,000 training observations in the `Default` data set, using a modified threshold value that predicts default for any individuals whose posterior default probability exceeds 20 %.

Confusion matrix

```
# Using peng_binary and predicted from earlier slide
pb_species <- factor(peng_binary$species, levels=c('Adelie', 'Chinstrap'))
table(pb_species, predicted)
```

```
> table(pb_species, predicted)
            predicted
pb_species   Adelie Chinstrap
  Adelie        141         5
  Chinstrap       6        62
```

Gentoo 0 0

```
# pb_species line is not necessary, but what happens if we instead did:
table(peng_binary$species, predicted)
```

still has Gentoo level

ROC curve

Recall the earlier "default" decision rule for binary responses: assign x to Yes if

$$P(\text{default} = \text{Yes} | X = x) > 0.5.$$

- This rule weights both types of mistakes (FN and FP) the same.
- But sometimes we care more about lowering false negatives. E.g., a credit card company trying to detect a fraudulent charge.
- Can lower the threshold from 0.5 to e.g., 0.2.
- What happens to TP rate and FP rate as threshold decreases?

ROC curve

The *ROC curve* simultaneously displays both types of errors for all thresholds.

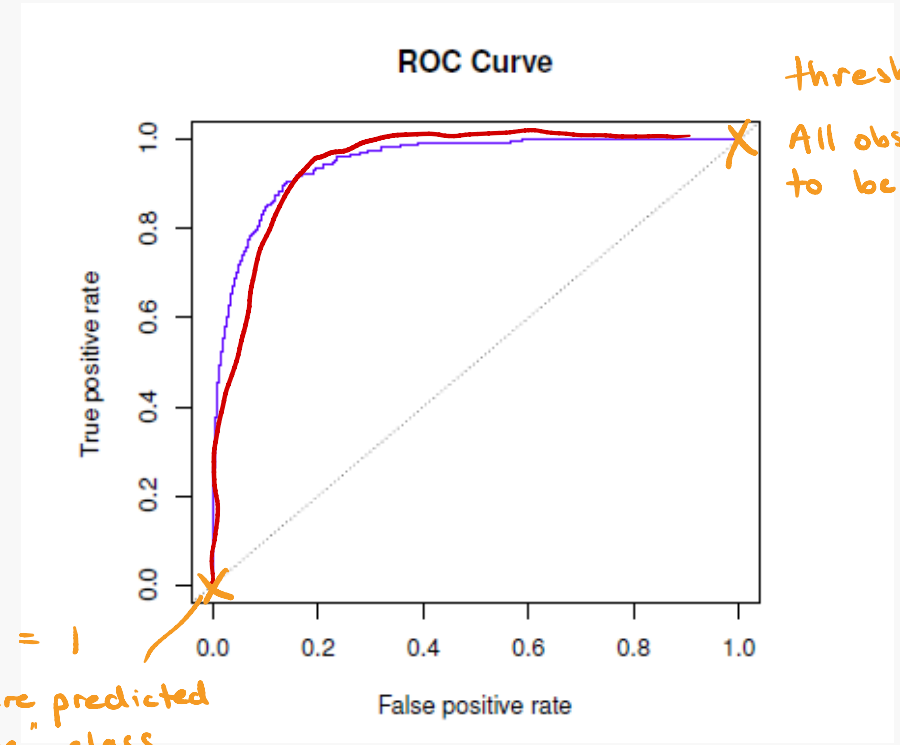


Figure 6: Image by James et al. (2021). An *ROC curve* for LDA classifier on `Default` data. Dotted line represents "no information" classifier, i.e., one that doesn't use predictors.

- ROC curve is parameterized by the possible threshold values.
- Overall performance of a classifier, summarized over all possible thresholds, is given by the *area under the ROC curve (AUC)*.
- The larger the AUC, the better the classifier.

Comparison of classification methods

Comparison

Analytical (or mathematical) comparison:

- Classifiers with a linear decision boundary are special cases of naive Bayes. Hence, LDA is a special case of naive Bayes. (This is not obvious.)
- No method uniformly dominates others: The appropriate model depends on the predictor's distribution in each class as well as n and p.
- K-nearest neighbors (KNN) is a ~~non-parametric~~ flexible approach, if K is small. Hence, one can expect that it dominates naive Bayes and LDA when the true decision boundary is highly non-linear. However, KNN requires many observations relative to the number of predictors to perform well.

For an *empirical* (or data-based) comparison, see Section 4.5.2 of ISLR2 textbook.