Section 10: Classification

STA 35C – Statistical Data Science III

Instructor: Akira Horiguchi

Fall Quarter 2025 (Sep 24 – Dec 12)
MWF, 12:10 PM – 1:00 PM, Olson 158
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. Errors in classification

Example (two categories)

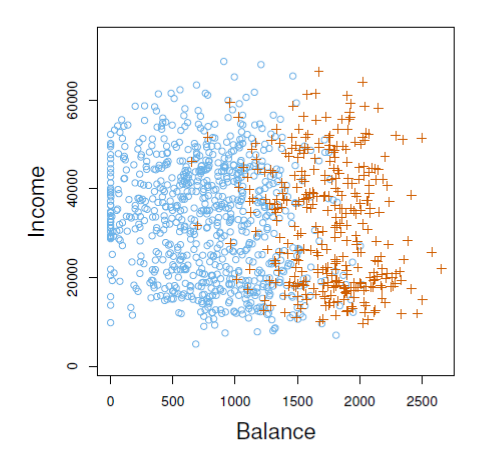


Figure 1: Image by James et al. (2021), based on the `Default` data set in R. The annual incomes and monthly credit card balances of a number of individuals, where the individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.

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?

   3 possible response values

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.

   2 possible response values

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.

   2 possible response values

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.,

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

  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.

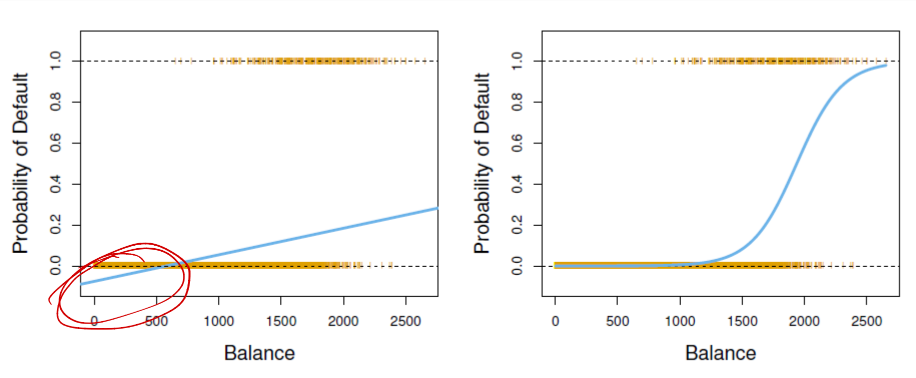


Figure 2: Image by James et al. (2021), based on the `Default` data set in R. Left: The estimated probability of default using *linear regression*, where the orange ticks indicate the values "0" for `No`, and "1" for `Yes`. Right: Predicted probabilities of default using *logistic regression*, where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

- Let's explore logistic regression.

Log odds

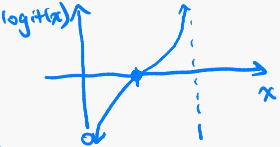
Let $P(A)$ be the *probability* that event A occurs. Then $P(A) \in [0, 1]$. Map to $(-\infty, \infty)$?

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

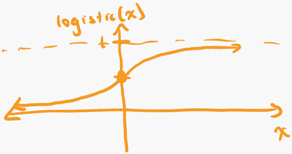
$$\log\left(\frac{P(A)}{1 - P(A)}\right) = \text{logit}\big(P(A)\big) \tag{1}$$

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 *logit*$\big(P(A)\big)$.

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right)$$



$P(A)$ $\qquad$ $\text{logit}\big(P(A)\big)$



$$\text{logistic}(x) = \frac{e^x}{1+e^x} = \frac{1}{e^{-x}+1}$$

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?)

$$P(Y=0 \mid X) = 1 - P(Y=1 \mid X)$$

The event $[Y=0 \mid X]$ is the complement of the event $[Y=1 \mid X]$

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:

$$\text{logit}\Big(p(X)\Big) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p. \tag{2}$$

- The conditional probabilities are then

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}. \tag{3}$$

Interpretation:

- 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 β_1.

- 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.

Estimation

Estimating the regression coefficients: β_0, β_1, β_2, ...

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

Estimating log odds or $p(X)$:

- We can estimate the log odds (2) at X by

$$\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p. \tag{4}$$

- Alternatively, we can estimate the conditional probability (3) at X by

$$\hat{p}(X) := \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p}}. \tag{5}$$

Example

If $\hat\beta_0 = -9.9$ and $\hat\beta_1 = 0.005$, — logistic regression estimates

- We estimate the log odds of default for individuals with balance $X = \$1,000$ and $X = \$2,000$ by

$$\hat\beta_0 + \hat\beta_1 \cdot 1,000 = -9.9 + 0.005 \cdot 1,000 = -4.9,$$
$$\hat\beta_0 + \hat\beta_1 \cdot 2,000 = -9.9 + 0.005 \cdot 2,000 = 0.1.$$

- We estimate the corresponding probabilities as

$$\hat p(X = 1,000) = \frac{e^{\hat\beta_0+\hat\beta_1 X}}{1+e^{\hat\beta_0+\hat\beta_1 X}} = \frac{e^{-9.9+0.005\cdot 1,000}}{1+e^{-9.9+0.005\cdot 1,000}} \approx 0.007,$$
$$\hat p(X = 2,000) = \frac{e^{\hat\beta_0+\hat\beta_1 X}}{1+e^{\hat\beta_0+\hat\beta_1 X}} = \frac{e^{-9.9+0.005\cdot 2,000}}{1+e^{-9.9+0.005\cdot 2,000}} \approx 0.525.$$

	default rule (0.5)	rule with threshold 0.8
X = 1000	0	0
X = 2000	1	0

Prediction

Suppose we have computed/estimated $P(Y=1|X)$ for a given value of predictor X. What class (0 or 1) should be assigned to X?

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

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

$\text{logit}(0.5) = 0$

$\Leftrightarrow$ if log odds > 0

$\Leftrightarrow$ if log odds ≤ 0

- If we don't know $P(Y=1|X)$, replace it with estimate (3).
- Is threshold of 0.5 appropriate if a false positive is worse than a false negative? E.g., is it worse to mark an innocent person as `guilty`, or mark a guilty person as `not guilty`?
  May want to change the decision rule to assign:

$$\begin{cases} \texttt{guilty} & \text{if } P(Y=1|X) > 0.8; \Leftrightarrow \text{if log odds} > \text{logit}(0.8) \\ \texttt{not guilty} & \text{if } \underbrace{P(Y=1|X) \leq 0.8}_{P(Y=0|X) > 0.8}. \Leftrightarrow \text{if log odds} \leq \text{logit}(0.8) \end{cases}$$

- (Back to example above.)

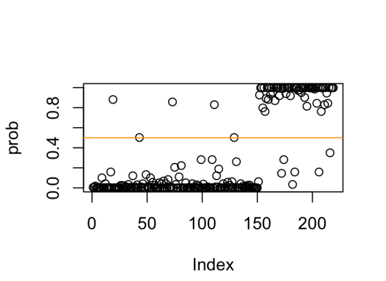
glm( )

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

$$\alpha_l(x) := \beta_{l,0} + \beta_{l,1}x_1 + \cdots + \beta_{l,p}x_p \qquad \text{for any } l = 1, \ldots, K-1. \tag{6}$$

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

End of 11/12 lecture

For convenience, copy-and-paste Eq. (6) 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 (~~??~~) with the model
   Eq. (3)

$$P(Y = k \mid X = x) = \begin{cases} \dfrac{e^{\alpha_k(x)}}{1+\sum_{l=1}^{K-1} e^{\alpha_l(x)}} & \text{for } k = 1, \ldots, K-1, \\[2ex] \dfrac{1}{1+\sum_{l=1}^{K-1} e^{\alpha_l(x)}} & \text{for } k = K. \end{cases}$$

For $k = 1, \ldots, K-1$, we have $P(Y=1 \mid X=x) + P(Y=2 \mid X=x) + \cdots + P(Y=K \mid X=x) = 1$

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

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

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

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

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 3: 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

```r
# 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
```

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

ROC curve

Recall the earlier "default" decision rule for binary responses: assign x to $\mathtt{Yes}$ if

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

- 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.

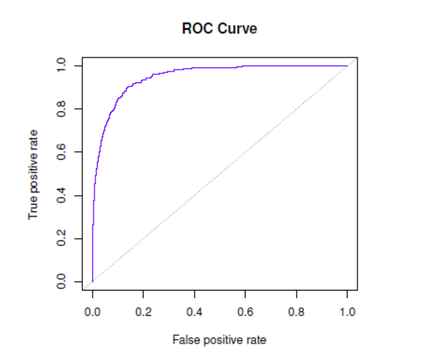


Figure 4: Image by James et al. (2021). An *ROC curve* for LDA classifier on `Default` data. Dotted line represents "no information" classifier, i.e., one that doesn't use predictors.

- 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.