STA 141A – Fundamentals of Statistical Data Science

Department of Statistics; University of California, Davis

Instructor: Dr. Akira Horiguchi (ahoriguchi@ucdavis.edu)
A01 TA: Zhentao Li (ztlli@ucdavis.edu)
A02 TA: Zijie Tian (zijtian@ucdavis.edu)
A03 TA: Lingyou Pang (lyopang@ucdavis.edu)

Section 7: Classification with R

Spring 2025 (Mar 31 – Jun 05), MWF, 01:10 PM – 02:00 PM, Young 198

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`

Section 7: Classification.

- Logistic regression
- Alternatives to logistic regression
- Linear discriminant analysis for $p = 1$
- Idea of linear discriminant analysis for $p > 1$
- Idea of quadratic discriminant analysis for $p > 1$
- Naive Bayes
- Errors in classification
- Comparison of classification methods

Examples

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.

A plot

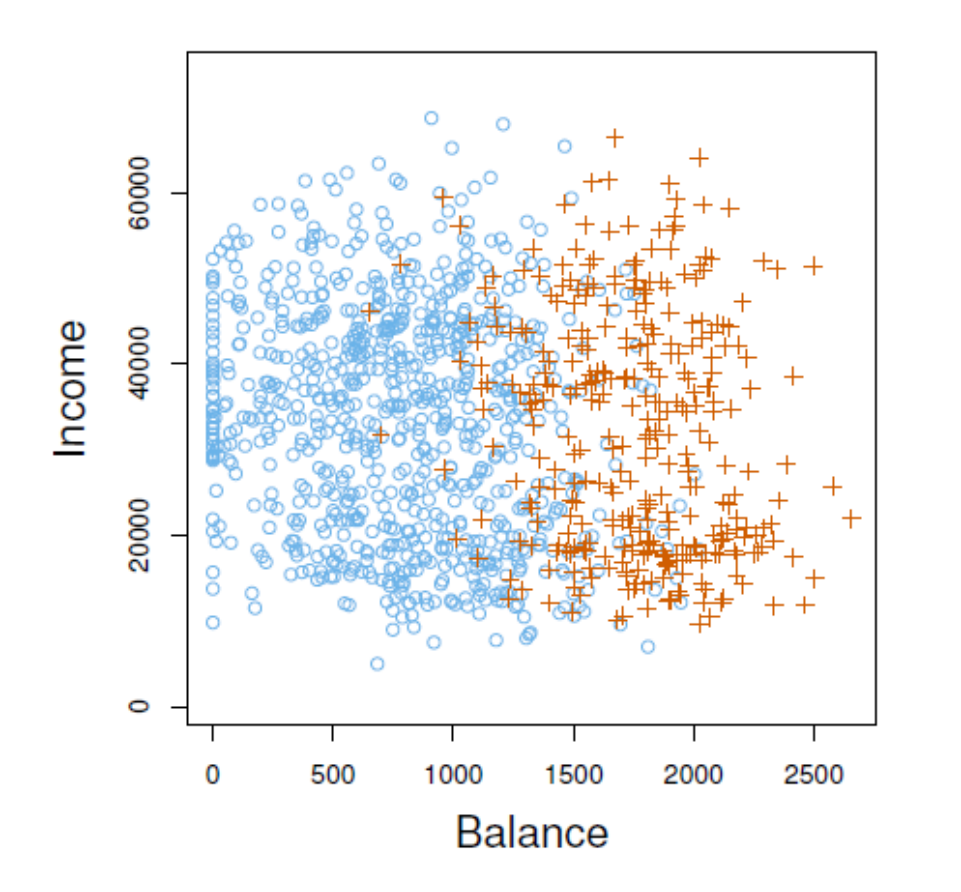


Figure 1: Image by James et al. (2021), based on the `Default` data set in R. The annual incomes and monthly credit card balances of a number of individuals, where the individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.

Classification refers to the task of predicting *qualitative*/*categorical* responses

- Each response y_i is a discrete value in a predetermined category.
- Predicting a qualitative response for an observation can be referred to as *classifying* that observation, as one assigns the observation to a certain category/class.
- (In contrast, regression deals with "continuous" numeric response values.)

As in regression...

- We use training observations $(x_1, y_1), \ldots, (x_n, y_n)$ to find the best estimator in the allowed class of models.
  - That is, we try to find the best estimator (among the allowed class) that fits the training data.
- We then evaluate how well the estimator performance generalizes to unseen data.
  - We can do this directly if there are also test observations $(x_{n+1}, y_{n+1}), \ldots, (x_{n+m}, y_{n+m})$.
  - Otherwise, we can use a resampling method to estimate the estimator's generalization ability (Sec 8).

Why not linear regression?

- 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 assign each medical condition Y a number from 1 to 3:

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

  Both approaches work, but imply a totally different relationship.
- Can we use linear regression for a *binary* (two levels) response? In the banking example, the two transaction categories can be coded as

$$Y = \begin{cases} 1, & \text{if "Fraudulent",} \\ 0, & \text{if "Not fraudulent".} \end{cases}$$

  We have no problem with ordering here, but using linear regression here (for estimating probabilities to assign values) could lead to values smaller than zero or larger than one (unreasonable for probabilities!).

Problem with using linear regression

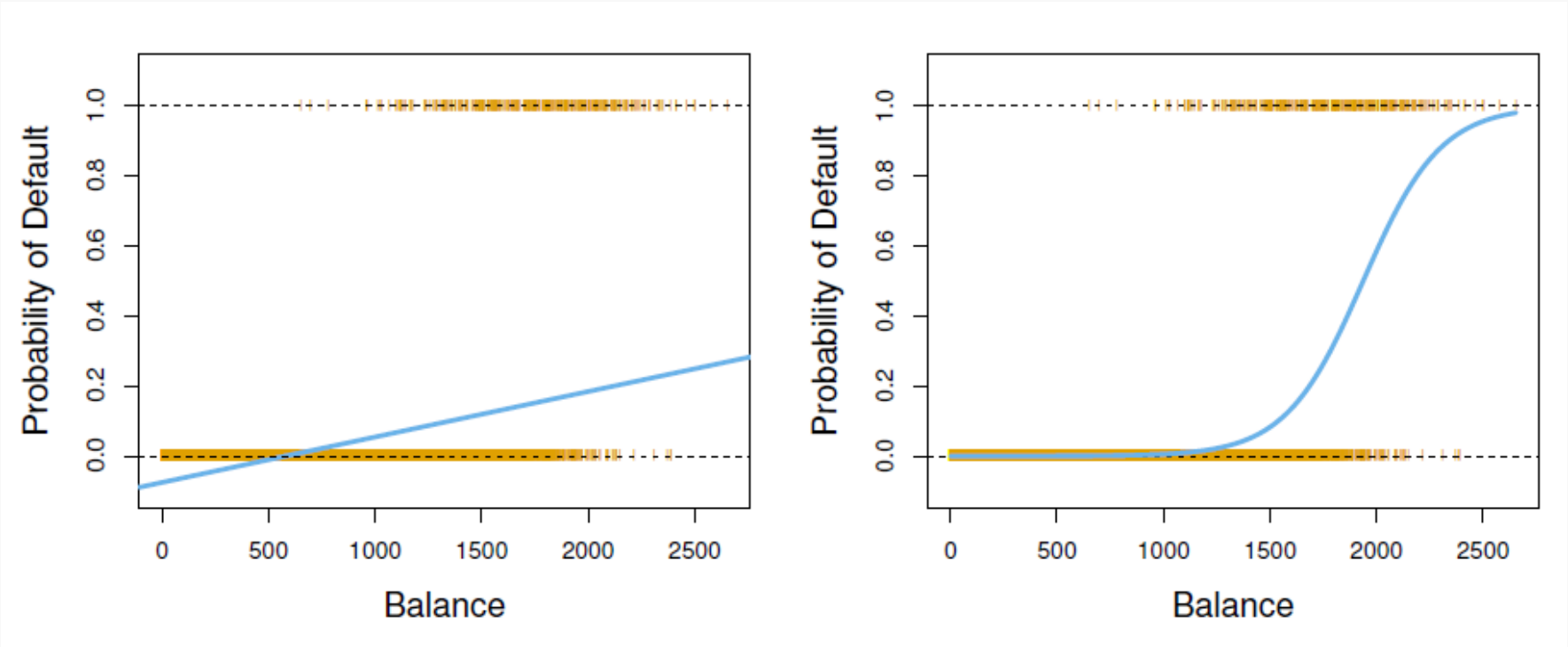


Figure 2: Image by James et al. (2021), based on the `Default` data set in R. Left: The estimated probability of default using linear regression, where the orange ticks indicate the values "0" for "No", and "1" for "Yes". Right: Predicted probabilities of default using logistic regression, where all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

Problem with using linear regression

If not linear regression, then what can we use?

- *Logistic regression* (usable only for binary responses) is perhaps the most related, so let's start there.

Classification with R

Logistic regression

Idea: Binary classification

Classification: compute/estimate conditional probability $P(Y = k|X)$ for each class k. $P(Y=1|X)$, $P(Y=2|X)$, $P(Y=3|X)$, ..., $P(Y=K|X)$

- If there are only two classes, we only need $P(Y = 1|X)$. (Why?)

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

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

If we know $P(Y=1|X)$, then we also know $P(Y=0|X)$.

Suppose our two classes are coded as 0 and 1 (e.g., "no" and "yes")

- Given a value or estimate of the conditional probability

$$p(X) = P(Y = 1|X). \tag{1}$$

  a default decision rule is to predict 1 (or "yes") if and only if $p(X) > 0.5$
- Consequence might be worse for misclassifying a "yes" than for misclassifying a "no", in which case we might use the decision rule: predict 1 (or "yes") if and only if $p(X) > 0.8$

Logistic function and odds

Logistic regression models probabilities of binary responses

- Boils down to modelling probability of "yes": (1)
- Since probabilities have values in $[0, 1]$, we cannot use the linear approach

$$p(X) = \beta_0 + \beta_1 X \tag{2}$$

  as on the left side in Figure 2, and need a function with values in $[0, 1]$.
- The *logistic function* $g(x) = \frac{e^x}{1+e^x}$ maps the real line into the interval $(0, 1)$.
- Logistic regression maps the linear form (2) into values between 0 and 1 in order to model the probabilities by

$$p(X) = g(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \tag{3}$$

- (3) implies (by rearranging the terms)

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}, \tag{4}$$

  where this quantity is called *odds*.
- Applying the (natural) logarithm on both sides of (4) yields

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

  where the LHS is called *log odds* or *logit*. Advantage: linear in X.

- We can make predictions from estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ for β_0 and β_1 by using the estimated probability

$$\hat{p}(X) = g\left(\hat{\beta}_0 + \hat{\beta}_1 X\right) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 X}},$$

  where we recall that g is the logistic function.
- If this estimated probability is over a certain pre-defined threshold (e.g. 0.25), then we would assign $X = x$ to the class 1.
- Example: If $\hat{\beta}_0 = -9.9$ and $\hat{\beta}_1 = 0.005$, we predict the default probabilities of individuals with balance $X = \$1,000$ and $X = \$2,000$ by

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

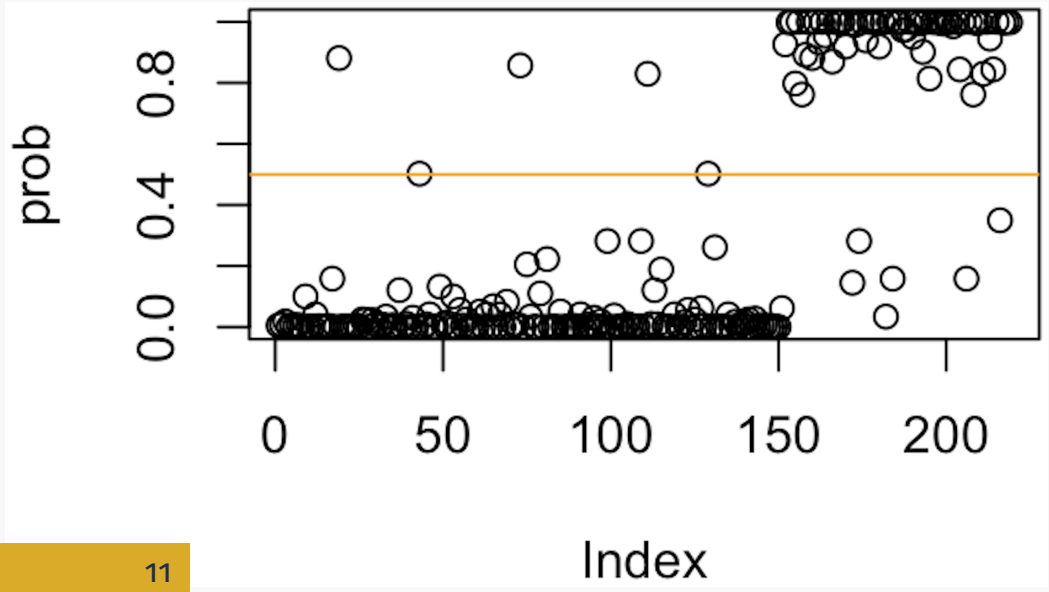
$$\hat{p}(X = 2,000) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-9.9+0.005\cdot 2,000}}{1 + e^{-9.9+0.005\cdot 2,000}} \approx 0.525.$$

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.

```
1  library(palmerpenguins)
2
3  # We work with only Adelie and Chinstrap species (we exclude Gentoo).
4  peng_binary <- na.omit(penguins[penguins$species != 'Gentoo', ])
5  logreg <- glm(species ~ bill_length_mm, data=peng_binary, family=binomial)
6  prob <- predict(logreg, type='response')  # 'link' also possible
7  predicted <- ifelse(prob<.5, 'Adelie', 'Chinstrap')
8
9  plot(prob)
10 abline(a=0.5, b=0, col='orange')
```



If we have p predictors $X = (X_1, \ldots, X_p)$, we have a *multiple logistic regression model*:

- The logit is

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

if $p=1$

- The probabilities are

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

Classification with R

Alternatives to logistic regression

Motivation

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

Some issues:

- When there are big differences between two classes, the parameter estimates in the logistic regression model are unstable.
- If the distribution of X is approximately normal in each of the classes and the sample size is small, then other approaches are more accurate than logistic regression.

Theorem 2: Bayes' theorem

Let $\Omega \neq \emptyset$. For any events $A, B \subseteq \Omega$ with $P(B) \neq 0$ holds,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{5}$$

$P(A \cap B) = P(A|B)P(B)$

$P(B \cap A) = P(B|A)P(A)$

$P(A|B)P(B) = P(B|A)P(A)$ — divide both sides by $P(B)$

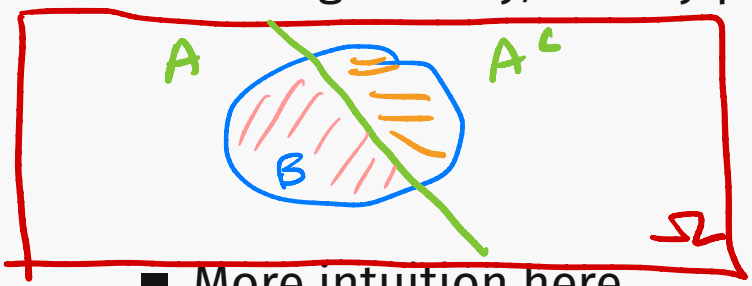
Often $P(B)$ is unknown, but law of total probability can help compute $P(B)$.

- Because the sets A and A^c partition Ω, we can write $P(B)$ as

$P(B) = P(B \cap \Omega) = P(B \cap [A \cup A^c]) = P([B \cap A] \cup [B \cap A^c]) = P(B \cap A) + P(B \cap A^c)$

$$P(B) = P(B|A)P(A) + P(B|A^c)P(A^c).$$

- More generally, for any partition $\{A_1, A_2, \ldots\}$ of Ω, we can write $P(B)$ as



$$P(B) = \sum_{j=1}^{\infty} P(B|A_j)P(A_j).$$

- More intuition here
  https://www.youtube.com/watch?v=9wCnvr7Xw4E

Using Bayes' theorem

We can instead use Bayes' theorem to estimate $P(Y = k|X = x)$:

$$p_k(x) := P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}, \tag{6}$$

where

- $\pi_k := P(Y = k)$ 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 $\pi = 3.14159\ldots$!
- $f_k(X)$ is the PMF/PDF of X given that the response is from the kth class.

How to estimate these quantities π_k and f_k?

- π_k — 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 — approaches will be discussed in the next few slides.

Classification with R

Linear discriminant analysis for $p = 1$

Assumptions of f_k in LDA for $p = 1$

This approach estimates $f_k(x)$ by making some simplifying assumptions:

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

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right\}. \tag{7}$$

- Same variance parameter across all K classes: $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_K^2 = \sigma^2$.

K classes $\Rightarrow$ $\mu_1, \ldots, \mu_K$ & σ^2

Bayes decision boundary

With these assumptions, we plug in this PDF (7) into (6) to get

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu_k)^2\right\}}{\sum_{\ell=1}^{K} \pi_\ell \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu_\ell)^2\right\}}. \quad (8)$$

doesn't depend on k.

Denominator is the same in $p_1(x), p_2(x), p_3(x), \ldots$

- Recall: *Bayes classifier* assigns obsn x to class

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

- This is equivalent to assigning x to class for which *discriminant function*

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k), \qquad k \in \{1,2\}, \quad (9)$$

  is largest. (Why? Take the log of (8).)
- 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 $\pi_1 = \pi_2$?

inequality

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

$\Leftrightarrow x\frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} > x\frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2}$

$\Leftrightarrow x > \frac{\mu_1^2 - \mu_2^2}{2} \frac{1}{\mu_1 - \mu_2} = \frac{\mu_1+\mu_2}{2}$

boundary

$\delta_1(x) = \delta_2(x)$

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

Example for decision boundary

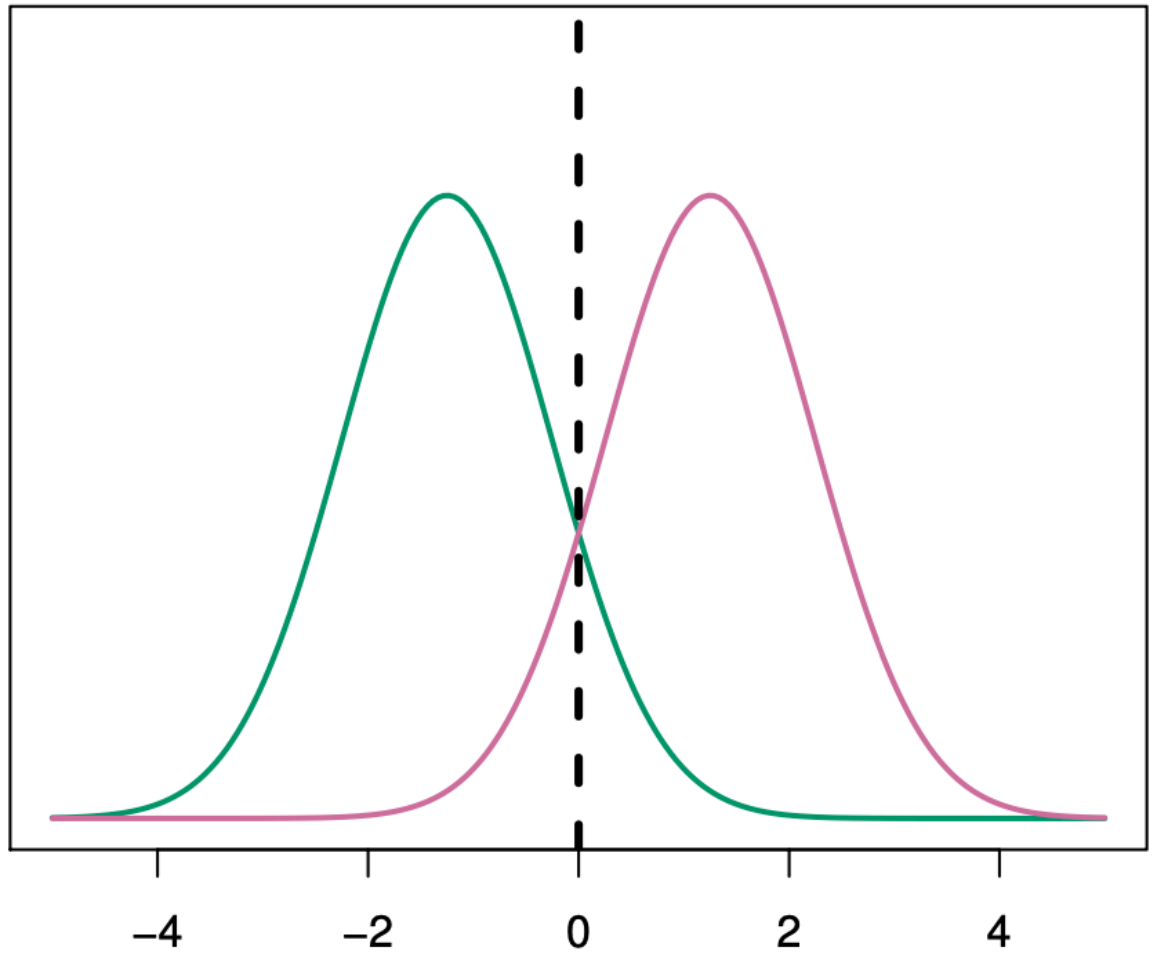


Figure 3: Image by James et al. (2021). Two pdfs of normal distributions with means $\mu_1 = -1.25$ and $\mu_2 = 1.25$, respectively, and variance $\sigma^2 = 1$. The dashed vertical line represents the Bayes decision boundary, so we assign the observation to class 1 if x is left of the line, and to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

In the plot above, we can calculate Bayes classifier because we know values for all parameters $\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \sigma^2$. $K=2 \Rightarrow \pi_1, \pi_2, \mu_1, \mu_2, \sigma^2$

- In practice, we must estimate these parameters to apply Bayes classifier.
- *Linear discriminant analysis* (LDA) method plugs the estimates

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

  into (9) to get the discriminant function

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k). \tag{10}$$

- $\hat{\pi}_k$ — 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.

This discriminant function $\hat{\delta}_k(x)$ is linear in x, hence the name LDA.

Example for decision boundary

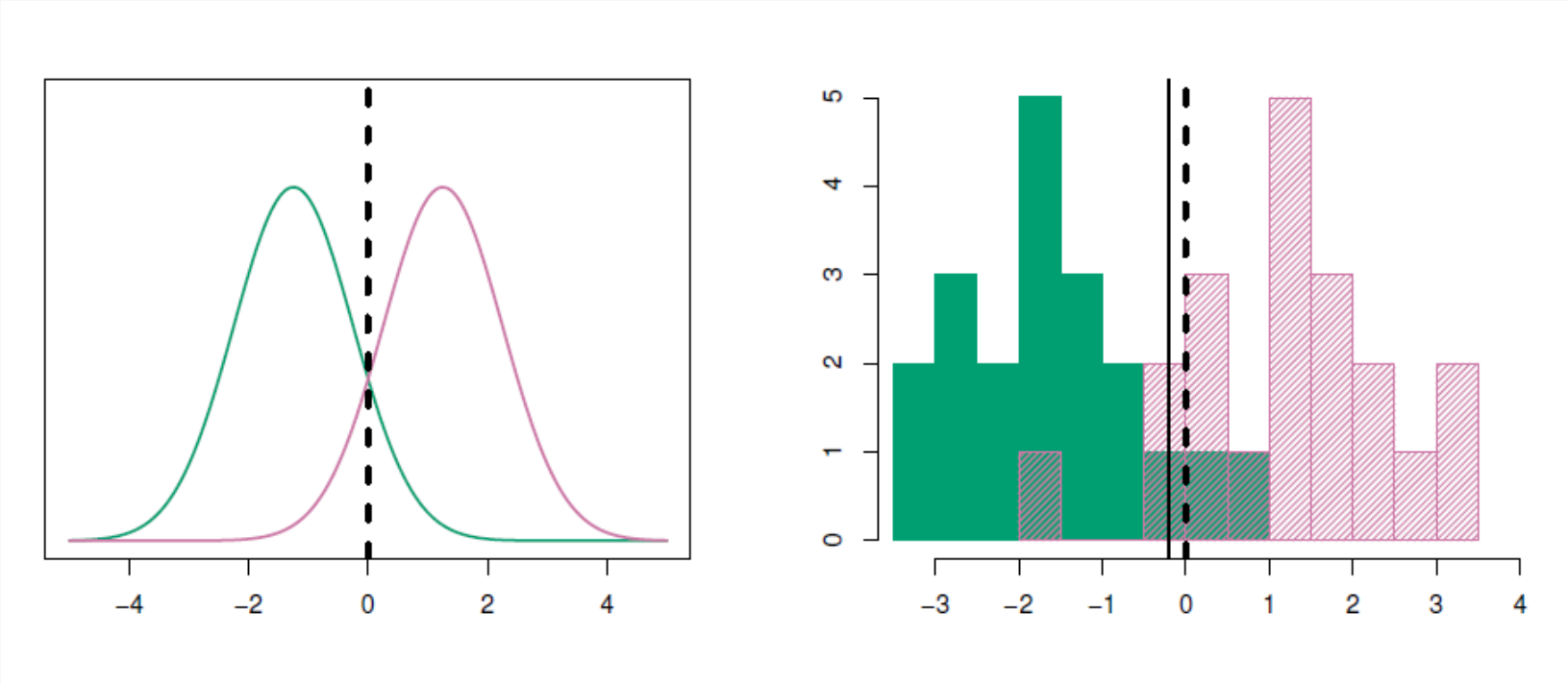


Figure 4: Image by James et al. (2021). Left: Two pdfs of normal distributions with means $\mu_1 = -1.25$ and $\mu_2 = 1.25$, respectively, and variance $\sigma^2 = 1$. The dashed vertical line represents the Bayes decision boundary, so we assign the observation to class 1 if $x < 0$ and class 2 otherwise. 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 (10) for $k = 0$ and $k = 1$.

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

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

$\hat{\pi}_0 = \frac{n_0}{n} = \frac{3}{5}$

$\hat{\pi}_1 = \frac{n_1}{n} = \frac{2}{5}$

$\hat{\mu}_0 = \frac{1}{3}[8+6+4] = 6$

$\hat{\mu}_1 = \frac{1}{2}[9+7] = 8$

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

class 0 class 1

$= \frac{1}{3}[8+2] = \frac{10}{3}$

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

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

Both discriminant functions are linear in x.

Assumptions of f_k in LDA for $p = 1$

Recall (stringent) assumptions: f_k is normal/Gaussian, and variance is same across all K classes.

- Next we introduce extensions that loosen these assumptions at the cost of increased computation and/or "too much" flexibility.

Classification with R

Idea of linear discriminant analysis for $p > 1$

Linear discriminant analysis can also be conducted for $p > 1$ predictors.

- For $p > 1$, it is also assumed that $X = (X_1, \ldots, X_p)$ is normally distributed, but since X is a vector, it is drawn from a *multivariate normal distribution* with a certain mean vector $(\mu_1, \ldots, \mu_p)$ and covariance matrix Σ.
- Discriminant function can be derived in a manner similar to as in $p = 1$.

A **p-variate normal distribution** is characterized by a p-dimensional mean vector and a $p \times p$ covariance matrix

- Mean vector: the center of the p-variate distribution.
- Covariance matrix Σ: captures the variance and covariance relationships between $X_1, \ldots, X_p$.
  - Element in row i and column j is the covariance between X_i and X_j.
  - ith diagonal element is the variance of X_i.
  - Thus Σ is a symmetric matrix.

Plot of multivariate normal distribution

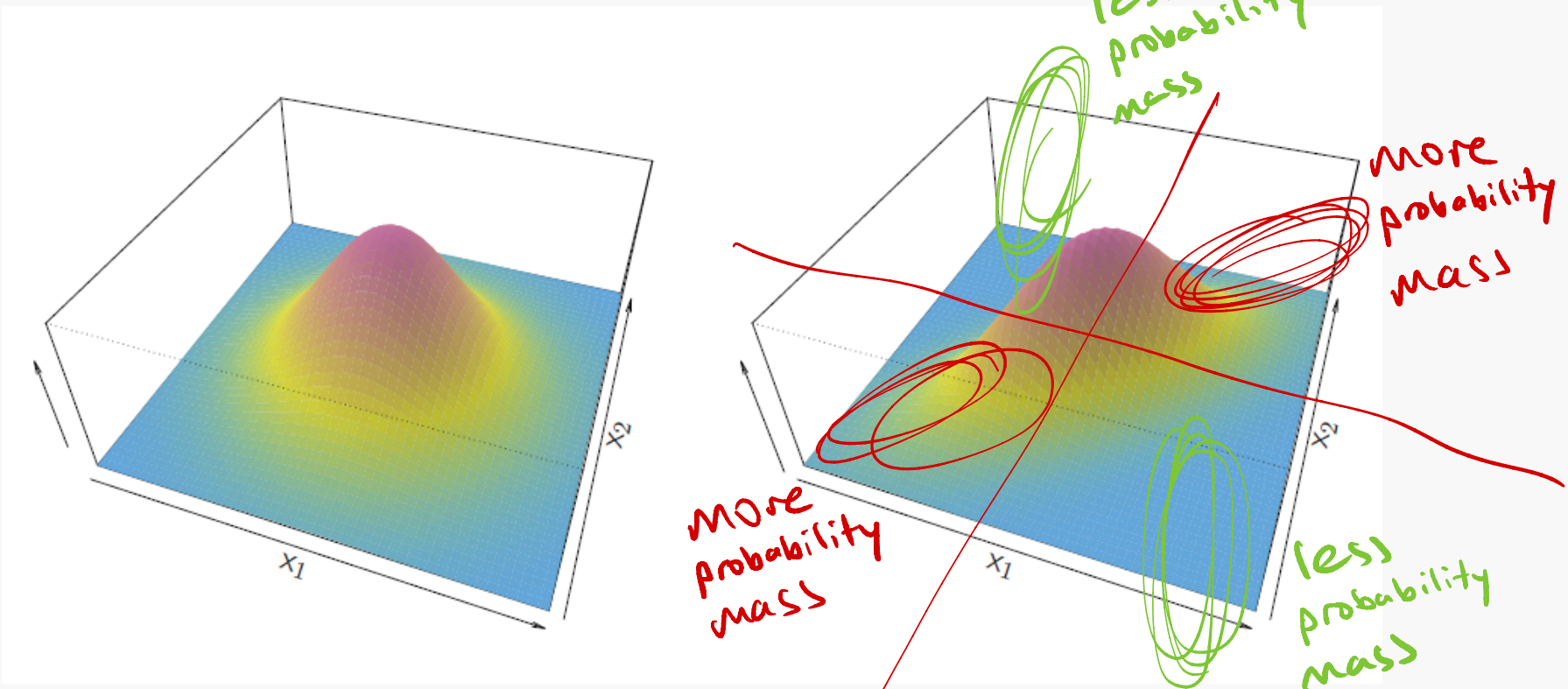


Figure 5: Image by James et al. (2021). Left: A two-dimensional normal distribution with $p = 2$ uncorrelated predictors. Right: A two-dimensional normal distribution with $p = 2$ predictors having a correlation of 0.7.

Classification with R

Idea of quadratic discriminant analysis for $p > 1$

Idea

Each class k is now allowed to have its own $p \times p$ "covariance matrix" $\mathbf{\Sigma}_k$.

- For $p = 1$, same as saying each class has its own variance parameter σ_k^2.
- Why use QDA over LDA, or vice-versa? *Bias-variance trade-off* (recall from Section 5, slide 29/35).
- For p predictors, number of parameters to estimate is
  - quadratic in p for QDA,
  - linear in p for LDA.
- LDA is much less flexible, and so has much lower variance.
- But if covariance matrices wildly differ between classes, then LDA can suffer from high bias.
- Roughly speaking,
  - use LDA if n is relatively small, which makes it crucial to reduce variance,
  - use QDA if n is very large, or if assumption of a common covariance matrix for the K classes is clearly wrong.

Plot of multivariate normal distribution

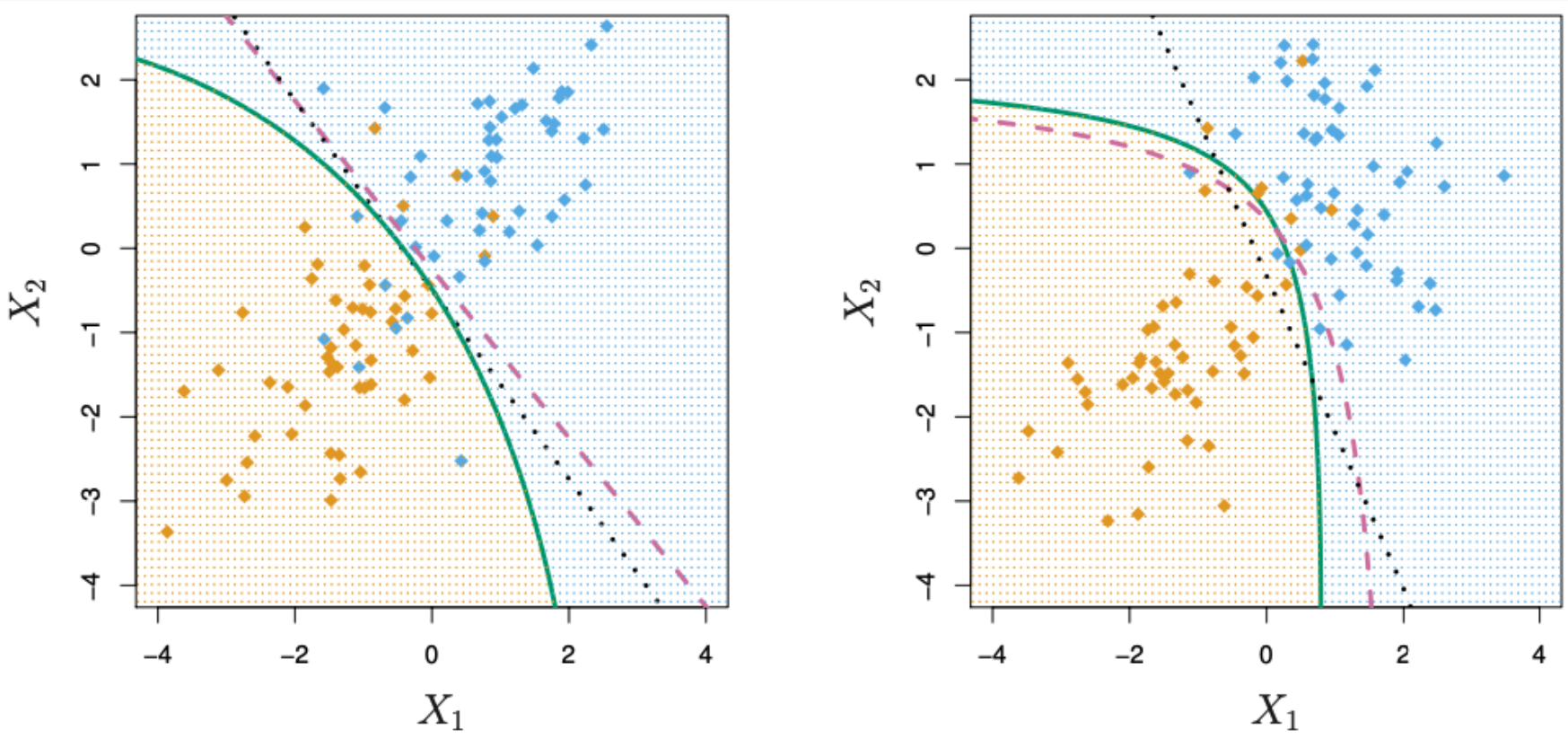


Figure 6: Image by James et al. (2021). Left: The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with $\Sigma_1 = \Sigma_2$. The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA. Right: here $\Sigma_1 \neq \Sigma_2$. Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.

Classification with R

Naive Bayes

The *naive Bayes classifier* relies on Bayes' theorem.

- LDA/QDA make distribution assumptions about $f_k(x)$.
- Instead, the naive Bayes classifier assumes for each class $k = 1, \ldots, K$:

  Within the kth class, the p predictors are independent.

- Mathematically, this means that for each class k:

$$f_k(x) = f_{k,1}(x_1) \times f_{k,2}(x_2) \times \cdots \times f_{k,p}(x_p) \tag{11}$$

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

- Plug (11) into (6) to get the posterior probability

$$p_k(x) := P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}, \tag{6}$$

$$p_k(x) = P(Y = k|X = x) = \frac{f_{k,1}(x_1) \cdot f_{k,2}(x_2) \cdot f_{k,p}(x_p)\pi_k}{\sum_{\ell=1}^{K} f_{\ell,1}(x_1) \cdot f_{\ell,2}(x_2) \cdot f_{\ell,p}(x_p)\pi_\ell}. \tag{12}$$

- The *naive Bayes classifier* will put an observation with predictor x into the class that maximizes posterior probability (12)

Independence assumption

The independence assumption, although often unrealistic, produces decent results, especially when n is too small to effectively estimate the joint distribution $f_k(x)$ of predictors x within each class k.

- Without strong simplifying assumptions, estimating a joint distribution typically requires a huge amount of data.
- Bias-variance tradeoff: introduce some bias to reduce variance.

Estimating the posterior probability (12) requires estimating the univariate density functions $f_{k,j}$ for all classes $k = 1, \ldots, K$ and all predictors $j = 1, \ldots, p$.

Several 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 (so $p = 1$). We survey 100 people:

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

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

	Math major	Art major	Poli Sci major
Study $>$ 10 hr /wk	0.6	0.55	0.35
Study $\leq$ 10 hr /wk	0.4	0.45	0.65

Classification with R

Errors in classification

Confusion matrix

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

- In binary classification one can make two mistakes: *false positives* and *false negatives*.
- Examples of 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*	

End of May 9 lecture

Figure 7: 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 previous 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
```

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

ROC curve

Recall: Bayes classifier assigns observation to class for which the posterior probability (12) is largest.

- For binary responses, assign x to "Yes" if

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

- 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.

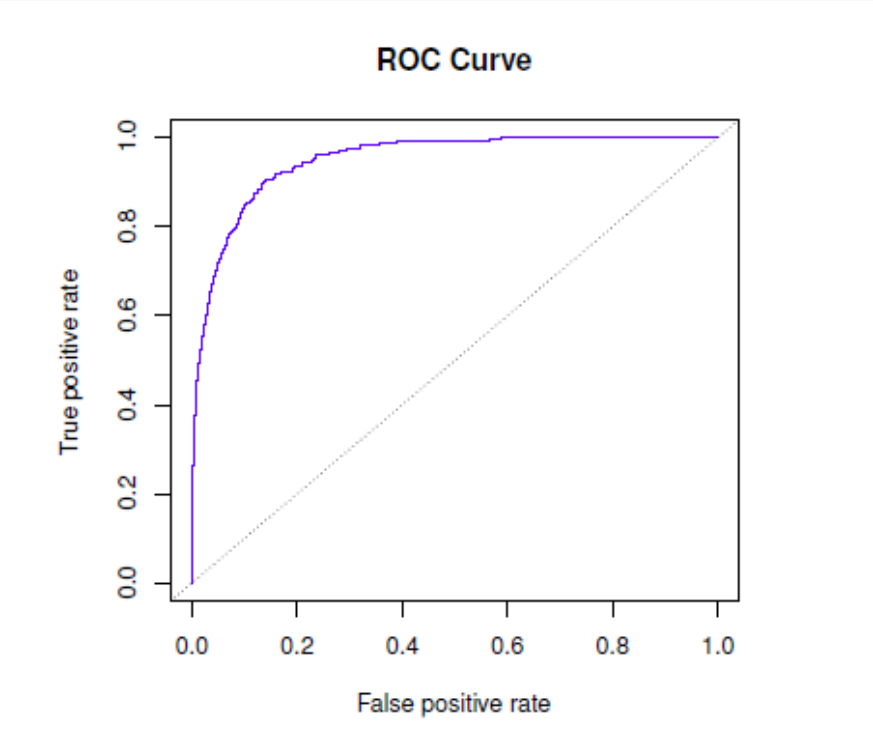


Figure 8: Image by James et al. (2021). An *ROC curve* for LDA classifier on `Default` data. Dotted line represents "no information" classifier, i.e., one that doesn't use predictors.

- 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.

Classification with R

Comparison of classification methods

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 nonparametric approach. 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.