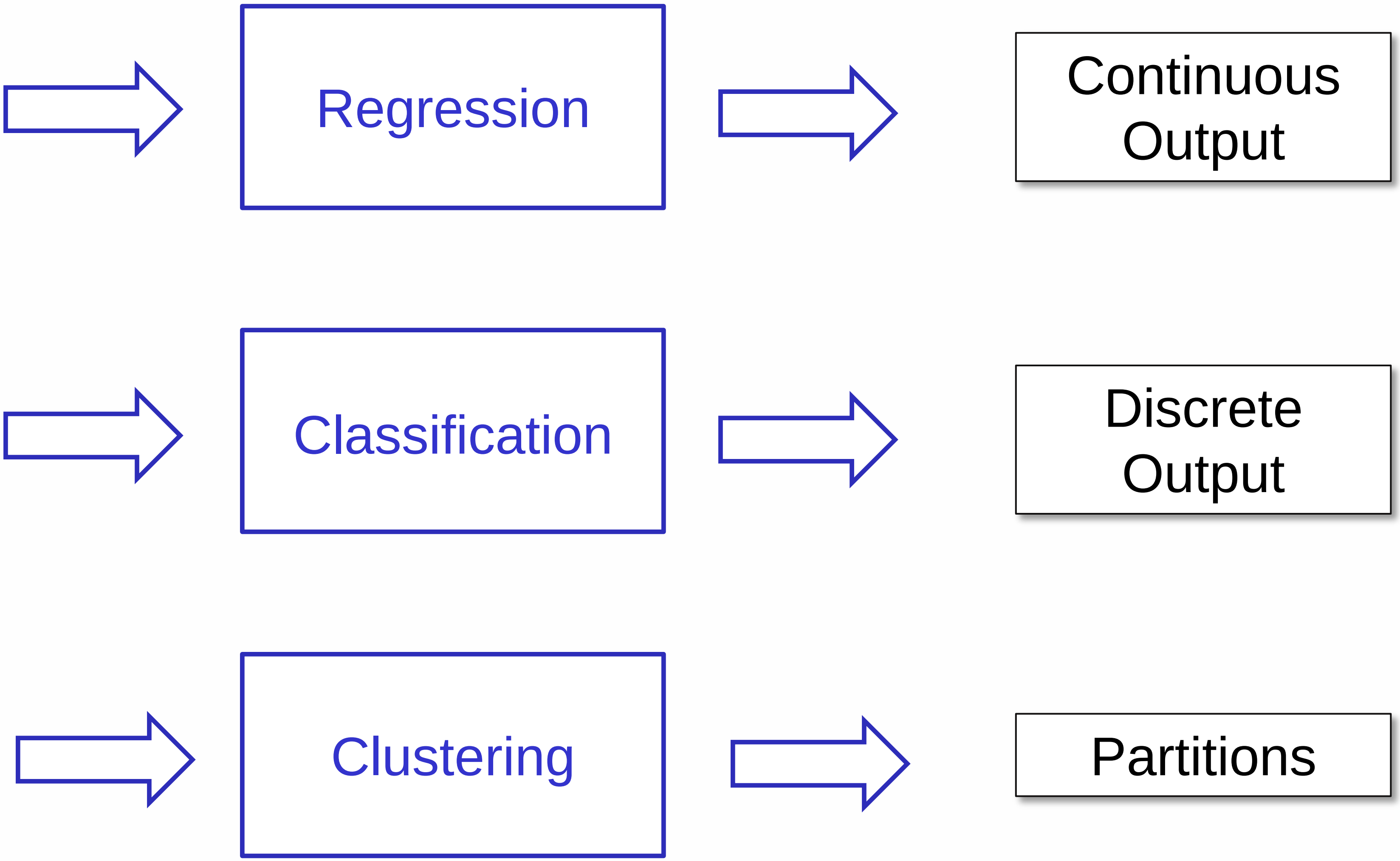
POLITECNICO DI MILANO

Pattern Analysis and Machine Intelligence

Linear Classification

Prof. Matteo Matteucci

Recall from the first lecture …



Example: Default dataset

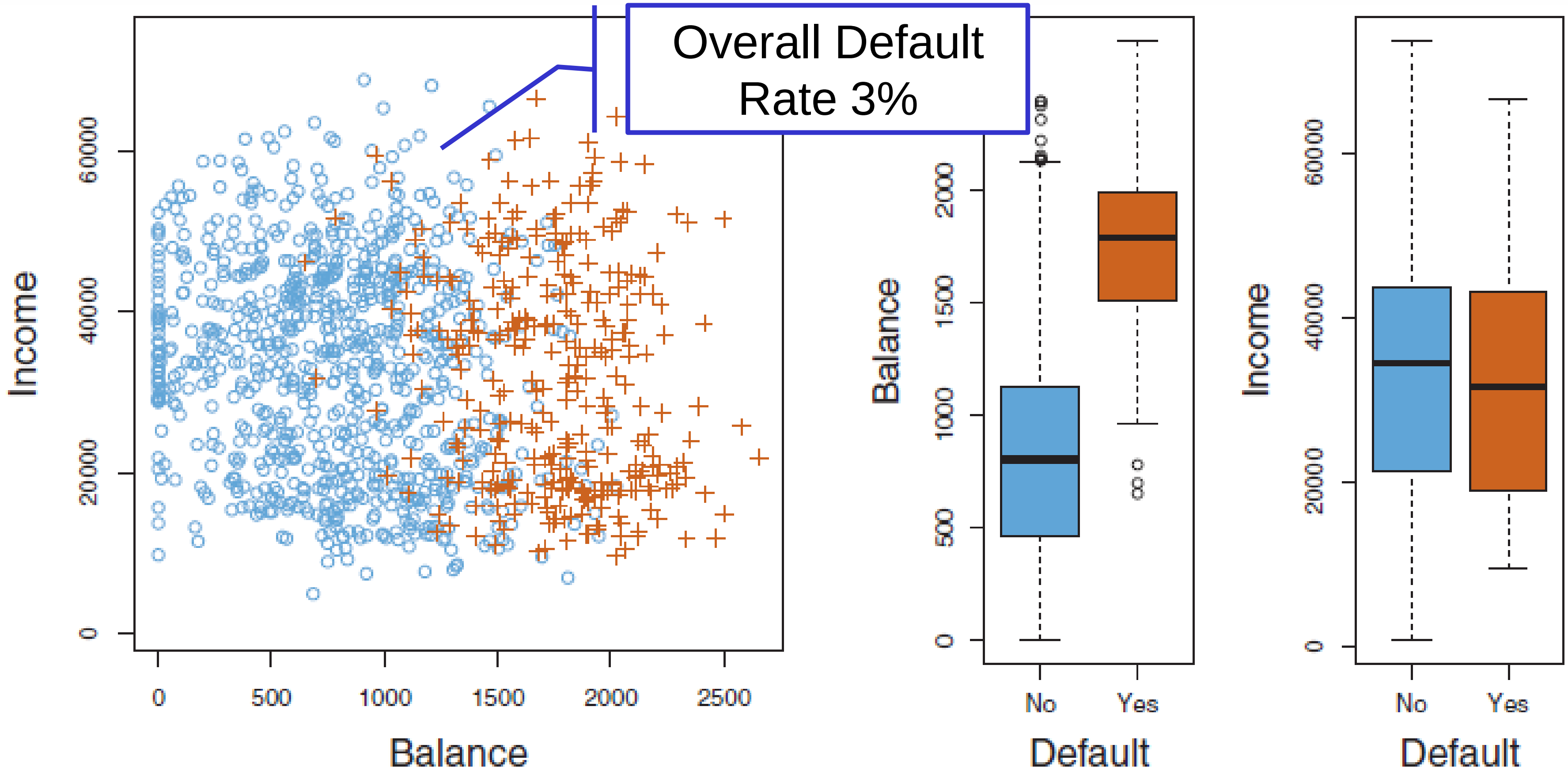


FIGURE 4.1. *The* `Default` *data set.* Left: *The annual incomes and monthly credit card balances of a number of individuals. The individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.* Center: *Boxplots of* `balance` *as a function of* `default` *status.* Right: *Boxplots of* `income` *as a function of* `default` *status.*

Linear regression for classification?

- Suppose to predict the medical condition of a patient. How should this be encoded?

  - We could use dummy variables in case of binary output

$$Y = \begin{cases} 0 & \text{if } \texttt{stroke}; \\ 1 & \text{if } \texttt{drug overdose}. \end{cases}$$

but how to deal with multiple output?

  - Different encodings could result in different models

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases} \quad Y = \begin{cases} 1 & \text{if } \texttt{epileptic seizure}; \\ 2 & \text{if } \texttt{stroke}; \\ 3 & \text{if } \texttt{drug overdose}. \end{cases}$$

Recall here the Bayesian Classifier

- For a classification problem we can use the error rate i.e.

$$Error\ Rate = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i)/n$$

  - Where $I(y_i \neq \hat{y}_i)$ is an indicator function, which will give 1 if the condition $(y_i \neq \hat{y}_i)$ is correct, otherwise
  - The error rate represents the fraction of inc classifications, or misclassifications

The best classifier possible estimates the class posterior probability!!

- The Bayes Classifier minimizes the Average Test Error Rate

$$\max_j P(Y = j \mid X = x_0)$$

- The **Bayes error rate** refers to the lowest possible Error Rate achievable knowing the "true" distribution of the data

$$1 - E\left(\max_j \Pr(Y = j|X)\right)$$

- We want to model the probability of the class given the input

$$p(X) = \Pr(Y = 1|X)$$

$$p(X) = \beta_0 + \beta_1 X$$

but a linear model has some drawbacks

Example: Default data & Linear Regression

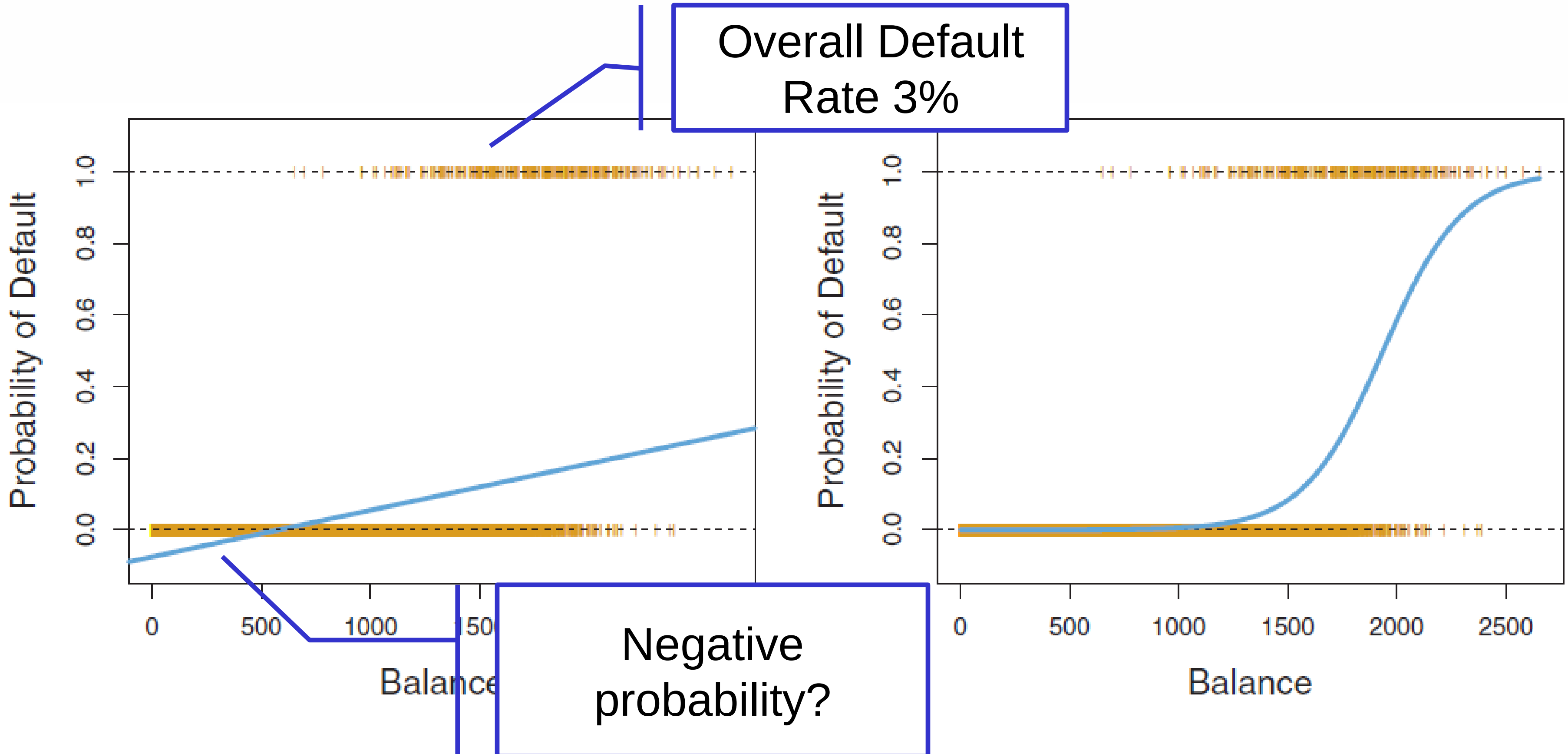


FIGURE 4.2. *Classification using the* Default *data. Left: Estimated probability of* default *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* default*(*No *or* Yes*). Right: Predicted probabilities of* default *using logistic regression. All probabilities lie between* 0 *and* 1.

- We want to model the probability of the class given the input

$$p(X) = \Pr(Y = 1|X)$$

$$p(X) = \beta_0 + \beta_1 X$$

  but a linear model has some drawbacks
- Logistic regression solves the negative probability (ad other issues as well) by regressing the logistic function

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

Example: Default data & Logistic Regression

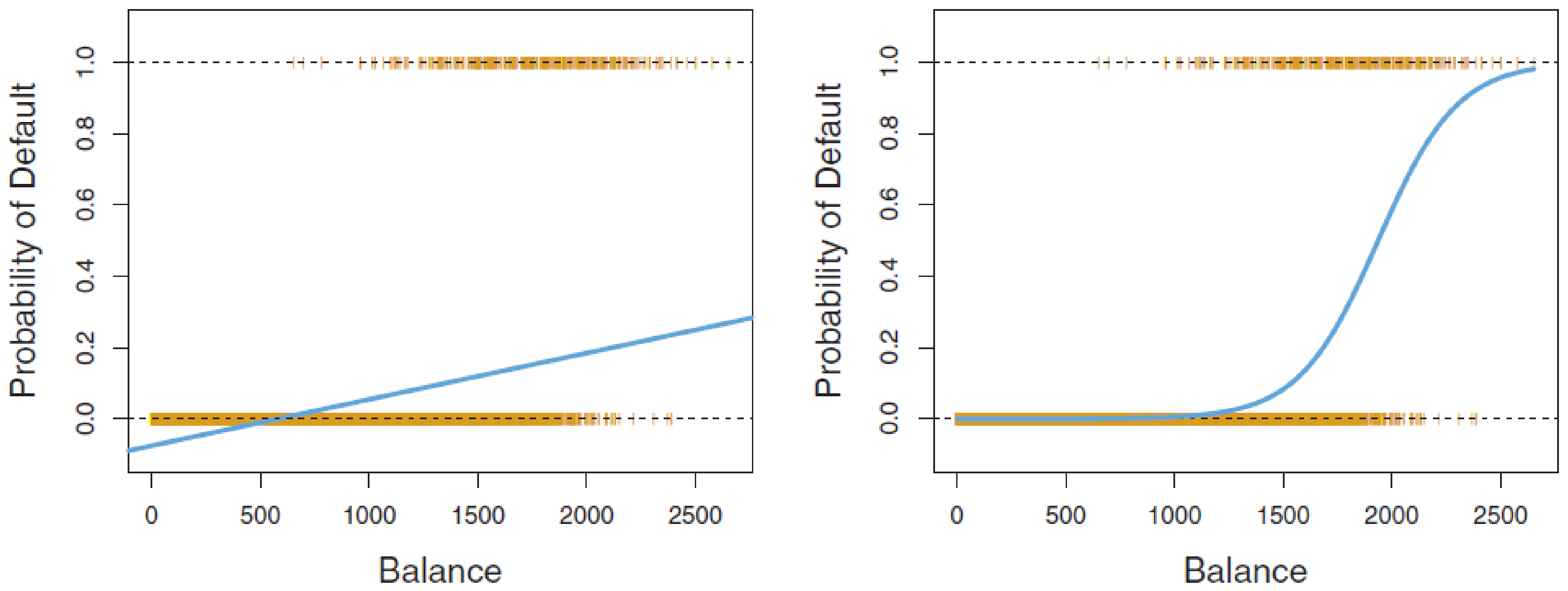


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default`*(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

Logistic Regression

- We want to model the probability of the class given the input

$$p(X) = \Pr(Y = 1|X)$$
$$p(X) = \beta_0 + \beta_1 X$$

Linear Regression

but a linear model has some drawbacks (see later slide)

- Logistic regression solves the negative probability (ad other issues as well) by regressing the logistic function

Logistic Regression

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

from this we derive

This is called *odds*

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

and taking logarithms

This is called *log-odds or logit*

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

Coefficient interpretation

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting P(Y) and not Y.

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

  - If β_1 =0, this means that there is no relationship between Y and X
  - If β_1 >0, this means that when X gets larger so does the probability that Y = 1
  - If β_1 <0, this means that when X gets larger, the probability that Y = 1 gets smaller.
- But how much bigger or smaller depends on where we are on the slope, i.e., it is not linear

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

Training Logistic Regression (1/4)

- For the basic logistic regression wee need two parameters

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

- In principle we could use (non linear) Least Squares fitting on the observed data the corresponding model

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

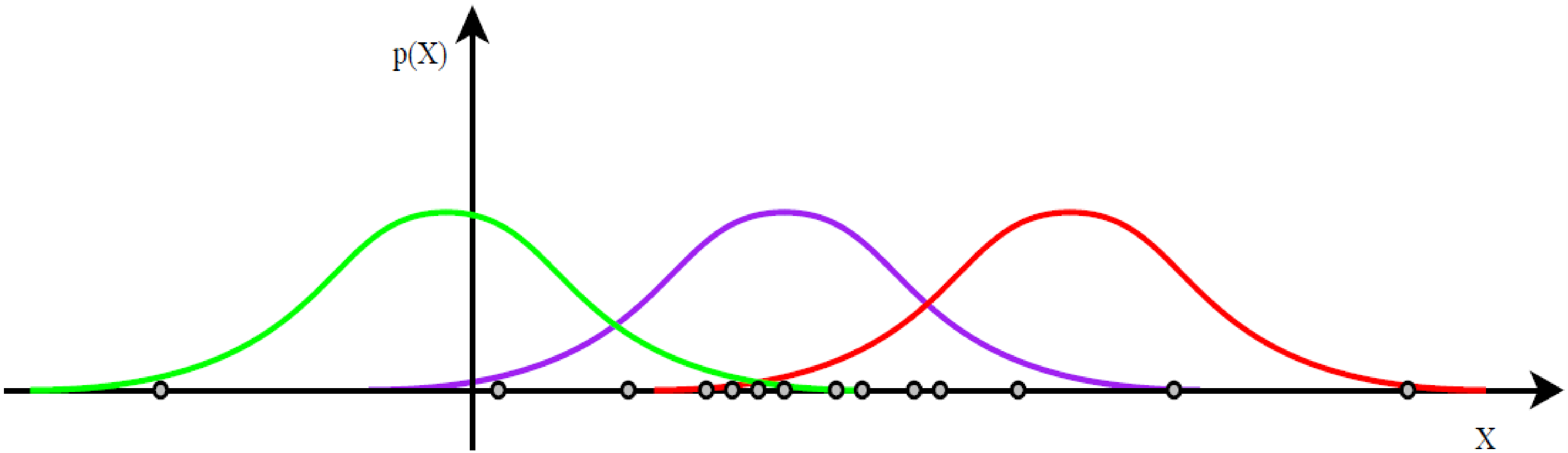
- But a more principled approach for training in classification problems is based on Maximum Likelihood
  - We want to find the parameters which maximize the likelihood function

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))$$

Maximum Likelihood flash-back (1/6)

- Suppose we observe some i.i.d. samples coming from a Gaussian distribution with known variance:

$$x_1, x_2, \ldots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$



<u>Which distribution do you prefer?</u>

Maximum Likelihood flash-back (2/6)

- There is a simple recipe for Maximum Likelihood estimation

1. Write the likelihood $L = P(Data|\theta)$ for the data
2. (Take the logarithm of likelihood $\mathcal{L} = \log P(Data|\theta)$)
3. Work out $\partial L/\partial\theta$ or $\partial\mathcal{L}/\partial\theta$ using high-school calculus
4. Solve the set of simultaneous equations $\partial\mathcal{L}/\partial\theta_i = 0$
5. Check that θ^{mle} is a maximum

- Let's try to apply it to our example

$$x_1, x_2, \ldots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

Maximum Likelihood flash-back (3/6)

- Let's try to apply it to our example

$$x_1, x_2, \ldots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

1. Write le likelihood for the data

$$L(\mu) = p(x_1, x_2, \ldots, x_N|\mu, \sigma^2) = \prod_{n=1}^{N} p(x_n|\mu, \sigma^2)$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_n-\mu)^2}{2\sigma^2}}$$

Maximum Likelihood flash-back (4/6)

- Let's try to apply it to our example

$$x_1, x_2, \dots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

2. (Take the logarithm of the likelihood -> log-likelihood)

$$\begin{aligned} \mathcal{L} &= log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ &= \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x_n-\mu)^2}{2\sigma^2}) \\ &= N(\log \frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 \end{aligned}$$

Maximum Likelihood flash-back (5/6)

- Let's try to apply it to our example

$$x_1, x_2, \ldots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

3. Work out the derivatives using high-school calculus

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mu} &= \frac{\partial}{\partial \mu} N(\log \frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu)^2 \\
&= -\frac{1}{2\sigma^2} \frac{\partial}{\partial \mu} \sum_{n=1}^{N} (x_n - \mu)^2 = \\
&= \frac{1}{2\sigma^2} \sum_{n=1}^{N} 2(x_n - \mu)
\end{aligned}$$

Maximum Likelihood flash-back (6/6)

- Let's try to apply it to our example

$$x_1, x_2, \ldots, x_K \sim N(\mu, \sigma^2) \qquad p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$$

4. Solve the unconstrained equations $\partial \mathcal{L} / \partial \theta_i = 0$

$$\frac{1}{2\sigma^2} \sum_{n=1}^{N} 2(x_n - \mu) = 0$$

$$\sum_{n=1}^{N} (x_n - \mu) = 0$$

$$\sum_{n=1}^{N} x_n = \sum_{n=1}^{N} \mu$$

$$\mu^{mle} = \frac{1}{R} \sum_{r=1}^{R} x_r$$

Training Logistic Regression (1/4)

- For the basic logistic regression wee need two parameters

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

- In principle we could use (non linear) Least Squares fitting on the observed data the corresponding model

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

- But a more principled approach for training in classification problems is based on Maximum Likelihood
  - We want to find the parameters which maximize the likelihood function

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

Training Logistic Regression (2/4)

- Let's find the parameters which maximize the likelihood function

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

- If we compute the log-likelihood for N observations

$$\ell(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta)$$

Taken from ESL

where $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$

- We obtain a log-likelihood in the form of

Can you derive it?

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{N} \Big\{ y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta)) \Big\} \\ &= \sum_{i=1}^{N} \Big\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \Big\}. \end{aligned}$$

Training Logistic Regression (3/4)

- Let's find the parameters which maximize the likelihood function

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

  - Z-statistics has the same role of the regression t-statistics, a large value means the parameter is not null
  - Intercept does not have a particular meaning is used to adjust the probability to class proportions

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

TABLE 4.1. *For the* Default *data, estimated coefficients of the logistic regression model that predicts the probability of* default *using* balance*. A one-unit increase in* balance *is associated with an increase in the log odds of* default *by* 0.0055 *units.*

Example: Default data & Logistic Regression

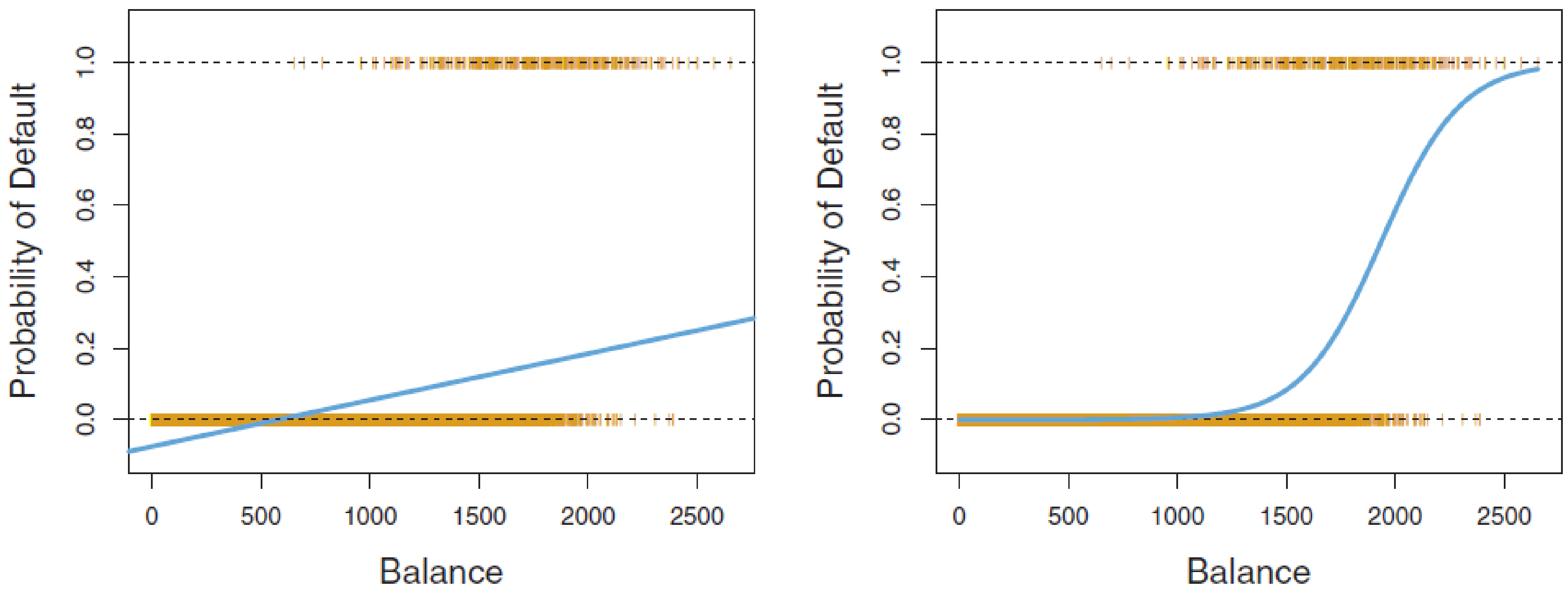


FIGURE 4.2. *Classification using the* Default *data.* Left: *Estimated probability of* default *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* default*(*No *or* Yes*).* Right: *Predicted probabilities of* default *using logistic regression. All probabilities lie between* 0 *and* 1.

Training Logistic Regression (4/4)

- Let's find the parameters which maximize the likelihood function

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

  - We can train the model using qualitative variables through the use of binary (dummy) variables

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	−3.5041	0.0707	−49.55	<0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

TABLE 4.2. *For the* `Default` *data, estimated coefficients of the logistic regression model that predicts the probability of* `default` *using student status. Student status is encoded as a dummy variable, with a value of 1 for a student and a value of 0 for a non-student, and represented by the variable* `student[Yes]` *in the table.*

Making predictions with Logistic Regression

- Once we have the model parameters we can predict the class

- The Default probability having 1000$ balance is <1%

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1,000}}{1+e^{-10.6513+0.0055\times 1,000}} = 0.00576$$

  while with a balance of 2000$ this becomes 58.6%

- With qualitative variables, i.e., dummy variables, we get that being a students results in

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292$$

- So far we have considered only one predictor, but we can extend the approach to multiple regressors

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

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

- By maximum likelihood we learn the corresponding parameters

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

What about this?

TABLE 4.3. *For the* Default *data, e… the logistic regression model that predicts the probability … nce,* income*, and* student *status. Student status is encoded as a dummy variable* student[Yes]*, with a value of 1 for a student and a value of 0 for a non-student. In fitting this model,* income *was measured in thousands of dollars.*

An apparent contradiction

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Positive

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Negative!!!

Example: Confounding in Default data set

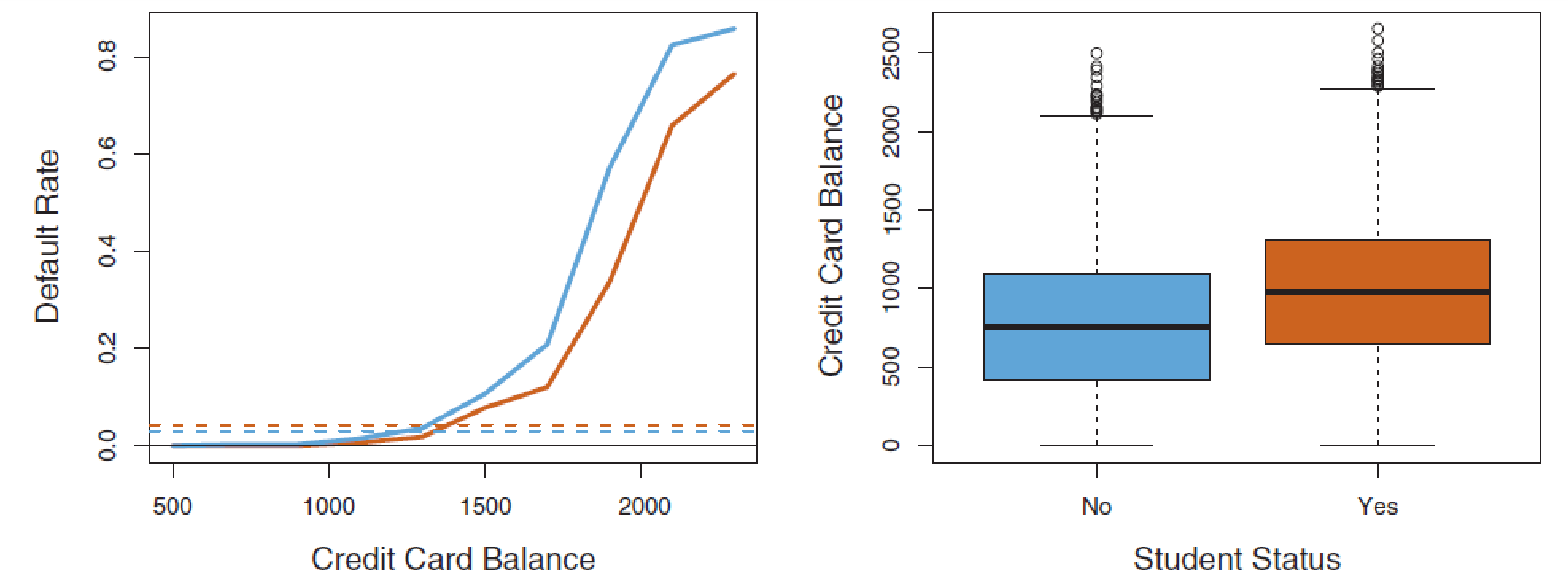


FIGURE 4.3. *Confounding in the* `Default` *data.* Left: *Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of* `balance`, *while the horizontal broken lines display the overall default rates.* Right: *Boxplots of* `balance` *for students (orange) and non-students (blue) are shown.*

Example: South African Heart Disease

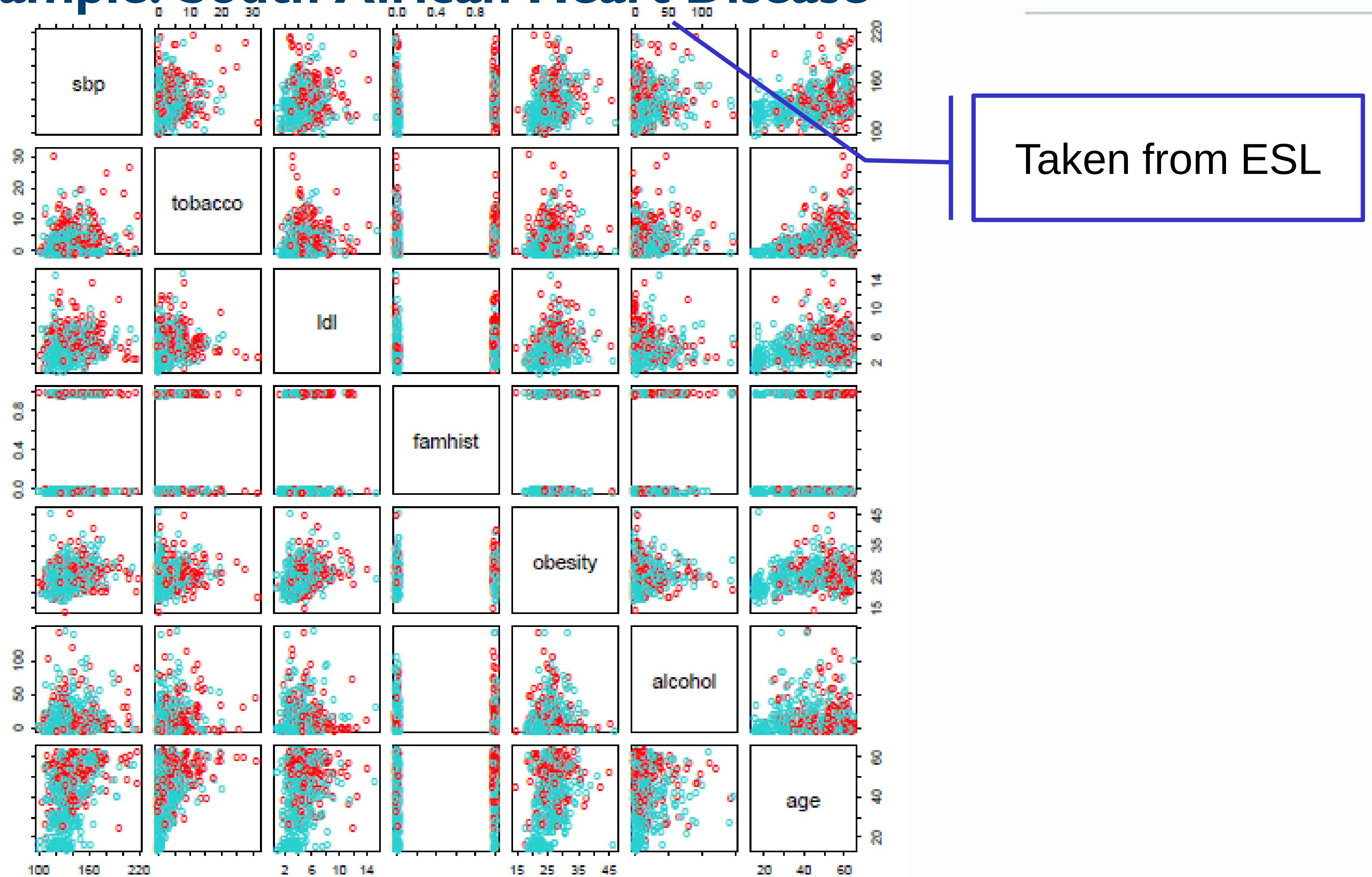


FIGURE 4.12. *A scatterplot matrix of the South African heart disease data. Each plot shows a pair of risk factors, and the cases and controls are color coded (red is a case). The variable* family history of heart disease (`famhist`) *is binary (yes or no).*

Logistic Regression for Feature Selection

Taken from ESL

- If we fit the complete model on these data we ge

TABLE 4.2. *Results from a logistic regression fit to the South Af*
disease data.

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Taken from ESL

- While if we use stepwise Logistic Regression

TABLE 4.3. *Results from stepwise logistic regression fit to South African heart disease data.*

	Coefficient	Std. Error	Z score
(Intercept)	−4.204	0.498	−8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

Logistic Regression parameters meaning

- Regression parameters represent the increment on the logit of probability given by a unitary increment of a variable

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

- Let consider the increase of *tobacco* consumption in life od 1Kg, this count for an increase in log-odds of exp(0.081)=1.084 which means an overall increase of 8.4%
- With a 95% confidence interval $\exp(0.081 \pm 2 \times 0.026) = (1.03, 1.14)$

TABLE 4.3. *Results from stepwise logistic regression fit to South African heart disease data.*

	Coefficient	Std. Error	Z score
(Intercept)	−4.204	0.498	−8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

Taken from ESL

Regularized Logistic Regression

- As for Linear Regression we can compute a "Lasso" version

$$\max_{\beta_0,\beta}\left\{\sum_{i=1}^{N}\left[y_i(\beta_0+\beta^T x_i)-\log(1+e^{\beta_0+\beta^T x_i})\right]-\lambda\sum_{j=1}^{p}|\beta_j|\right\}$$

- As for Linear Regression we can compute a “Lasso” version

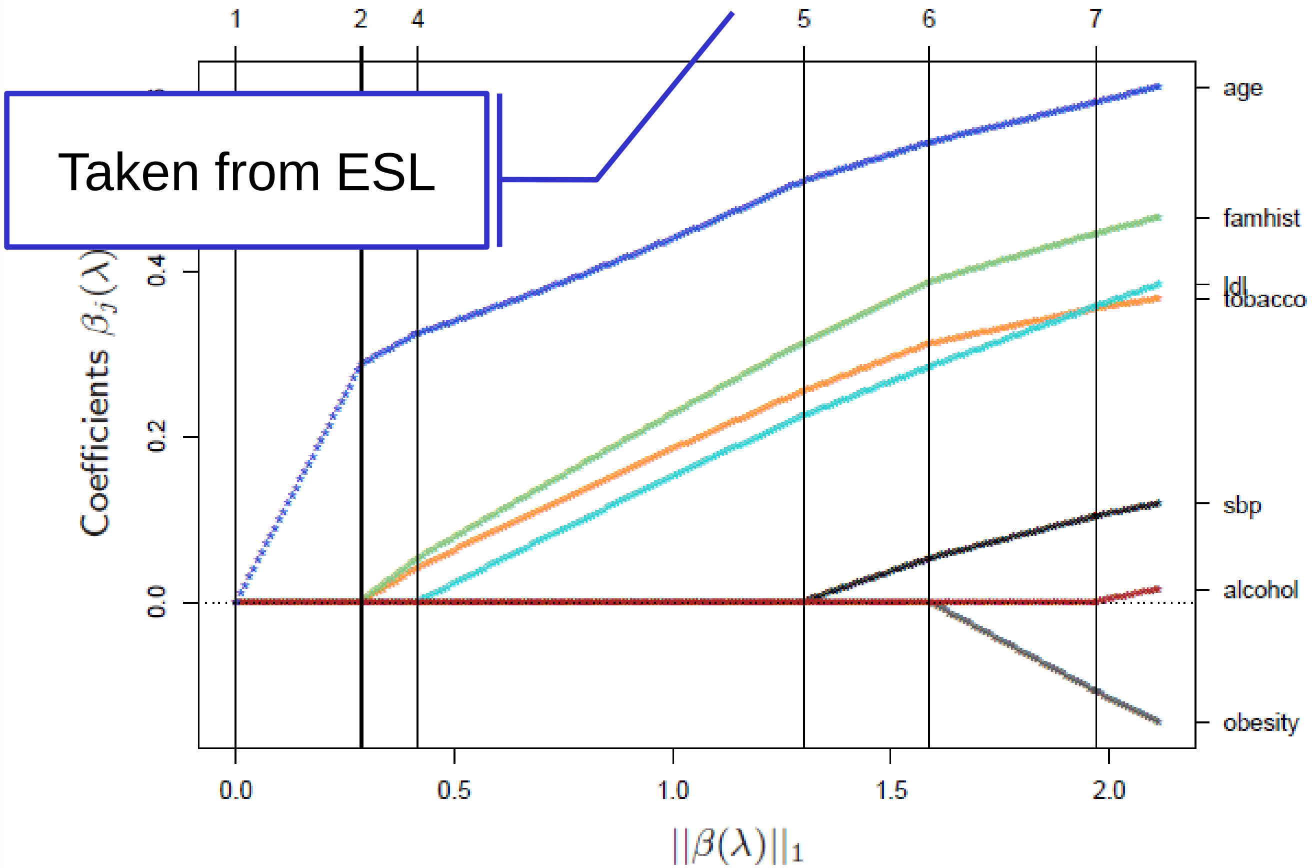


FIGURE 4.13. *L_1 regularized logistic regression coefficients for the South African heart disease data, plotted as a function of the L_1 norm. The variables were all standardized to have unit variance. The profiles are computed exactly at each of the plotted points.*

Multiclass Logistic Regression

- Logistic Regression extends naturally to multiclass problems by computing the log-odds w.r.t. the K[th] class

$$\log \frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$\log \frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$\log \frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

Comes from ESL, but it's worth knowing!!!

Notation different because it comes from ESL

- This is equivalent to

$$\Pr(G=k|X=x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T x)}, \quad k = 1, \ldots, K-1,$$
$$\Pr(G=K|X=x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T x)}, \qquad (4.18)$$

- Can you prove it ?!?!?

Wrap-up on Logistic Regression

- We model the log-odds as a linear regression model

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

- This means the posterior probability becomes

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

- Parameters represent log-odds increase per variable unit increment keeping fixed the others
- We can use it to perform feature selection using z-scores and forward stepwise selection
- The class decision boundary is linear, but points close to the boundary count more … this will be discussed later

Beyond Logistic Regression …

- Logistic Regression models directly class posterior probability

$$\Pr(Y = k|X = x)$$

- Linear Discriminant Analysis uses the Bayes Theorem

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

- What improvements come with this model?
  - Parameter learning unstable in Logistic Regressione for well separated classes
  - With little data and normal predictor distribution LDA is more stable
  - A very popular algorithm with more than 2 response classes

Linear Discriminant Analysis (1/3)

- Suppose we want to discriminate among K>2 classes
- Each class has a prior probability π_k
- Given the class we model the density function of predictors as

$$f_k(X) \equiv \Pr(X = x | Y = k)$$

- Using the Bayes Theorem we obtain

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

  - Prior probability π_k is relatively simple to learn
  - Likelihood $f_k(X)$ might be more tricky and we need some assumptions to simplify it
- If we correctly estimate the likelihood $f_k(X)$ we obtain the Bayes Classifier, i.e., the one with the smallest error rate!

Linear Discriminant Analysis (2/3)

- Let assume p=1 and use a Gaussian distribution

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

- Let assume all classes have the same covariance

$$\sigma_1^2 = \ldots = \sigma_K^2$$

- The posteriors probability as computed by LDA becomes

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

- The selected class is the one with the highest posterior which is equivalent to the highest discriminating function

Can you derive this?

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

Linear discriminant function in x

- With 2 classes having the same prior probability $\pi_1 = \pi_2$ we decide the class according to the inequality

$$2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$$

- The Bayes decision boundary corresponds to

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

- Training is as simple as estimating the model parameters

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

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

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

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

LDA Simple Example (with p=1)

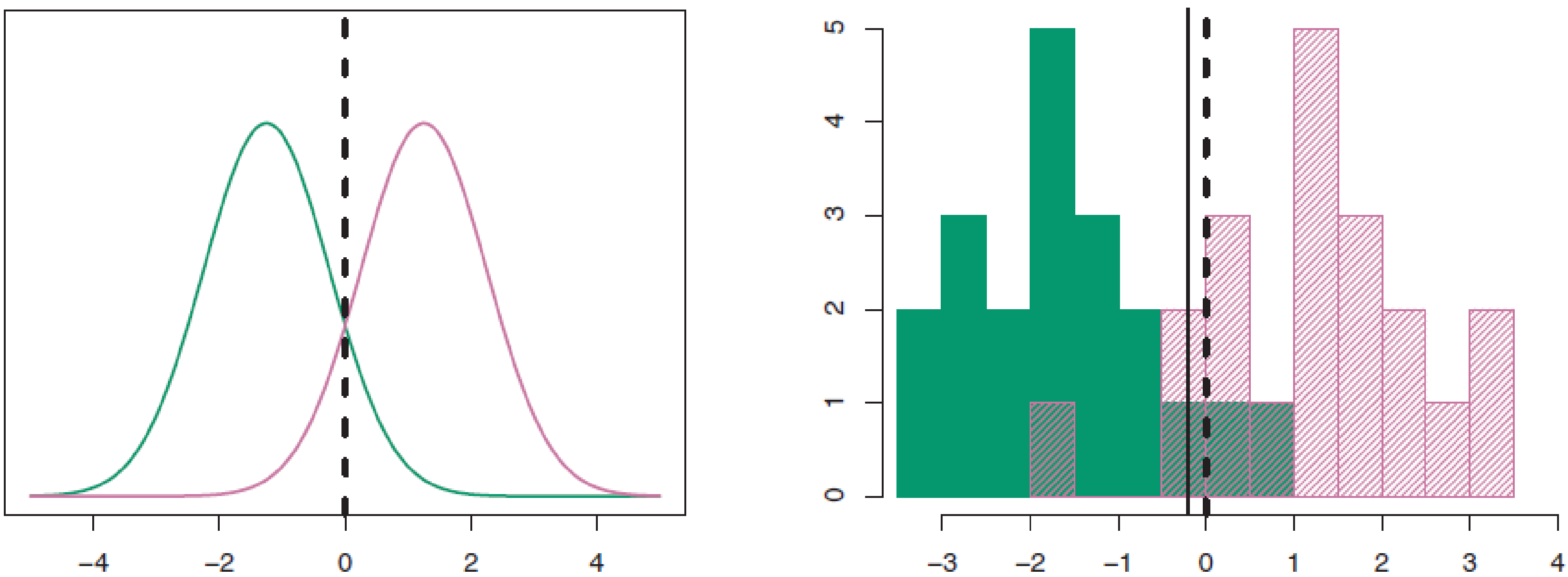


FIGURE 4.4. Left: *Two one-dimensional normal density functions are shown. The dashed vertical line represents the Bayes decision boundary.* Right: *20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is again shown as a dashed vertical line. The solid vertical line represents the LDA decision boundary estimated from the training data.*

Linear Discriminant Analysis with p>1 (1/3)

- In case p>1 we assume $X = (X_1, X_2, \ldots, X_p)$ comes from

$$f(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \boldsymbol{\Sigma}^{-1}(x-\mu)\right)$$

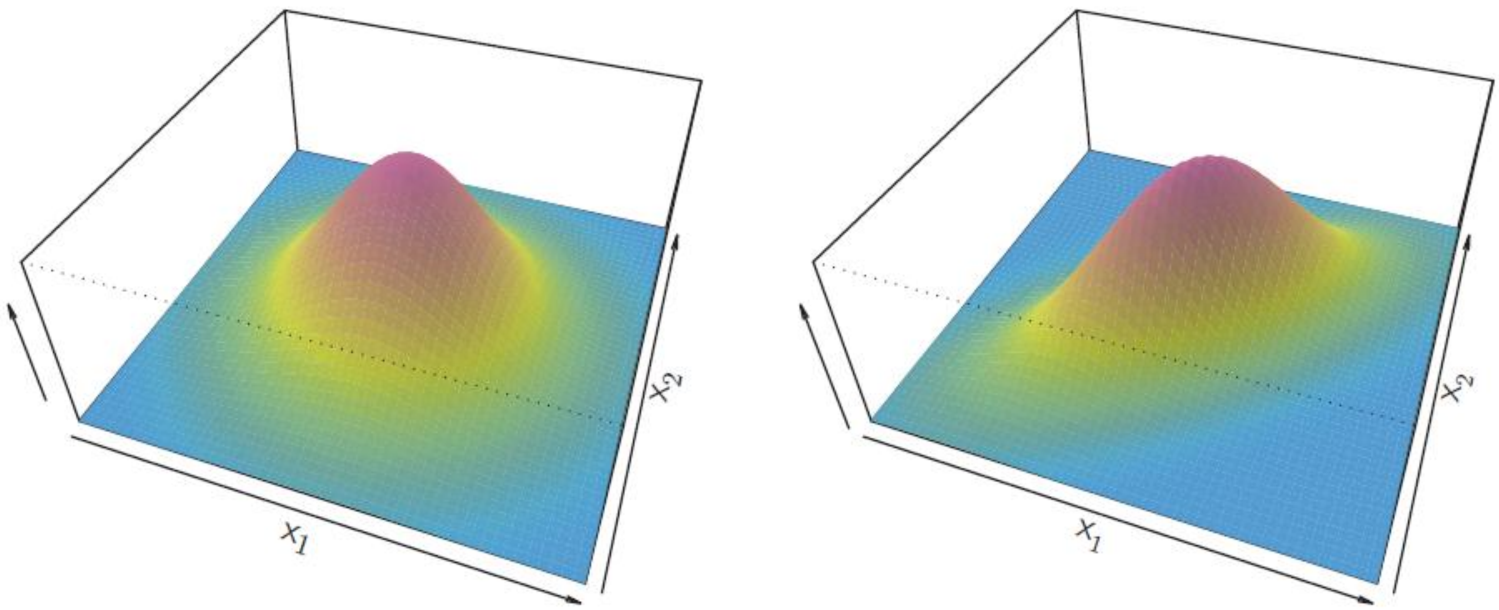


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

Linear Discriminant Analysis with p>1 (2/3)

- In the case of p>1 the LDA classifier assumes
  - Observations from the k-th class are drawn from $N(\mu_k, \Sigma)$
  - The covariance structure is common to all classes
- The Bayes discriminating function becomes

Can you derive this?

Still linear in x!!!

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

- From this we can compute the boundary between each class (considering the two classes having the same prior probability)

$$x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k = x^T \Sigma^{-1} \mu_l - \frac{1}{2} \mu_l^T \Sigma^{-1} \mu_l$$

- Training formulas for the LDA parameters are similar to the case of p=1 …

Linear Discriminant Analysis Example

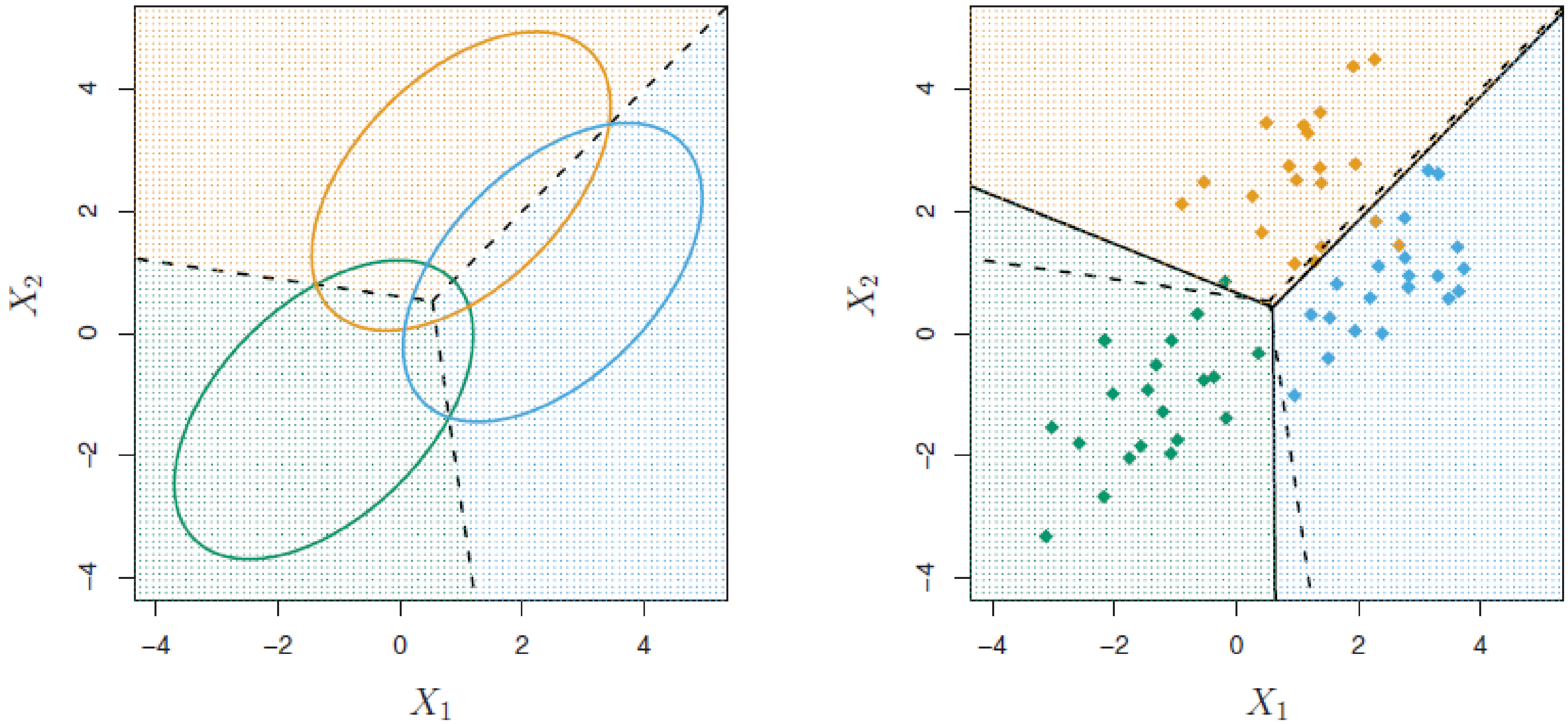


FIGURE 4.6. *An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.* Left: *Ellipses that contain 95 % of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries.* Right: *20 observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.*

Example: LDA on the Default Dataset

- LDA on the Default dataset gets 2.75% training error rate
  - Having 10000 records and p=3 we do not expect much overfitting … by the way how many parameters we have?
  - Being 3.33% the number of defaulters a dummy classifier would get a similar error rate

$252/333 = 75.7\,\%$

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

TABLE 4.4. *A confusion matrix compares the LDA predictions to the true default statuses for the 10,000 training observations in the* `Default` *data set. Elements on the diagonal of the matrix represent individuals whose default statuses were correctly predicted, while off-diagonal elements represent individuals that were misclassified. LDA made incorrect predictions for 23 individuals who did not default and for 252 individuals who did default.*

On the performance of LDA

- Errors in classification are often reported as a Confusion Matrix

		True default status		
99.8 %		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

24.3 %

  - Sensitivity: percentage of true defaulters
  - Specificity: percentage of non-defaulters correctly identified

- The Bayes classifier optimize the overall error rate independently from the class they belong to an it does this by thresholding

$$\Pr(\texttt{default} = \texttt{Yes}|X = x) > 0.5$$

- Can we improve on this?

Example: Increasing LDA Sensitivity

- We might want to improve classifier sensitivity with respect to a given class because we consider it more "critical"

$$P(\texttt{default} = \texttt{Yes} | X = x) > 0.2$$

  - Reduced "Default" error rate from 75.7% to 41.4%
  - Increased overall error of 3.73% (but it is worth)

		True default status		
		No	Yes	Total
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
	Total	9,667	333	10,000

TABLE 4.5. *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 %.*

Tweaking LDA sensitivity

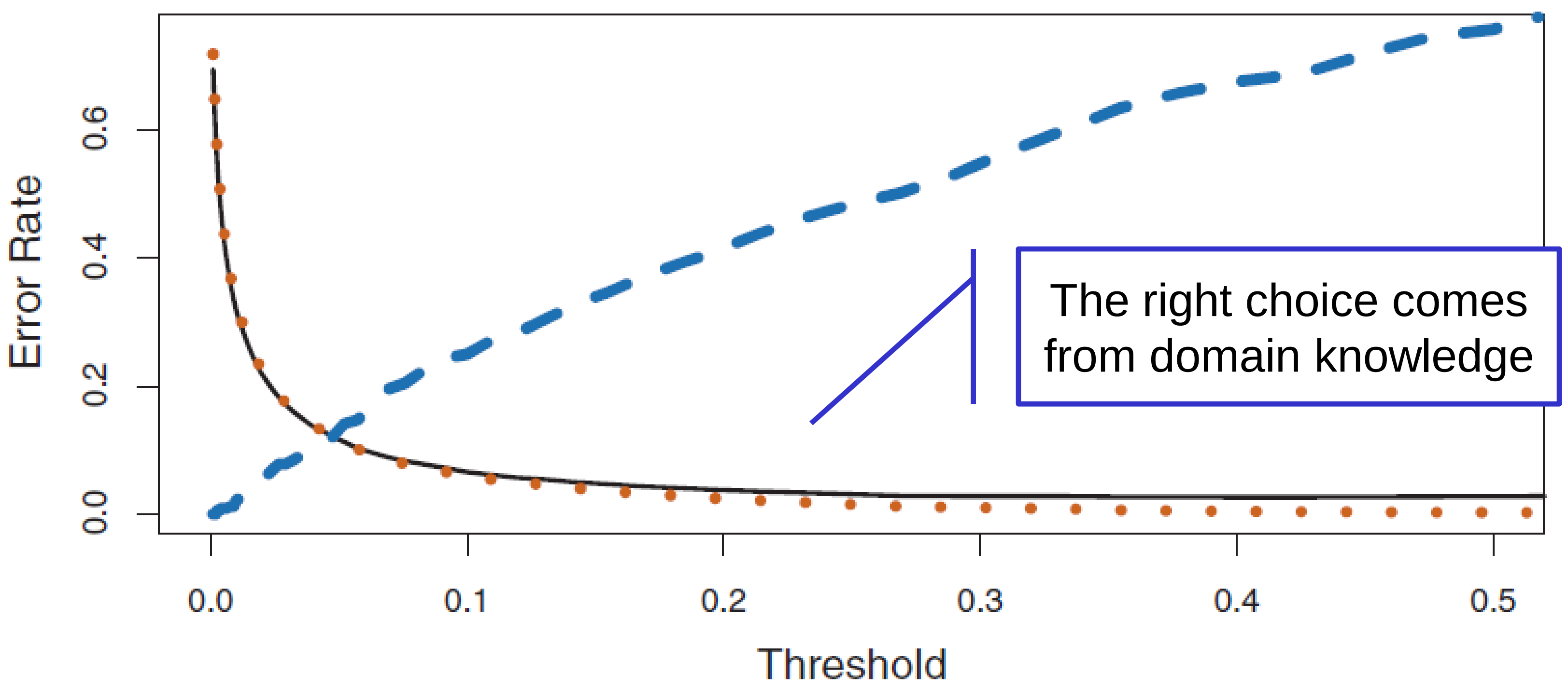


FIGURE 4.7. *For the* Default *data set, error rates are shown as a function of the threshold value for the posterior probability that is used to perform the assignment. The black solid line displays the overall error rate. The blue dashed line represents the fraction of defaulting customers that are incorrectly classified, and the orange dotted line indicates the fraction of errors among the non-defaulting customers.*

ROC Curve

- The ROC (Receiver Operating Characteristics) summarizes false positive and false negative errors

- Obtained by testing all possible thresholds
  - Overall performance given by Area Under the ROC Curve
  - A classifier which randomly guesses (with two classes) has an AUC = 0.5 a perfect classifier has AUC = 1

- ROC curve considers true positive and false positive rates
  - Sensitivity is equivalent to true positive rate
  - Specificity is equivalent to 1 – false positive rate

ROC Curve for LDA on Default data

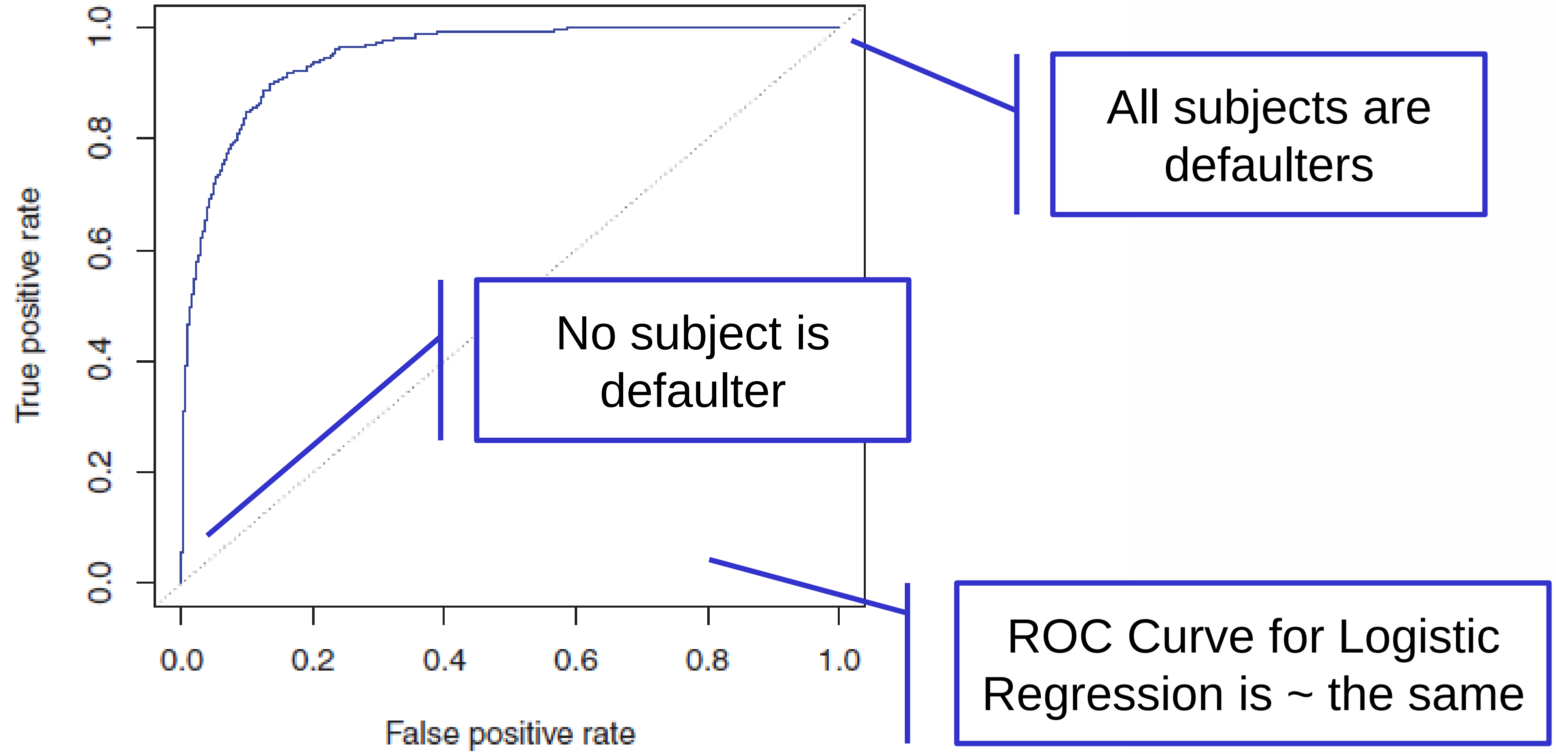


FIGURE 4.8. *A ROC curve for the LDA classifier on the* `Default` *data. It traces out two types of error as we vary the threshold value for the posterior probability of default. The actual thresholds are not shown. The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.*

Clearing out terminology

- When applying a classifier we can obtain

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

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

- Out of this we can define the following

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.*

Quadratic Discriminant Analysis

- Linear Discriminant Analysis assumes all classes having a common covariance structure
- Quadratic Discriminant Analysis assumes different covariances

$$X \sim N(\mu_k, \mathbf{\Sigma}_k)$$

- Under this hypothesis the Bayes discriminant function becomes

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k\end{aligned}$$

Can you derive this?

Quadratic function

- The decision LDA vs. QDA boils down to bias-variance trade-off
  - QDA requires $Kp(p+1)/2$ parameters while LDA only Kp

QDA vs LDA Example

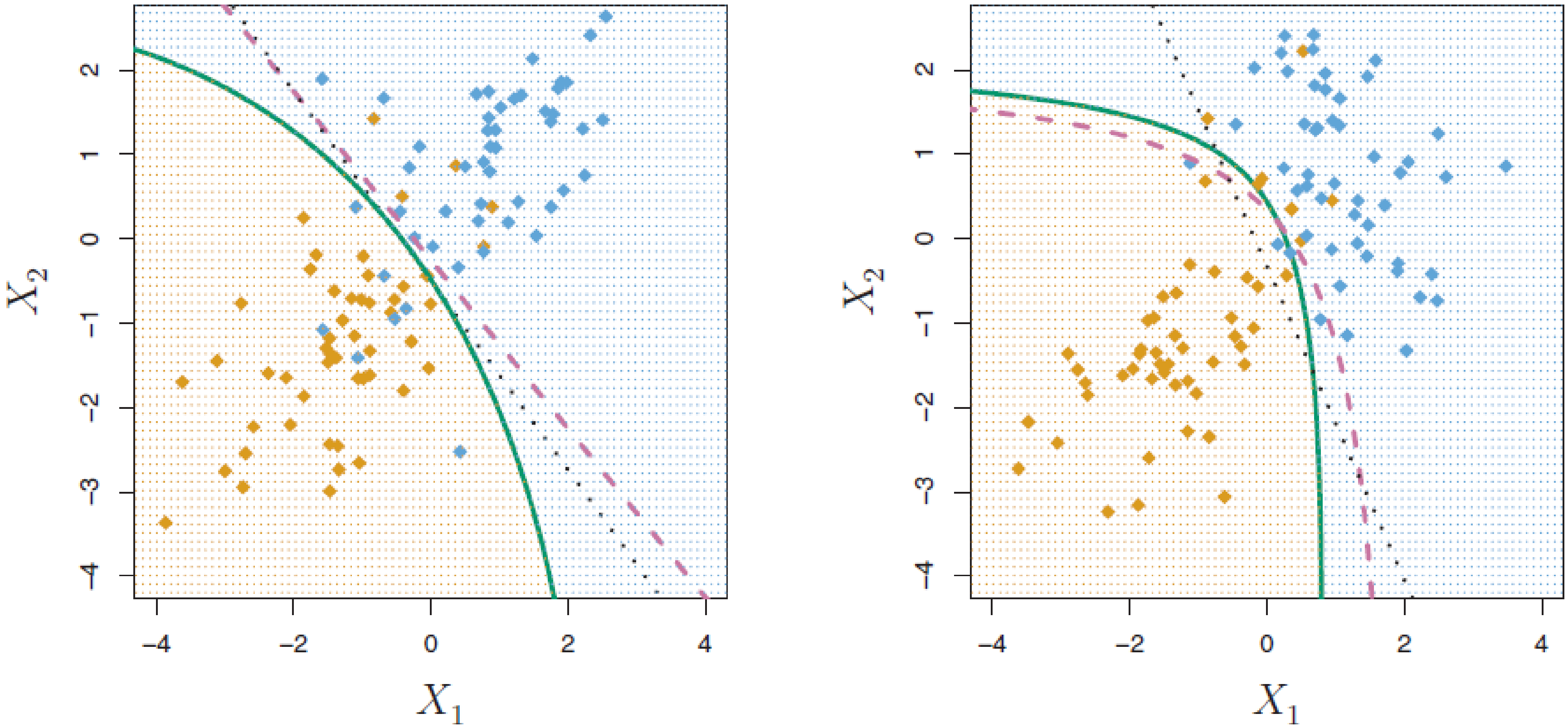


FIGURE 4.9. Left: *The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with* $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$*. The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.* Right: *Details are as given in the left-hand panel, except that* $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$*. Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.*

Which classifier is better?

- Let consider 2 classes and 1 predictor
  - It can be seen that for LDA the log odds is given by

Can you derive this?

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x$$

  - While for Logistic Regression the log odds is

$$\log\left(\frac{p_1}{1-p_1}\right) = \beta_0 + \beta_1 x$$

  - Both linear functions but learning procedures are different …

- Linear Discriminant Analysis is the Optimal Bayes if its hypothesis holds otherwise Logistic Regression outperforms it!
- Quadratic Discriminant Analysis is to be preferred if the class covariances are different and we have a non linear boundary

Some scenarios tested …

- Linear Baoundary Scenarios
  1. Samples from 2 uncorrelated normal distributions
  2. Samples from 2 slightly correlated normal distributions
  3. Samples from t-student distributed classes

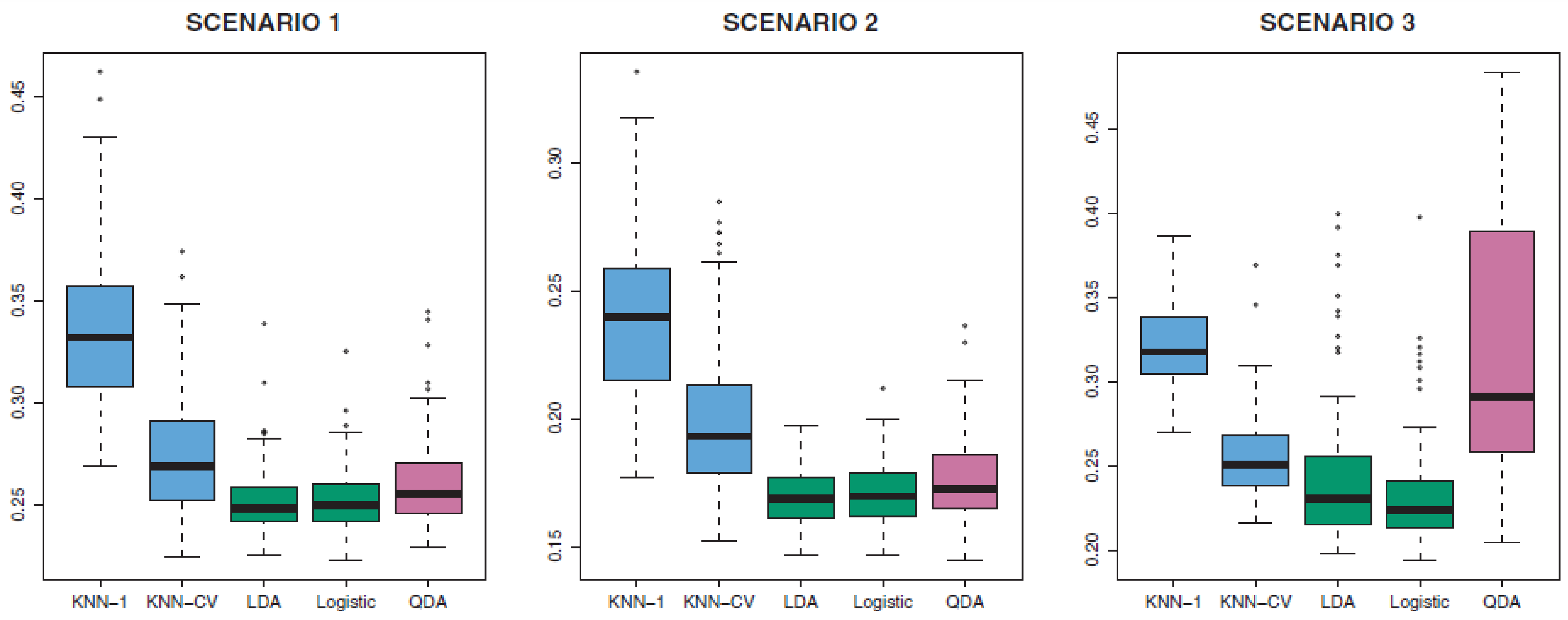


FIGURE 4.10. *Boxplots of the test error rates for each of the linear scenarios described in the main text.*

Some other scenarios tested …

- Non Linear Boundary Scenarios
  4. Samples from 2 normal distribution with different correlation
  5. Samples from 2 normals, predictors are quadratic functions
  6. As previous but with a more complicated function

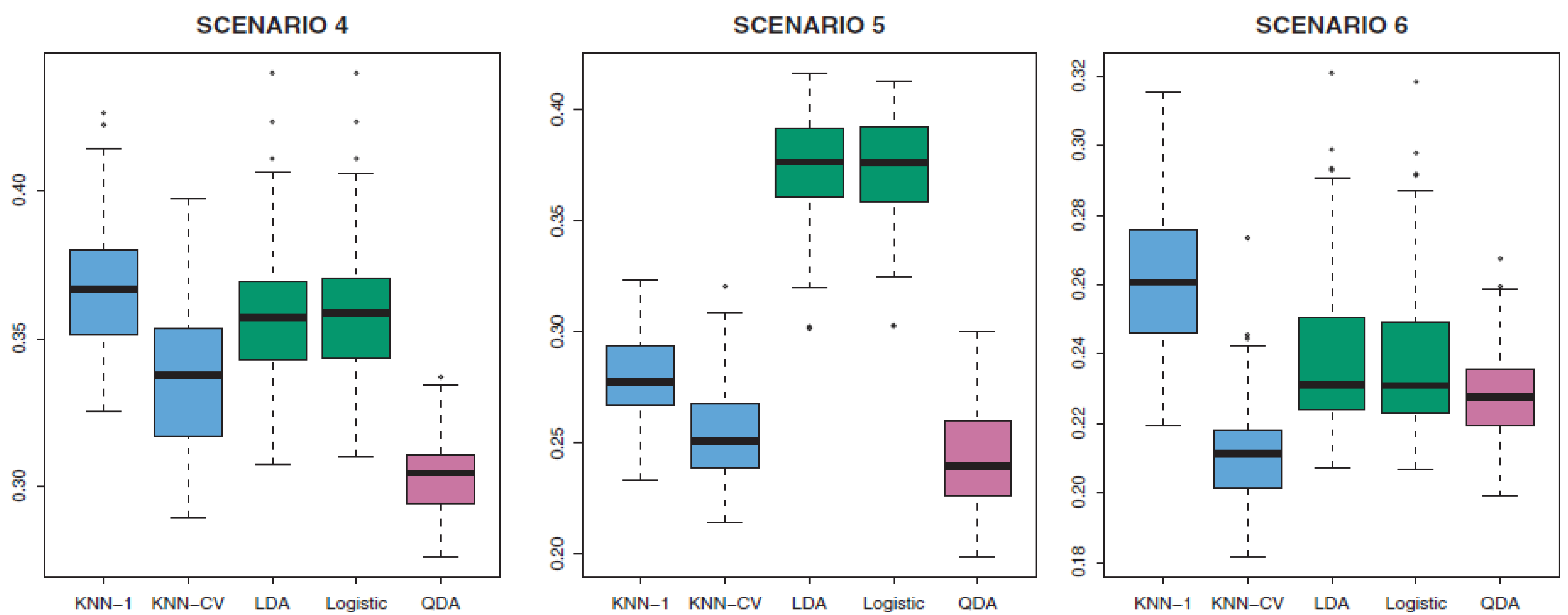


FIGURE 4.11. *Boxplots of the test error rates for each of the non-linear scenarios described in the main text.*

Overall conclusion on the comparison

- No method is better than all the others
  - In the decision boundary is linear then LDA and Logistic Regression are those performing better
  - When the decision boundary is moderately non linear QDA may give better results
  - For much complex decision boundaries non parametric approaches such as KNN perform better, but the right level of smoothness has to be chosen