University of BRISTOL

Linear classification

Linear discriminant analysis and logistic regression

Henry W J Reeve

henry.reeve@bristol.ac.uk

Include EMATM0061 in the subject of your email.

Statistical Computing & Empirical Methods (EMATM0061)

MSc in Data Science, Teaching block 1, 2021.

What will we cover today?

- We will introduce the concept of a linear classifier with two examples.

- Linear discriminant analysis:

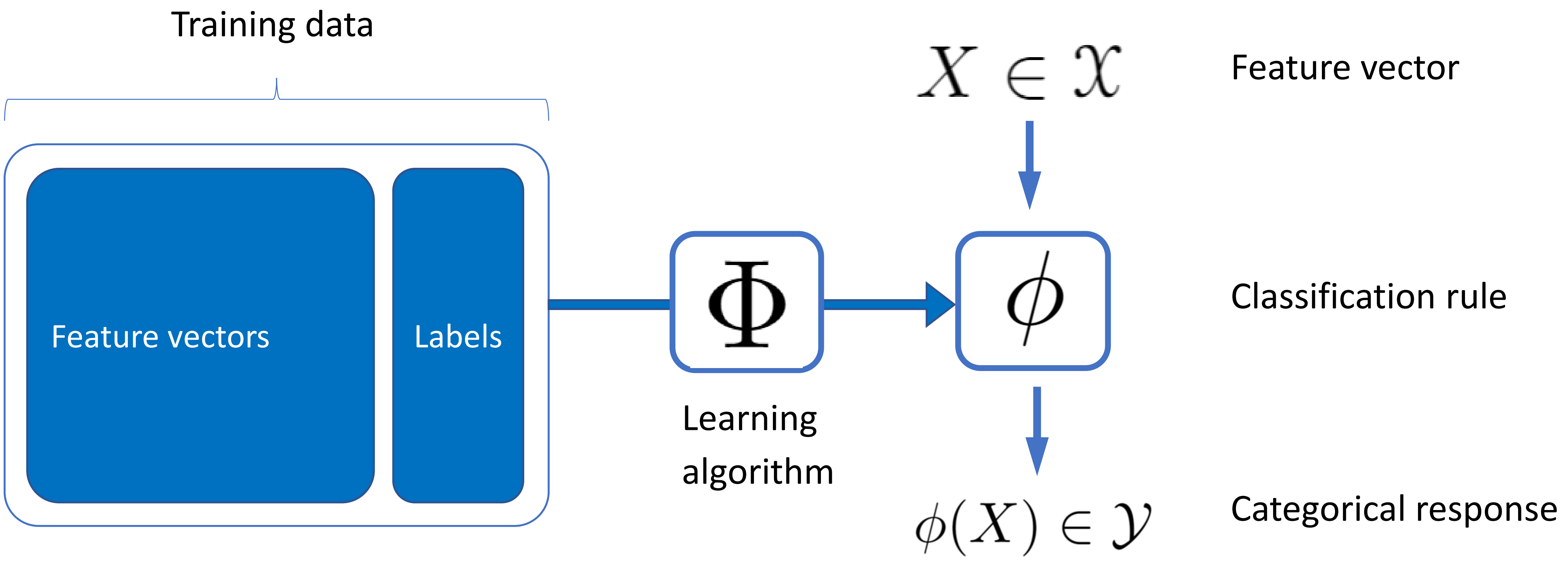
  LDA models the joint distributions as a mixture of Gaussians.

- Logistic regression:

  Logistic regression models the class-conditional distribution directly.

Classification

In the previous lecture we introduced the classification pipe-line.



In this lecture we will consider learning algorithms for linear classification.

Linear classifiers

Let's suppose we want to learn a binary classifier $\phi : \mathcal{X} \to \{0, 1\}$.

Let's suppose that our feature space $\mathcal{X} = \mathbb{R}^d$ has d continuous features,

$$X = (X^1, \cdots, X^d) \in \mathcal{X} = \mathbb{R}^d$$

<u>Example</u>: Penguin classification

Predict penguin species based on $X = (X^1, X^2) \in \mathcal{X} = \mathbb{R}^2$ where,

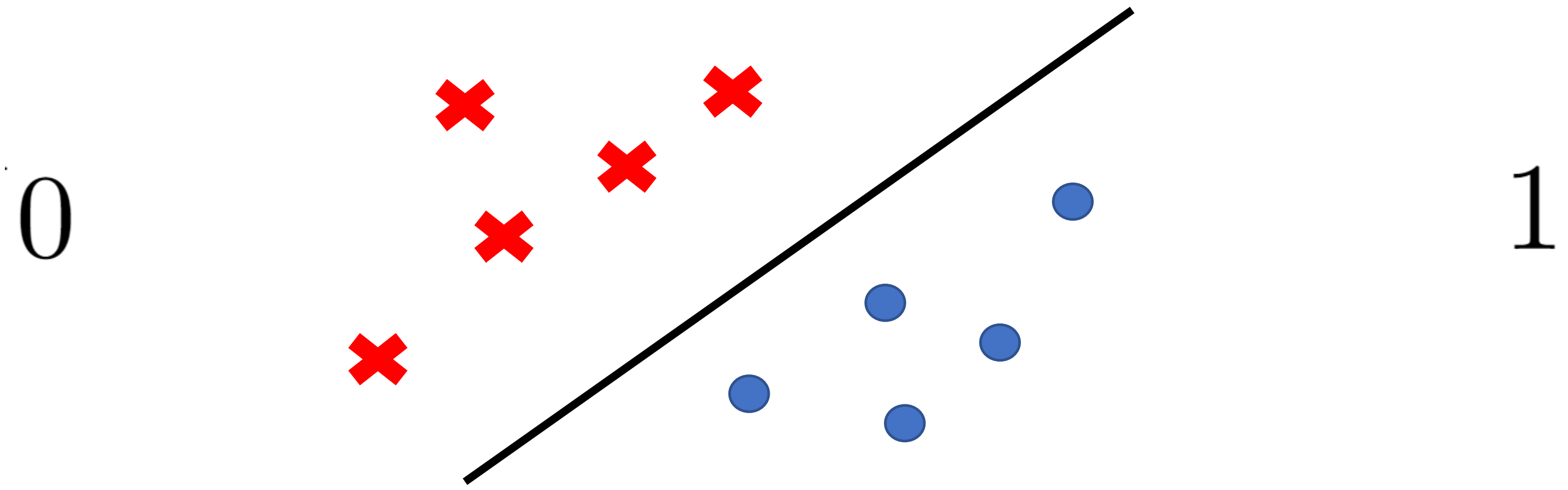
$X^1 =$ the weight of the penguin (grams).

$X^2 =$ the flipper length of the penguin (mm).

Linear classifiers

Suppose we have d continuous features $X = (X^1, \cdots, X^d) \in \mathcal{X} = \mathbb{R}^d$

A linear classifier $\phi : \mathcal{X} \to \{0, 1\}$ cuts the feature space into two with a linear hyper-plane.



If d=2 then the feature space is divided by a line.

In general, the feature space is divided by a (d-1)-dimensional hyper-pane called the decision boundary.

Linear classifiers

Suppose we have d continuous features $X = (X^1, \cdots, X^d) \in \mathcal{X} = \mathbb{R}^d$

A linear classifier $\phi : \mathcal{X} \to \{0, 1\}$ is of the form

$$\phi(x) = \begin{cases} 1 & \text{if } w^0 + w^1 \cdot x^1 + \cdots + w^d \cdot x^d \geq 0 \\ 0 & \text{if } w^0 + w^1 \cdot x^1 + \cdots + w^d \cdot x^d < 0. \end{cases}$$

Weights $w = (w^1, \cdots, w^d) \in \mathbb{R}^d$

Bias $w^0 \in \mathbb{R}$

Linear classifiers

Suppose we have d continuous features $X = (X^1, \cdots, X^d) \in \mathcal{X} = \mathbb{R}^d$

A linear classifier $\phi : \mathcal{X} \to \{0, 1\}$ is of the form

$$\phi(x) = \begin{cases} 1 & \text{if } w^0 + w^1 \cdot x^1 + \cdots + w^d \cdot x^d \geq 0 \\ 0 & \text{if } w^0 + w^1 \cdot x^1 + \cdots + w^d \cdot x^d < 0. \end{cases}$$

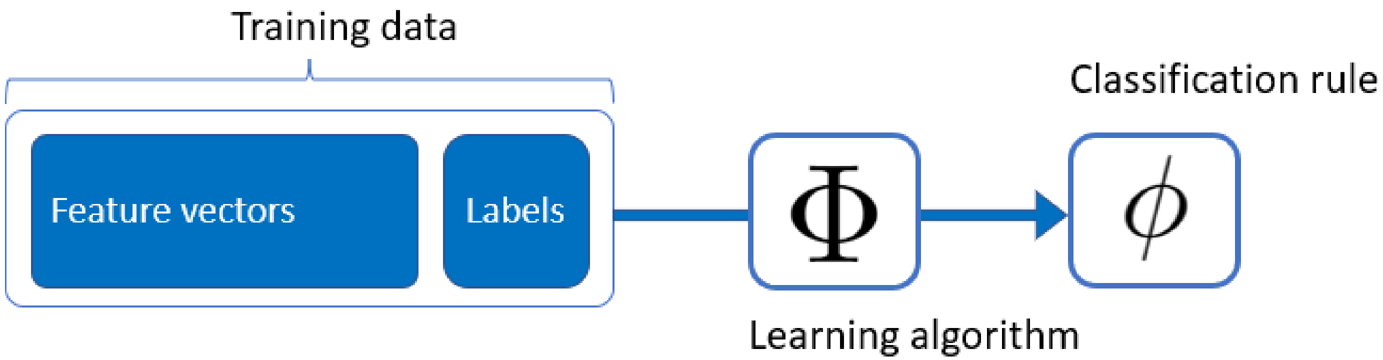
with weights $w = (w^1, \cdots, w^d) \in \mathbb{R}^d$ and a bias $w^0 \in \mathbb{R}$

We can rewrite this as $\phi(x) = \mathbb{1}\left\{w\ x^\top + w_0 \geq 0\right\}$

where $(a^1, \cdots, a^d)\ (b^1, \cdots, b_d)^\top = a^1 \cdot b^1 + \cdots + a^d b^d.$

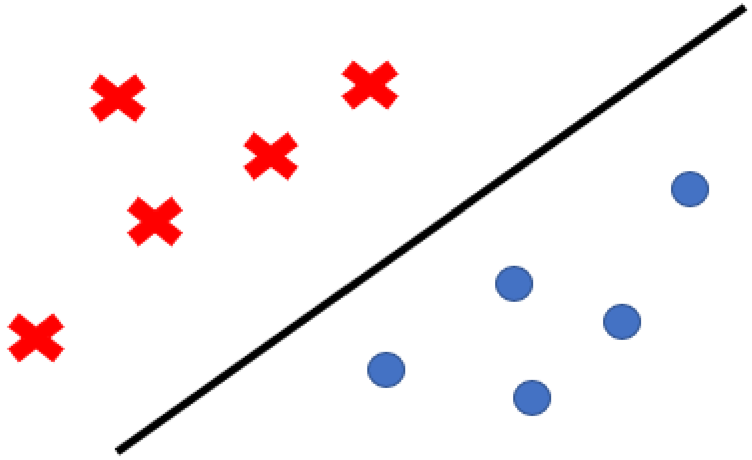
Learning algorithms for linear classifiers

Linear classifiers are a type of classification rule $\phi(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$



What's a good learning algorithm for linear classification?

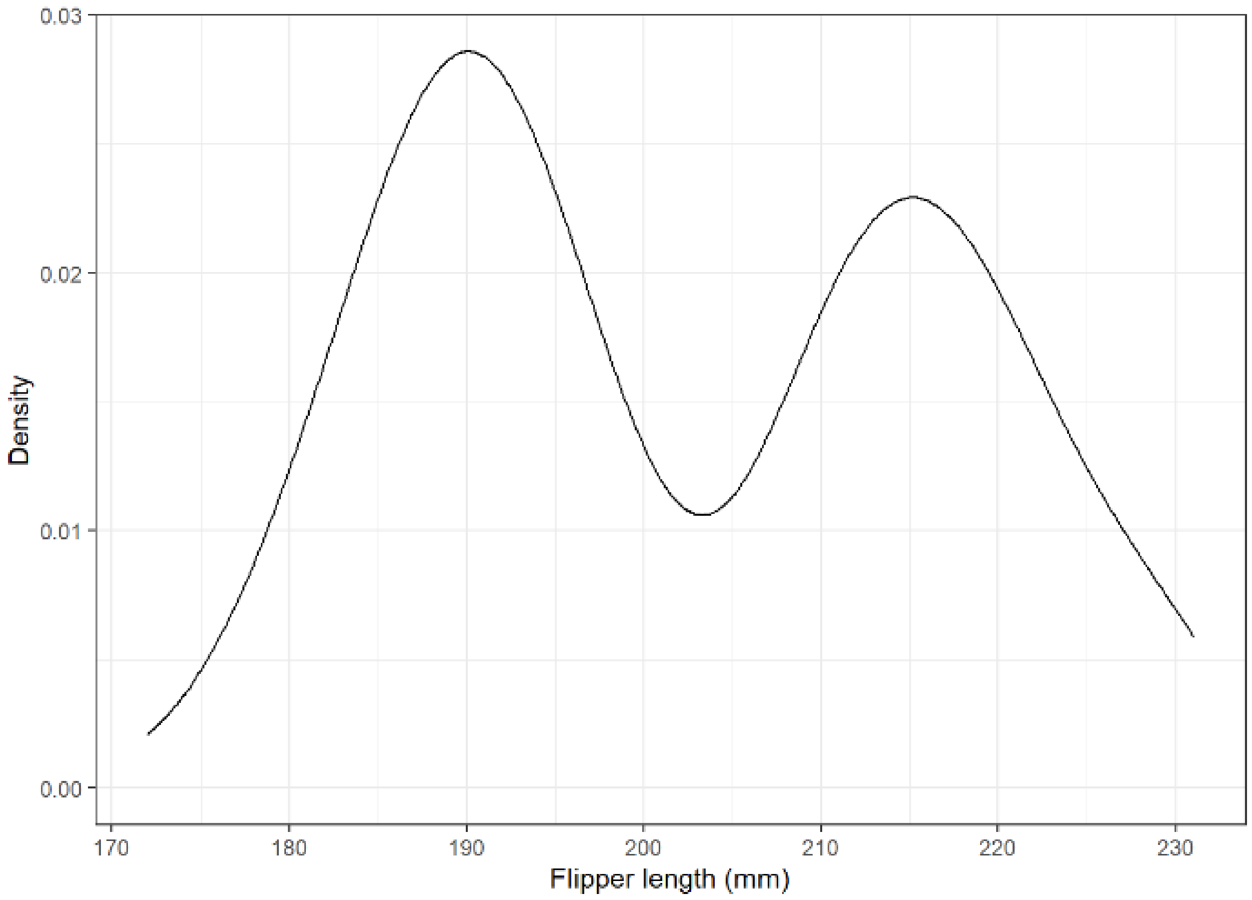
- Linear discriminant analysis
- Logistic regression
- others…



Now take a break!

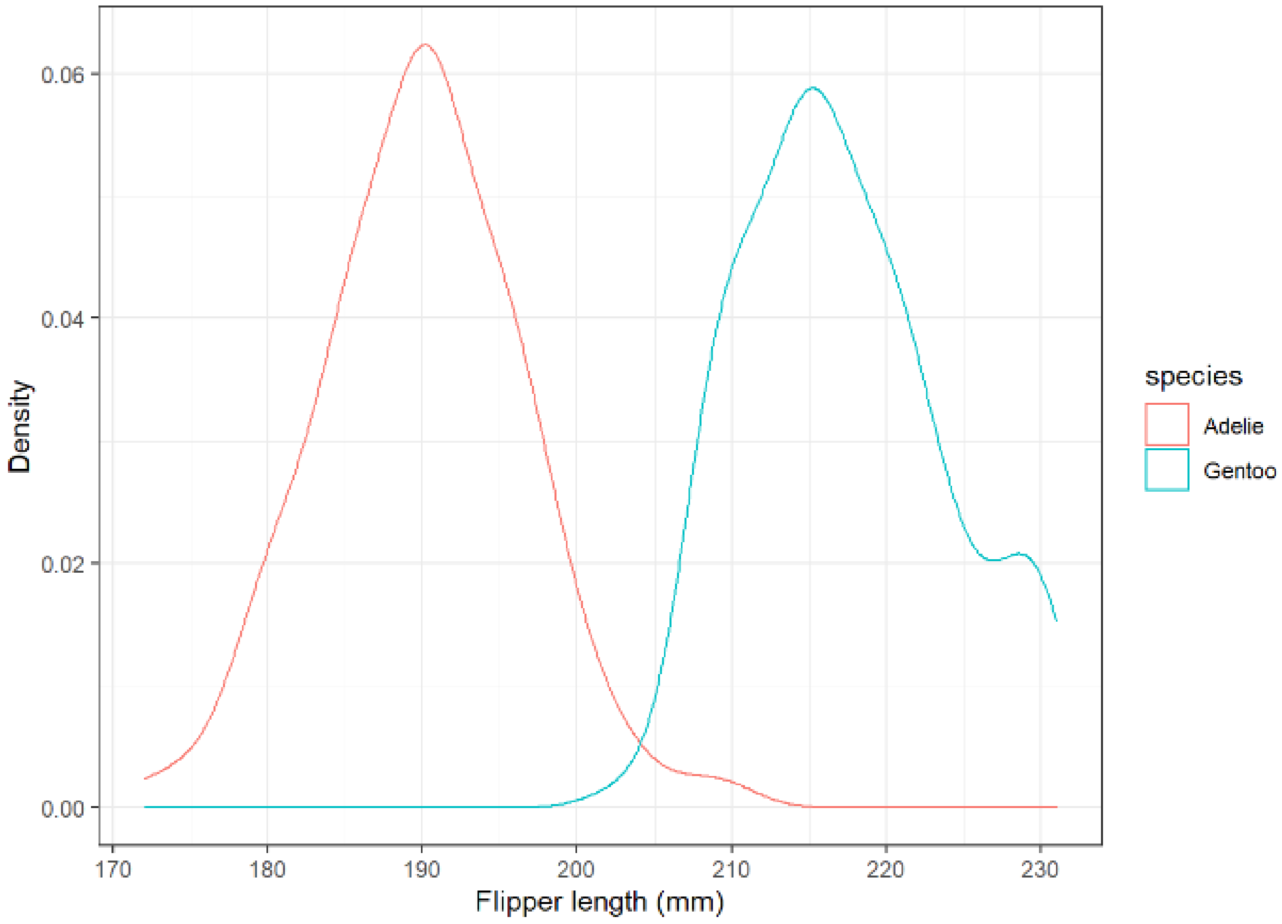
Linear discriminant analysis

Suppose we want to learn a classifier to distinguish between Gentoo and Adelie penguins.



Linear discriminant analysis

Suppose we want to learn a classifier to distinguish between Gentoo and Adelie penguins.



The data associated with individual

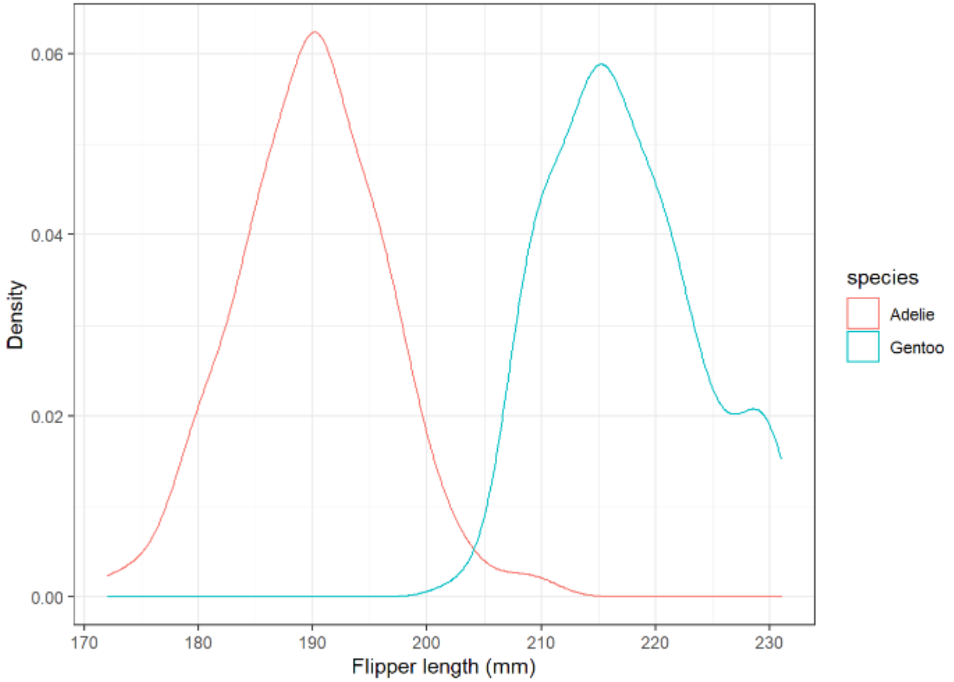
classes looks quite Gaussian.

Linear discriminant analysis

The central idea within linear discriminant analysis is:

Let's fit a Gaussian distribution to the data from each of the two classes.

We can then use this probabilistic model to generate a rule based on the probability of the two classes.



Review 1: Multivariate Gaussians

Let's remember multivariate Gaussian random variables $X \sim \mathcal{N}(\mu, \Sigma)$

Its parameters are a) A mean vector $\mu = \mathbb{E}[X] \in \mathbb{R}^d$

b) A covariance matrix $\Sigma = \mathbb{E}\left[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^\top\right] \in \mathbb{R}^{d \times d}.$

The probability density function $f_{\mu,\Sigma} : \mathbb{R}^d \to (0, \infty)$ is given by

$$f_{\mu,\Sigma}(x) := \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)\right).$$

This generalizes the idea of a univariate Gaussian.

Review 2: A Bernoulli random variable

Recall that a **Bernoulli** random variable Y is a have a binary random variable in $\{0, 1\}$

We can think of a Bernoulli random variable as corresponding to a biased coin flip.

The Bernoulli distribution is determined by a single parameter $q \in [0, 1]$

We write $Y \sim \mathcal{B}(q)$ to mean Y is a Bernoulli random variable with

$$\mathbb{P}(Y = 1) = q$$

$$\mathbb{P}(Y = 0) = 1 - q.$$

The linear discriminant analysis model

Aim: Build a probabilistic model for the data generation process for $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$.

We model $Y \in \{0, 1\}$ and $X \in \mathbb{R}^d$ as a Bernoulli followed by a Gaussian:

$Y \sim \mathcal{B}(q)$

$X \sim \mathcal{N}(\mu_Y, \Sigma)$

The linear discriminant analysis model

Aim: Build a probabilistic model for the data generation process for $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$.

1. The binary labels $Y \in \{0, 1\}$ are modelled as Bernoulli random variables.

$$Y \sim \mathcal{B}(q) \quad \text{for some fixed} \quad q \in [0, 1]$$

2. The feature vectors $X \in \mathbb{R}^d$ are modelled as class-conditional Gaussians,

$$X \sim \mathcal{N}(\mu_0, \Sigma) \text{ if } Y = 0 \qquad \text{where} \quad \mu_0, \mu_1 \in \mathbb{R}^d$$

$$X \sim \mathcal{N}(\mu_1, \Sigma) \text{ if } Y = 1 \qquad \text{and} \quad \Sigma \in \mathbb{R}^{d \times d}$$

The linear discriminant analysis model

Aim: Build a probabilistic model for the data generation process for $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$.

1. The binary labels $Y \in \{0,1\}$ are modelled as Bernoulli random variables.

$$\mathbb{P}(Y = y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

2. The feature vectors $X \in \mathbb{R}^d$ are modelled as class-conditional Gaussians,

$$\mathbb{P}(X = x | Y = y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2} (x - \mu_y) \Sigma^{-1} (x - \mu_y)^\top \right).$$

The optimal classifier for the LDA model

Let's determine the optimal classifier for the linear discriminant analysis probabilistic model.

The **Bayes classifier** $\phi^* : \mathcal{X} \to \mathcal{Y}$ minimizes the test error over all possible classifiers

$$\mathcal{R}(\phi^*) = \min \{\mathbb{P}(\phi(X) \neq Y) : \quad \phi : \mathcal{X} \to \mathcal{Y} \text{ is a classifier}\}.$$

Recall that the Bayes classifier can be defined as follows,

$$\phi^*(x) := \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x) \\ 0 & \text{if} \quad \mathbb{P}(Y = 0|X = x) > \mathbb{P}(Y = 1|X = x). \end{cases}$$

The optimal classifier for the LDA model

The **Bayes classifier** $\phi^* : \mathcal{X} \to \mathcal{Y}$ minimizes the test error over all possible classifiers

$$\mathcal{R}(\phi^*) = \min \left\{ \mathbb{P}(\phi(X) \neq Y) : \quad \phi : \mathcal{X} \to \mathcal{Y} \text{ is a classifier} \right\}.$$

Recall that the Bayes classifier can be defined as follows,

$$\phi^*(x) := \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y = 1 | X = x) \geq \mathbb{P}(Y = 0 | X = x) \\ 0 & \text{if} \quad \mathbb{P}(Y = 0 | X = x) > \mathbb{P}(Y = 1 | X = x). \end{cases}$$

Hence, $\phi^*(x) = 1$,

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1 | X = x) \geq \mathbb{P}(Y = 0 | X = x)$$

The optimal classifier for the LDA model

Remember that by Bayes theorem,

$$\mathbb{P}(Y = y|X = x) = \frac{\mathbb{P}(X = x|Y = y) \cdot \mathbb{P}(Y = y)}{\mathbb{P}(X = x)}.$$

The optimal classifier for the LDA model

Remember that by Bayes theorem,

$$\mathbb{P}(Y = y|X = x) = \frac{\mathbb{P}(X = x|Y = y) \cdot \mathbb{P}(Y = y)}{\mathbb{P}(X = x)}.$$

Note that $\phi^*(x) = 1$,

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

The optimal classifier for the LDA model

Remember that by Bayes theorem,

$$\mathbb{P}(Y = y|X = x) = \frac{\mathbb{P}(X = x|Y = y) \cdot \mathbb{P}(Y = y)}{\mathbb{P}(X = x)}.$$

Hence, $\phi^*(x) = 1$,

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

$$\Longleftrightarrow \quad \frac{\mathbb{P}(X = x|Y = 1) \cdot \mathbb{P}(Y = 1)}{\mathbb{P}(X = x)} \geq \frac{\mathbb{P}(X = x|Y = 0) \cdot \mathbb{P}(Y = 0)}{\mathbb{P}(X = x)}$$

The optimal classifier for the LDA model

Remember that by Bayes theorem,

$$\mathbb{P}(Y = y | X = x) = \frac{\mathbb{P}(X = x | Y = y) \cdot \mathbb{P}(Y = y)}{\mathbb{P}(X = x)}.$$

Hence, $\phi^*(x) = 1$,

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1 | X = x) \geq \mathbb{P}(Y = 0 | X = x)$$

$$\Longleftrightarrow \quad \frac{\mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1)}{\mathbb{P}(X = x)} \geq \frac{\mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0)}{\mathbb{P}(X = x)}$$

$$\Longleftrightarrow \quad \mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0)$$

The optimal classifier for the LDA model

Within the linear discriminant model we have,

$$\mathbb{P}(Y = y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

$$\mathbb{P}(X = x|Y = y) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu_y)\Sigma^{-1}(x - \mu_y)^\top\right).$$

The optimal classifier for the LDA model

Within the linear discriminant model we have,

$$\mathbb{P}(Y = y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

$$\mathbb{P}(X = x | Y = y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu_y)\Sigma^{-1}(x - \mu_y)^\top \right).$$

Now $\phi^*(x) = 1$

$$\Longleftrightarrow \quad \mathbb{P}(X = x|Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x|Y = 0) \cdot \mathbb{P}(Y = 0)$$

The optimal classifier for the LDA model

Within the linear discriminant model we have,

$$\mathbb{P}(Y = y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

$$\mathbb{P}(X = x | Y = y) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu_y)\Sigma^{-1}(x - \mu_y)^\top \right).$$

Now $\phi^*(x) = 1$

$$\Longleftrightarrow \quad \mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0)$$

$$\Longleftrightarrow \quad \exp\left(-\frac{1}{2}(x - \mu_1)\Sigma^{-1}(x - \mu_1)^\top \right) \cdot q \geq \exp\left(-\frac{1}{2}(x - \mu_0)\Sigma^{-1}(x - \mu_0)^\top \right) \cdot (1 - q)$$

The optimal classifier for the LDA model

Within the linear discriminant model we have,

$$\mathbb{P}(Y = y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases}$$

$$\mathbb{P}(X = x|Y = y) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu_y)\Sigma^{-1}(x - \mu_y)^\top\right).$$

Now $\phi^*(x) = 1$

$$\Longleftrightarrow \quad \mathbb{P}(X = x|Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x|Y = 0) \cdot \mathbb{P}(Y = 0)$$

$$\Longleftrightarrow \quad \exp\left(-\frac{1}{2}(x - \mu_1)\Sigma^{-1}(x - \mu_1)^\top\right) \cdot q \geq \exp\left(-\frac{1}{2}(x - \mu_0)\Sigma^{-1}(x - \mu_0)^\top\right) \cdot (1 - q)$$

$$\Longleftrightarrow \quad -\frac{1}{2}(x - \mu_1)\Sigma^{-1}(x - \mu_1)^\top + \log q \geq -\frac{1}{2}(x - \mu_0)\Sigma^{-1}(x - \mu_0)^\top + \log(1 - q)$$

The optimal classifier for the LDA model

Hence, we have $\phi^*(x) = 1$

$$\Leftrightarrow \quad -\frac{1}{2}(x-\mu_1)\Sigma^{-1}(x-\mu_1)^\top + \log q \geq -\frac{1}{2}(x-\mu_0)\Sigma^{-1}(x-\mu_0)^\top + \log(1-q)$$

$$\Leftrightarrow \quad \left\{(\mu_1-\mu_0)\Sigma^{-1}\right\} x^\top + \left\{\log\left(\frac{q}{1-q}\right) + \frac{1}{2}\left(\mu_0\Sigma^{-1}\mu_0^\top - \mu_1\Sigma^{-1}\mu_1^\top\right)\right\} \geq 0.$$

Hence, we have $\phi^*(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$

With $w = (\mu_1 - \mu_0)\Sigma^{-1}$

$$w^0 = \log\left(\frac{q}{1-q}\right) + \frac{1}{2}\left(\mu_0\Sigma^{-1}\mu_0^\top - \mu_1\Sigma^{-1}\mu_1^\top\right)$$

The optimal classifier for the LDA model

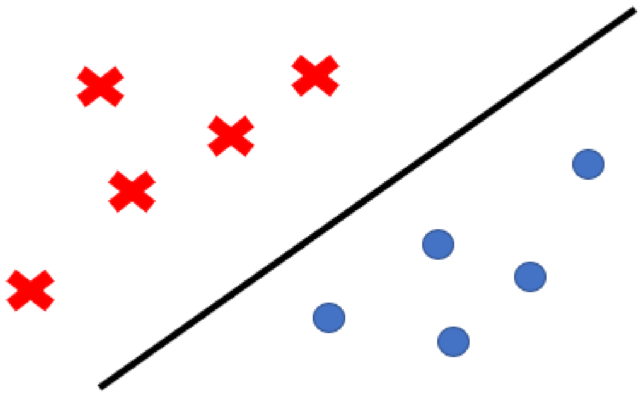
Hence, we have $\phi^*(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$

With $$w = (\mu_1 - \mu_0)\Sigma^{-1}$$

$$w^0 = \log\left(\frac{q}{1-q}\right) + \frac{1}{2}\left(\mu_0 \Sigma^{-1} \mu_0^\top - \mu_1 \Sigma^{-1} \mu_1^\top\right)$$

The Bayes classifier in the linear discriminant model is linear!

However, we don’t know the parameters q, μ_0, μ_1, Σ



Now take a break!

Parameter estimation for LDA

How can we learn the parameters q, μ_0, μ_1, Σ for the linear discriminant analysis model?

We can use the maximum likelihood principle!

Recall that $$\mathbb{P}(Y = y) = q^y (1-q)^{1-y} = \begin{cases} q & y = 1 \\ 1-q & y = 0. \end{cases}$$

For simplicity let's look the one-dimensional case.

It follows that $\mathbb{P}(X = x | Y = y)$ has density $\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x - \mu_y\right)^2\right)$

Parameter estimation for LDA

We have $\mathbb{P}(Y=y) = q^y(1-q)^{1-y}$ and

$\mathbb{P}(X=x|Y=y)$ has density $\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_y)^2\right)$.

Let's use this to compute the density for the random variables (X,Y)

$$
\begin{aligned}
f_{q,\mu_0,\mu_1,\sigma}(x,y) &\approx \mathbb{P}(X=x, Y=y) \\
&\approx \mathbb{P}(Y=y)\cdot\mathbb{P}(X=x|Y=y) \\
&\approx q^y(1-q)^{1-y}\cdot\left\{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_y)^2\right)\right\}.
\end{aligned}
$$

Parameter estimation for LDA

With $$f_{q,\mu_0,\mu_1,\sigma}(x,y) \approx q^y(1-q)^y \cdot \left\{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(x-\mu_y\right)^2\right)\right\}.$$

We can compute the likelihood of the data $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$,

$$\begin{aligned}\ell(q,\mu_0,\mu_1,\sigma) &= \prod_{i=1}^{n} f_{q,\mu_0,\mu_1,\sigma}(X_i,Y_i)\\ &= \prod_{i=1}^{n}\left\{q^{Y_i}(1-q)^{1-Y_i}\cdot\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(X_i-\mu_{Y_i}\right)^2\right)\right.\end{aligned}$$

Let's try to maximise the likelihood to estimate the parameters q, μ_0, μ_1, σ

Parameter estimation for LDA

The likelihood is $\ell(q,\mu_0,\mu_1,\sigma) = \prod_{i=1}^{n} \left\{ q^{Y_i}(1-q)^{1-Y_i} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(X_i - \mu_{Y_i}\right)^2 \right) \right\}$

To maximise the likelihood we will maximise the log-likelihood,

$$
\begin{aligned}
&\log \ell(q,\mu_0,\mu_1,\sigma) \\
&= \sum_{i=1}^{n} \left\{ Y_i \log(q) + (1-Y_i)\log(1-q) - \frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(X_i - \mu_{Y_i}\right)^2 \right\} \\
&= \left(\sum_{i=1}^{n} Y_i \right) \log\left(\frac{q}{1-q} \right) + n\log(1-q) - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left\{ \sum_{i:Y_i=0} \left(X_i-\mu_0\right)^2 + \sum_{i:Y_i=1} \left(X_i-\mu_1\right)^2 \right\} \\
&= n_1 \log\left(\frac{q}{1-q} \right) + n\log(1-q) - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left\{ \sum_{i:Y_i=0} \left(X_i-\mu_0\right)^2 + \sum_{i:Y_i=1} \left(X_i-\mu_1\right)^2 \right\}.
\end{aligned}
$$

Where $n_1 = |\{i : Y_i = 1\}| = \sum_{i=1}^{n} Y_i$ and $n_0 = |\{i : Y_i = 0\}| = n - n_1$

Parameter estimation for LDA

The log-likelihood is $\log \ell(q, \mu_0, \mu_1, \sigma)$

$$= n_1 \log\left(\frac{q}{1-q}\right) + n\log(1-q) - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left\{\sum_{i:Y_i=0}(X_i-\mu_0)^2 + \sum_{i:Y_i=1}(X_i-\mu_1)^2\right\}.$$

To maximise the log-likelihood we find the point where the derivatives are equal to zero.

1. We can compute the derivative

$$\frac{\partial}{\partial q}\log \ell(q, \mu_0, \mu_1, \sigma) = n_1\left(\frac{1}{q} + \frac{1}{1-q}\right) - \frac{n}{1-q}.$$

Taking $\frac{\partial}{\partial q}\log \ell(q, \mu_0, \mu_1, \sigma) = 0$ gives an MLE of $\hat{q} = \frac{n_1}{n} = \frac{1}{n}\sum_{i=1} Y_i$

Parameter estimation for LDA

The log-likelihood is $\log \ell(q, \mu_0, \mu_1, \sigma)$

$$= n_1 \log\left(\frac{q}{1-q}\right) + n\log(1-q) - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left\{\sum_{i:Y_i=0}(X_i-\mu_0)^2 + \sum_{i:Y_i=1}(X_i-\mu_1)^2\right\}.$$

To maximise the log-likelihood we find the point where the derivatives are equal to zero.

2. For $y \in \{0,1\}$ we compute the derivative

$$\frac{\partial}{\partial \mu_y}\log \ell(q, \mu_0, \mu_1, \sigma) = \frac{1}{\sigma^2}\sum_{i:Y_i=y}(X_i - \mu_y)$$

Setting the derivative to zero gives a maximum likelihood estimate of $\hat{\mu}_y = \frac{1}{n_y}\sum_{i:Y_i=y} X_i$

Parameter estimation for LDA

The log-likelihood is $\log \ell(q, \mu_0, \mu_1, \sigma)$

$$= n_1 \log\left(\frac{q}{1-q}\right) + n\log(1-q) - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left\{\sum_{i:Y_i=0}(X_i-\mu_0)^2 + \sum_{i:Y_i=1}(X_i-\mu_1)^2\right\}.$$

To maximise the log-likelihood we find the point where the derivatives are equal to zero.

2. We compute the derivative

$$\frac{\partial}{\partial\sigma}\log \ell(q, \mu_0, \mu_1, \sigma) = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\left\{\sum_{i:Y_i=0}(X_i-\mu_0)^2 + \sum_{i:Y_i=1}(X_i-\mu_1)^2\right\}.$$

The MLE is $$\hat{\sigma}^2 = \frac{1}{n}\left\{\sum_{i:Y_i=0}(X_i-\hat{\mu}_0)^2 + \sum_{i:Y_i=1}(X_i-\hat{\mu}_1)^2\right\}.$$

The optimal classifier for the LDA model

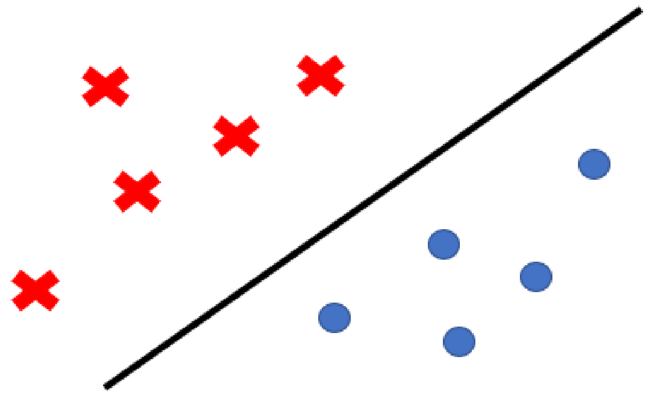
Hence, we have $\phi^*(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$

With $w = (\mu_1 - \mu_0)\Sigma^{-1}$

$$w^0 = \log\left(\frac{q}{1-q}\right) + \frac{1}{2}\left(\mu_0\Sigma^{-1}\mu_0^\top - \mu_1\Sigma^{-1}\mu_1^\top\right)$$

The Bayes classifier in the linear discriminant model is linear!

We estimate q, μ_0, μ_1, Σ by maximum likelihood.



The linear discriminant analysis classifier

The linear discriminant analysis classifier is given by $\hat{\phi}(x) = \mathbb{1}\left\{\hat{w}\ x^\top + \hat{w}_0 \geq 0\right\}$

Where $\hat{q} = \frac{n_1}{n} = \frac{1}{n}\sum_{i=1} Y_i \qquad \hat{\mu}_y = \frac{1}{n_y}\sum_{i:Y_i=y} X_i$

$$\hat{\Sigma} = \frac{1}{n}\left\{\sum_{i:Y_i=0}(X_i - \hat{\mu}_0)^\top (X_i - \hat{\mu}_0) + \sum_{i:Y_i=0}(X_i - \hat{\mu}_1)^\top (X_i - \hat{\mu}_1)\right\}.$$

$$\hat{w} = (\hat{\mu}_1 - \hat{\mu}_0)\hat{\Sigma}^{-1} \qquad \hat{w}^0 = \log\left(\frac{\hat{q}}{1-\hat{q}}\right) + \frac{1}{2}\left(\hat{\mu}_0\Sigma^{-1}\hat{\mu}_0^\top - \hat{\mu}_1\hat{\Sigma}^{-1}\hat{\mu}_1^\top\right)$$

This is a linear classifier fitted with maximum likelihood estmation!

The linear discriminant analysis classifier

The linear discriminant analysis classifier is given by $\hat{\phi}(x) = \mathbb{1}\left\{\hat{w}\, x^\top + \hat{w}_0 \geq 0\right\}$

Where $\hat{q} = \frac{n_1}{n} = \frac{1}{n}\sum_{i=1} Y_i \qquad \hat{\mu}_y = \frac{1}{n_y}\sum_{i:Y_i=y} X_i$

$$\hat{\Sigma}_{\mathrm{U}} = \frac{1}{n-2}\left\{\sum_{i:Y_i=0}(X_i-\hat{\mu}_0)^\top (X_i-\hat{\mu}_0) + \sum_{i:Y_i=0}(X_i-\hat{\mu}_1)^\top (X_i-\hat{\mu}_1)\right\}.$$

$$\hat{w} = (\hat{\mu}_1 - \hat{\mu}_0)\hat{\Sigma}^{-1} \qquad \hat{w}^0 = \log\left(\frac{\hat{q}}{1-\hat{q}}\right) + \frac{1}{2}\left(\hat{\mu}_0\Sigma^{-1}\hat{\mu}_0^\top - \hat{\mu}_1\hat{\Sigma}^{-1}\hat{\mu}_1^\top\right)$$

This is a linear classifier fitted with maximum likelihood estmation!

The linear discriminant analysis classifier in R

Suppose we want to learn a classifier $\phi : \mathcal{X} \rightarrow \mathcal{Y}$ which takes a feature vector of morphological features and predicts whether a penguin belongs to either the Adelie species or the Gentoo species.

```
library(tidyverse)
library(palmerpenguins)

peng_total<-penguins%>% # prepare our data
  select(body_mass_g,flipper_length_mm,species)%>%
  filter(species!="Chinstrap")%>%
  drop_na()%>%
  mutate(species=as.numeric(species=="Adelie"))
```

The linear discriminant analysis classifier in R

Suppose we want to learn a classifier $\phi : \mathcal{X} \rightarrow \mathcal{Y}$ which takes a feature vector of morphological features and predicts whether a penguin belongs to either the Adelie species or the Gentoo species.

```r
library(tidyverse)
library(palmerpenguins)

peng_total<-penguins%>% # prepare our data
  select(body_mass_g,flipper_length_mm,species)%>%
  filter(species!="Chinstrap")%>%
  drop_na()%>%
  mutate(species=as.numeric(species=="Adelie"))
```

```r
peng_total
```

```
## # A tibble: 219 x 3
##    body_mass_g flipper_length_mm species
##          <int>             <int>   <dbl>
##  1        3750               181       1
##  2        3800               186       1
##  3        3250               195       1
##  4        3450               193       1
##  5        3650               190       1
##  6        3625               181       1
##  7        4675               195       1
##  8        3475               193       1
##  9        4250               190       1
## 10        3300               186       1
## # ... with 209 more rows
```

The linear discriminant analysis classifier in R

Suppose we want to learn a classifier $\phi : \mathcal{X} \to \mathcal{Y}$ which takes a feature vector of morphological features and predicts whether a penguin belongs to either the Adelie species or the Gentoo species

$$\overbrace{}^{X} \quad \overbrace{}^{Y}$$

```
## # A tibble: 219 x 3
##    body_mass_g flipper_length_mm species
##          <int>             <int>   <dbl>
## 1         3750               181       1
## 2         3800               186       1
## 3         3250               195       1
## 4         3450               193       1
## 5         3650               190       1
```

Feature vector $X = (X^1, X^2) \in \mathcal{X} = \mathbb{R}^2$

$X^1 =$ the weight of the penguin (grams).

$X^2 =$ the flipper length of the penguin (mm).

Label $Y \in \mathcal{Y} = \{0, 1\}$

$$Y = \begin{cases} 1 \text{ if the penguin is an Adelie} \\ 0 \text{ if the penguin is a Gentoo.} \end{cases}$$

The linear discriminant analysis classifier in R

Now let’s carry out a train test split.

```
num_total<-peng_total%>%nrow() # number of penguin data
num_train<-floor(num_total*0.75) # number of train examples
num_test<-num_total-num_train # number of test samples

set.seed(1) # set random seed for reproducibility
test_inds<-sample(seq(num_total),num_test) # random sample of test indicies
train_inds<-setdiff(seq(num_total),test_inds) # training data indicies

peng_train<-peng_total%>%filter(row_number() %in% train_inds) # train data
peng_test<-peng_total%>%filter(row_number() %in% test_inds) # test data
```

Remember to set a random seed for reproducibility.

The linear discriminant analysis classifier in R

Now let's carry out a train test split.

```
num_total<-peng_total%>%nrow() # number of penguin data
num_train<-floor(num_total*0.75) # number of train examples
num_test<-num_total-num_train # number of test samples

set.seed(1) # set random seed for reproducibility
test_inds<-sample(seq(num_total),num_test) # random sample of test indicies
train_inds<-setdiff(seq(num_total),test_inds) # training data indicies

peng_train<-peng_total%>%filter(row_number() %in% train_inds) # train data
peng_test<-peng_total%>%filter(row_number() %in% test_inds) # test data
```

```
peng_train_x<-peng_train%>%select(-species) # train feature vectors
peng_train_y<-peng_train%>%pull(species) # train labels

peng_test_x<-peng_test%>%select(-species) # test feature vectors
peng_test_y<-peng_test%>%pull(species) # test labels
```

The linear discriminant analysis classifier in R

We can now build a linear discriminant analysis model with R as follows:

```
lda_model <- MASS::lda(species ~ ., data=peng_train) # fit LDA model
```

The linear discriminant analysis classifier in R

We can now build a linear discriminant analysis model with R as follows:

```
lda_model <- MASS::lda(species ~ ., data=peng_train) # fit LDA model
```

We can make predictions and check the training error as follows:

```
lda_train_predicted_y<-predict(lda_model,peng_train_x)$class%>%
  as.character()%>%as.numeric() # get vector of predicted ys

lda_train_error<-mean(abs(lda_train_predicted_y-peng_train_y)) # compute train error

lda_train_error
```

```
## [1] 0.01463415
```

The linear discriminant analysis classifier in R

We can now build a linear discriminant analysis model with R as follows:

```
lda_model <- MASS::lda(species ~ ., data=peng_train) # fit LDA model
```

We can make test predictions and estimate our test error as follows:

```
lda_test_predicted_y<-predict(lda_model,peng_test_x)$class%>%
  as.character()%>%as.numeric() # get vector of predicted ys

lda_test_error<-mean(abs(lda_test_predicted_y-peng_test_y)) # compute test error

lda_test_error
```

```
## [1] 0.01449275
```

Now take a break!

Logistic regression

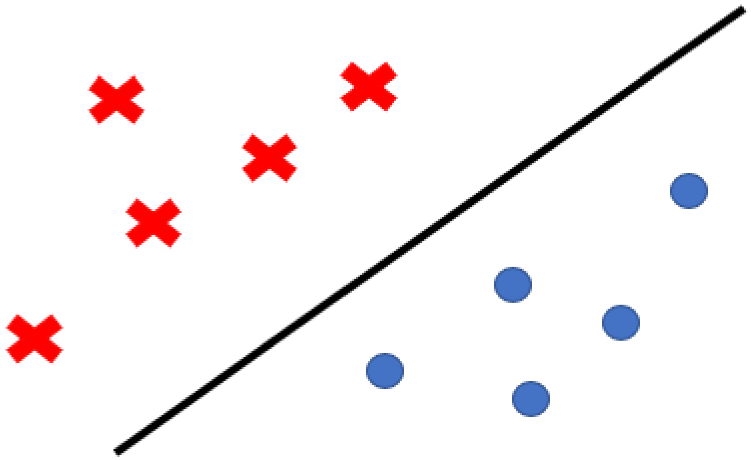
The first thing to remember about logistic regression is that its <u>not a regression algorithm</u>!

Logistic regression is a probabilistic method for learning a linear classifier

$$\phi(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$$

Linear discriminant analysis modelled the whole

distribution $\mathbb{P}(X = x, Y = y)$



Logistic regression will directly model

$$\mathbb{P}(Y = y | X = x)$$

Logistic regression

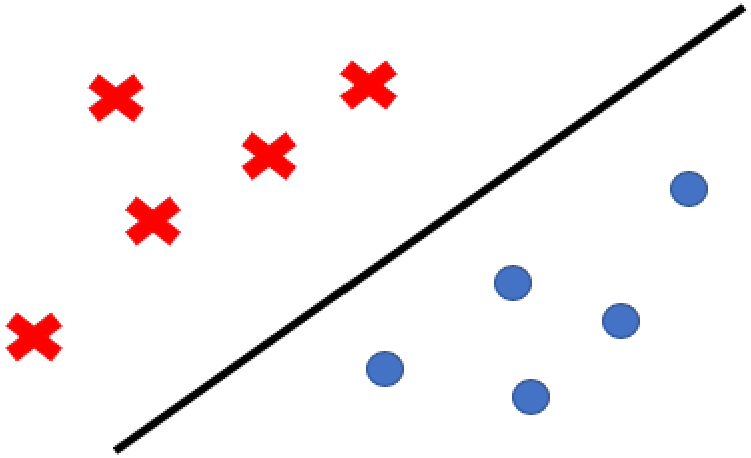
Logistic regression is a method for learning a linear classifier $\phi(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$

Idea 1: The Bayes classifier is given by

$$\phi^*(x) := \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y=1|X=x) \geq \mathbb{P}(Y=0|X=x) \\ 0 & \text{if} \quad \mathbb{P}(Y=0|X=x) > \mathbb{P}(Y=1|X=x). \end{cases}$$

Hence, we only need to model $\mathbb{P}(Y=y|X=x)$

This is easier than modelling $\mathbb{P}(X=x, Y=y)$



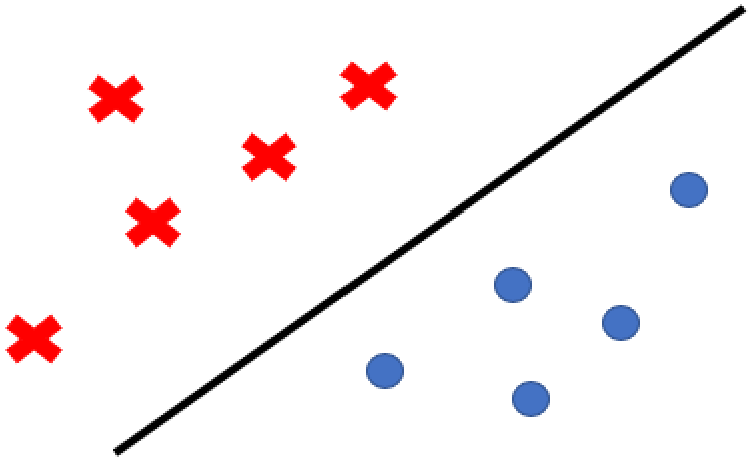
Logistic regression

Logistic regression is a method for learning a linear classifier $\phi(x) = \mathbb{1}\left\{w\, x^\top + w_0 \geq 0\right\}$

Idea 1: We only need to model $\mathbb{P}(Y = y|X = x)$

We could try

$\mathbb{P}(Y = y|X = x) = w\, x^\top + w^0$?



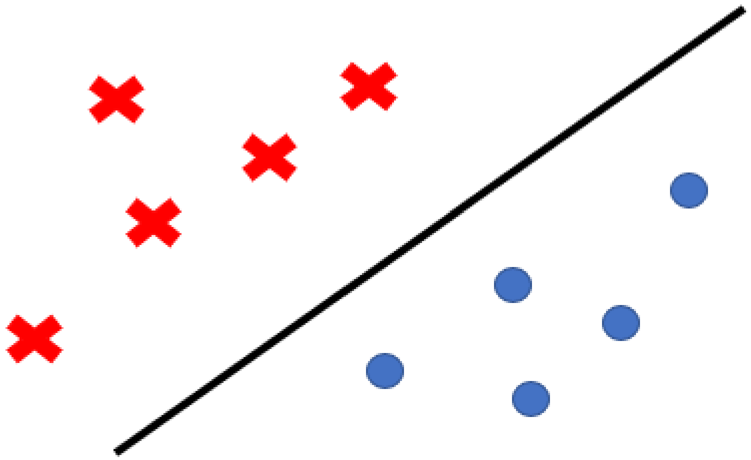
Logistic regression

Logistic regression is a method for learning a linear classifier $\phi(x) = \mathbb{1}\left\{w\,x^\top + w_0 \geq 0\right\}$

Idea 1: We only need to model $\mathbb{P}(Y = y|X = x)$

~~We could try~~

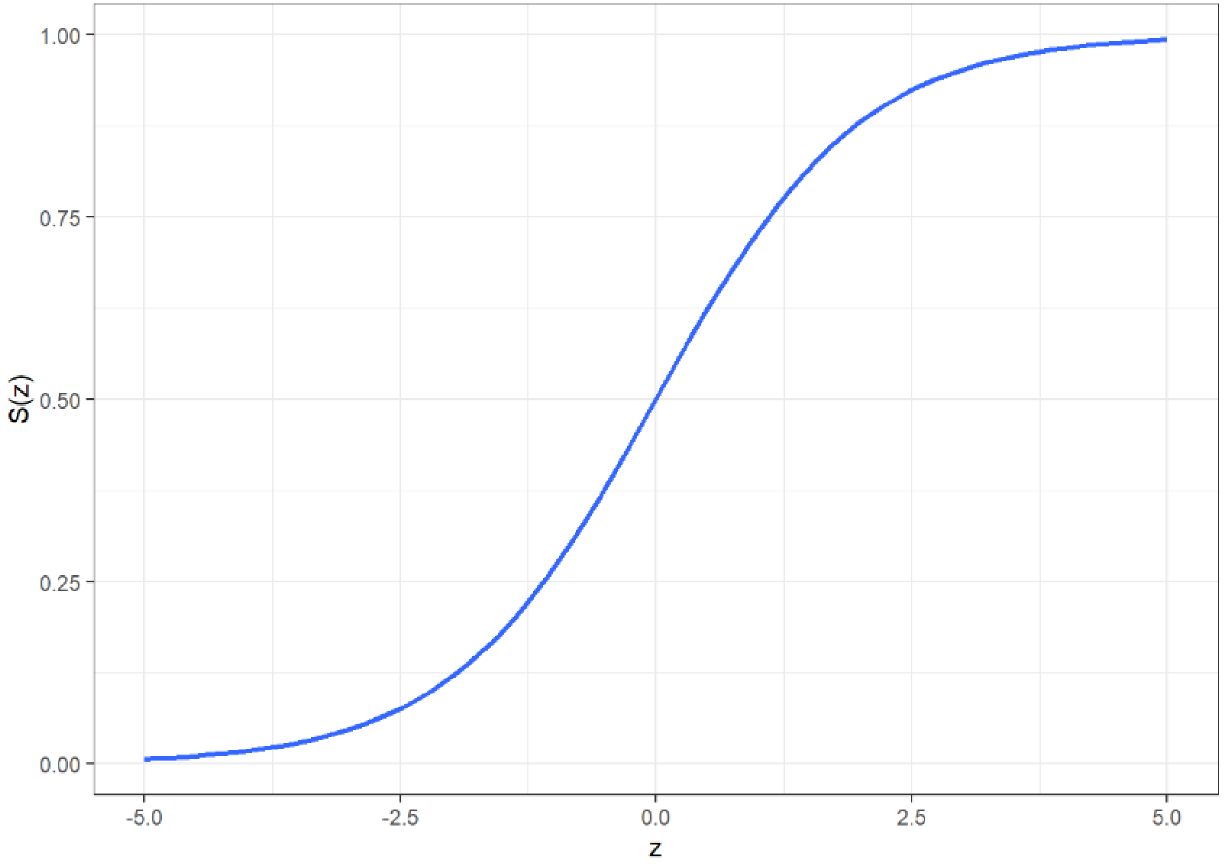
$\cancel{\mathbb{P}(Y = y|X = x) = w\,x^\top + w^0}$?



This is a **bad idea** as it would lead to probabilities outside of $[0, 1]$.

Logistic regression

Idea 2: Use the sigmoid function $S:\mathbb{R}\rightarrow(0,1)$ to map real numbers to probabilities.



$$S(z)=\frac{1}{1+e^{-z}}$$

Logistic regression

Idea 2: Use the sigmoid function $\mathrm{S} : \mathbb{R} \to (0,1)$ to map real numbers to probabilities.

The sigmoid function $$\mathrm{S}(z) = \frac{1}{1+e^{-z}}$$ is also known as the logistic function.

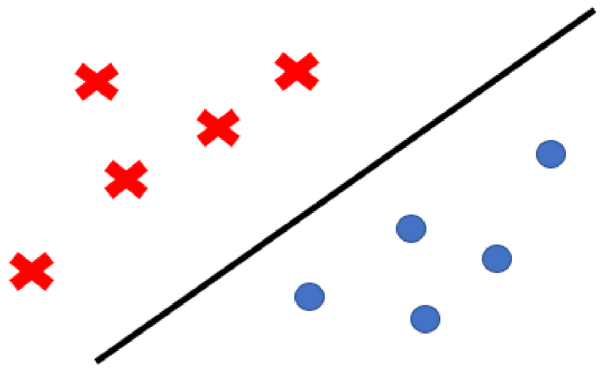
Facts

1. $$1 - \mathrm{S}(z) = \mathrm{S}(-z)$$

2. $$\frac{\partial \log \mathrm{S}(z)}{\partial z} = \mathrm{S}(-z)$$

Logistic regression

Logistic regression is a method for learning a linear classifier



Idea 1: We only need to model $\mathbb{P}(Y = y|X = x)$

Idea 2: Use the sigmoid function $\mathrm{S}(z) = \dfrac{1}{1 + e^{-z}}$

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^{\top} + w^{0}\right) = \frac{1}{1 + e^{-w\, x^{\top} - w^{0}}}.$$

The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^\top + w^0\right) = \frac{1}{1 + e^{-w\, x^\top - w^0}}.$$

The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^\top + w^0\right) = \frac{1}{1 + e^{-w\, x^\top - w^0}}.$$

Recall that the Bayes classifier satisfies $\phi^*(x) = 1$

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\,x^\top + w^0\right) = \frac{1}{1 + e^{-w\,x^\top - w^0}}.$$

Recall that the Bayes classifier satisfies $\phi^*(x) = 1$

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

$$\Longleftrightarrow \quad \mathbb{P}(Y = 1|X = x) \geq 1/2$$

The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^\top + w^0\right) = \frac{1}{1 + e^{-w\, x^\top - w^0}}.$$

Recall that the Bayes classifier satisfies $\phi^*(x) = 1$

$$\Longleftrightarrow \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

$$\Longleftrightarrow \mathbb{P}(Y = 1|X = x) \geq 1/2$$

$$\Longleftrightarrow \mathrm{S}\left(w\, x^\top + w^0\right) \geq 1/2$$

The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\,x^\top + w^0\right) = \frac{1}{1 + e^{-w\,x^\top - w^0}}.$$

Recall that the Bayes classifier satisfies $\phi^*(x) = 1$

$$\iff \mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$$

$$\iff \mathbb{P}(Y = 1|X = x) \geq 1/2$$

$$\iff \mathrm{S}\left(w\,x^\top + w^0\right) \geq 1/2$$

$$\iff w\,x^\top + w^0 \geq 0$$

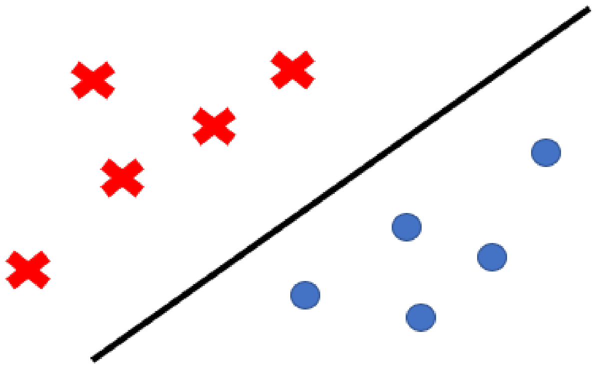
The Bayes classifier for the logistic model

We use the logistic sigmoid model

$$\mathbb{P}(Y = 1 | X = x) = \mathrm{S}\left(w\, x^{\top} + w^{0}\right) = \frac{1}{1 + e^{-w\, x^{\top} - w^{0}}}.$$

The Bayes classifier satisfies $\phi^*(x) = 1 \Longleftrightarrow w\, x^{\top} + w^{0} \geq 0$

Equivalently, we have $\phi^*(x) = \mathbb{1}\left\{w\, x^{\top} + w_0 \geq 0\right\}$



Logistic regression also has a linear Bayes optimal classifier.

Learning parameters for a logistic model

The logistic sigmoid model is given by

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^\top + w^0\right) = \frac{1}{1 + e^{-w\, x^\top - w^0}}.$$

Suppose we have training data $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$.

We can learn the parameters of the model $w,\ w^0$ with an approximate maximum likelihood.

Learning parameters for a logistic model

The logistic sigmoid model is given by

$$\mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(w\, x^\top + w^0\right) = \frac{1}{1 + e^{-w\, x^\top - w^0}}.$$

Suppose we have training data $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$.

We can learn the parameters of the model $w,\ w^0$ with an approximate maximum likelihood.

By fact 1 the sigmoid $\mathrm{S}(z) = \dfrac{1}{1 + e^{-z}}$ satisfies $1 - \mathrm{S}(z) = \mathrm{S}(-z)$

This implies $\mathbb{P}(Y = 0|X = x) = 1 - \mathbb{P}(Y = 1|X = x) = \mathrm{S}\left(-w\, x^\top - w^0\right).$

Learning parameters for a logistic model

The logistic sigmoid model satisfies

$$\mathbb{P}(Y = y|X = x) = \begin{cases} \mathrm{S}\left(w\, x^\top + w^0\right) & \text{if } y = 1 \\ \mathrm{S}\left(-w\, x^\top - w^0\right) & \text{if } y = 0. \end{cases}$$

Equivalently, $\mathbb{P}(Y = y|X = x) = \mathrm{S}\left((2y - 1) \cdot (w\, x^\top + w^0)\right)$

Suppose we have training data $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$.

The likelihood $\ell(w, w^0) = \prod_{i=1}^{n} \mathrm{S}\left((2Y_i - 1) \cdot (w\, X_i^\top + w^0)\right)$

Learning parameters for a logistic model

Suppose we have training data $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$

The likelihood $\ell(w, w^0) = \prod_{i=1}^{n} \mathrm{S}\left((2Y_i - 1) \cdot (w\ X_i^\top + w^0)\right)$

We maximise the log-likelihood .

$$\log \ell(w, w^0) = \sum_{i=1}^{n} \log \mathrm{S}\left((2Y_i - 1) \cdot (w\ X_i^\top + w^0)\right)$$

To maximise the log-likelihood we first compute the gradient.

Learning parameters for a logistic model

Given $\mathcal{D} = ((X_1, Y_1), \cdots, (X_n, Y_n))$ we aim to maximise the log-likelihood

$$\log \ell(w, w^0) = \sum_{i=1}^{n} \log \mathrm{S}\left((2Y_i - 1) \cdot (w\ X_i^\top + w^0)\right)$$

The derivatives are given by

$$\frac{\partial}{\partial w} \log \ell(w, w^0) = \sum_{i=1}^{n} (2Y_i - 1) \mathrm{S}\left((1 - 2Y_i) \cdot (w\ X_i^\top + w^0)\right) \cdot X_i$$

$$\frac{\partial}{\partial w^0} \log \ell(w, w^0) = \sum_{i=1}^{n} (2Y_i - 1) \mathrm{S}\left((1 - 2Y_i) \cdot (w\ X_i^\top + w^0)\right)$$

Learning parameters for a logistic model

We want to maximise $$\log \ell(w, w^0) = \sum_{i=1}^{n} \log \mathrm{S}\left((2Y_i - 1) \cdot (w\ X_i^\top + w^0)\right)$$

The derivatives are given by

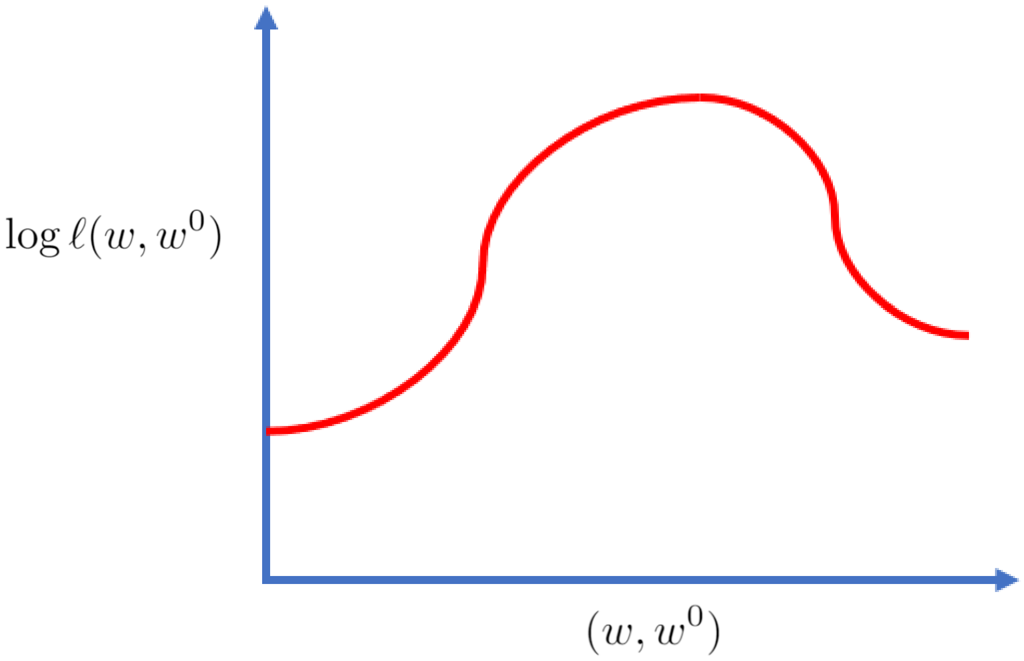
$$\frac{\partial}{\partial w} \log \ell(w, w^0) = \sum_{i=1}^{n} (2Y_i - 1) \mathrm{S}\left((1 - 2Y_i) \cdot (w\ X_i^\top + w^0)\right) \cdot X_i$$
$$\frac{\partial}{\partial w^0} \log \ell(w, w^0) = \sum_{i=1}^{n} (2Y_i - 1) \mathrm{S}\left((1 - 2Y_i) \cdot (w\ X_i^\top + w^0)\right)$$

For linear discriminant analysis we set the derivative to zero and solved analytically.

Unfortunately for logistic regression there is no analytic solution.

Learning parameters for a logistic model

We maximise the log-likelihood $\log \ell(w, w^0)$ iteratively.



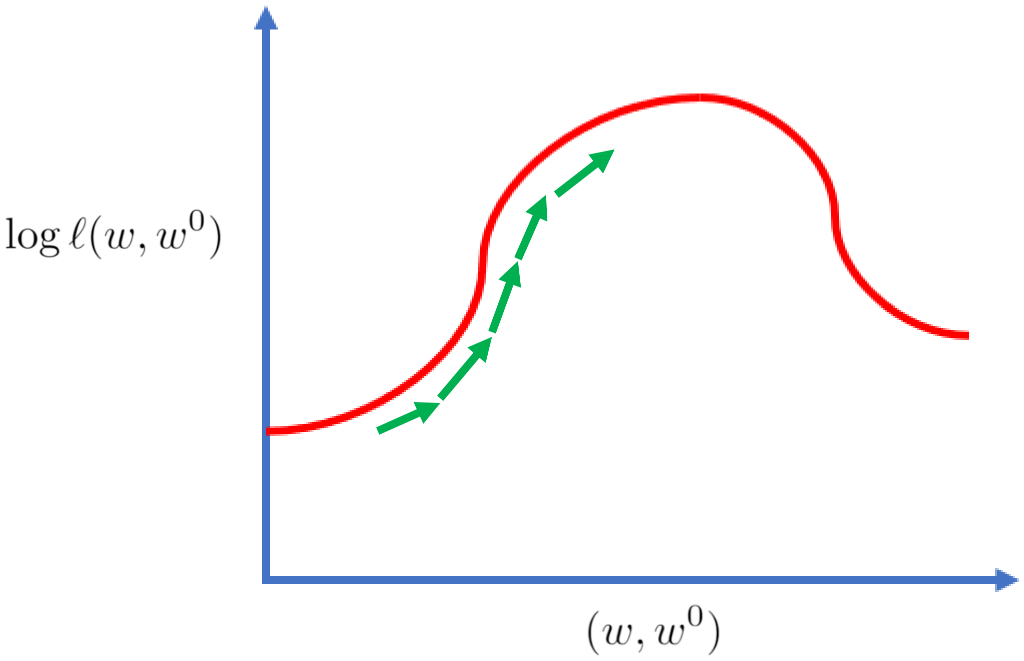
We can use a process of gradient ascent:

$$w \mapsto w + \alpha \cdot \frac{\partial}{\partial w} \log \ell(w, w^0)$$

$$w^0 \mapsto w^0 + \alpha \cdot \frac{\partial}{\partial w^0} \log \ell(w, w^0)$$

Learning parameters for a logistic model

We maximise the log-likelihood $\log \ell(w, w^0)$ iteratively.



We can use a process of gradient ascent:

$$w \mapsto w + \alpha \cdot \frac{\partial}{\partial w} \log \ell(w, w^0)$$

$$w^0 \mapsto w^0 + \alpha \cdot \frac{\partial}{\partial w^0} \log \ell(w, w^0)$$

Logistic regression with R

Suppose we want to learn a classifier $\phi : \mathcal{X} \to \mathcal{Y}$ which takes a feature vector of morphological features and predicts whether a penguin belongs to either the Adelie species or the Gentoo species

$$\overbrace{}^{X}\ \overbrace{}^{Y}$$

```
## # A tibble: 219 x 3
##   body_mass_g flipper_length_mm species
##         <int>             <int>   <dbl>
## 1        3750               181       1
## 2        3800               186       1
## 3        3250               195       1
## 4        3450               193       1
## 5        3650               190       1
```

Feature vector $X = (X^1, X^2) \in \mathcal{X} = \mathbb{R}^2$

$X^1 =$ the weight of the penguin (grams).

$X^2 =$ the flipper length of the penguin (mm).

Label $Y \in \mathcal{Y} = \{0, 1\}$

$$Y = \begin{cases} 1 \text{ if the penguin is an Adelie} \\ 0 \text{ if the penguin is a Chinstrap.} \end{cases}$$

Logistic regression with R

Now let's carry out a train test split.

```
num_total<-peng_total%>%nrow() # number of penguin data
num_train<-floor(num_total*0.75) # number of train examples
num_test<-num_total-num_train # number of test samples

set.seed(1) # set random seed for reproducibility
test_inds<-sample(seq(num_total),num_test) # random sample of test indicies
train_inds<-setdiff(seq(num_total),test_inds) # training data indicies

peng_train<-peng_total%>%filter(row_number() %in% train_inds) # train data
peng_test<-peng_total%>%filter(row_number() %in% test_inds) # test data
```

```
peng_train_x<-peng_train%>%select(-species) # train feature vectors
peng_train_y<-peng_train%>%pull(species) # train labels

peng_test_x<-peng_test%>%select(-species) # test feature vectors
peng_test_y<-peng_test%>%pull(species) # test labels
```

Logistic regression with R

We can train a logistic model with the glmnet library.

```
library(glmnet) # load the glmnet library
logistic_model<-glmnet(x=peng_train_x%>%as.matrix(),y=peng_train_y,
                       family ="binomial",alpha=0,lambda=0) # train a logistic model
```

Logistic regression with R

We can train a logistic model with the glmnet library.

```
library(glmnet) # load the glmnet library
logistic_model<-glmnet(x=peng_train_x%>%as.matrix(),y=peng_train_y,
                       family ="binomial",alpha=0,lambda=0) # train a logistic model
```

We compute the train error as follows.

```
logistic_train_predicted_y<-predict(logistic_model,peng_train_x%>%
                                      as.matrix(),type="class")%>%as.integer()

logistic_train_error<-mean(abs(logistic_train_predicted_y-peng_train_y)) # train error

logistic_train_error
```

```
## [1] 0.009756098
```

Logistic regression with R

We can train a logistic model with the glmnet library.

```
library(glmnet) # load the glmnet library
logistic_model<-glmnet(x=peng_train_x%>%as.matrix(),y=peng_train_y,
                       family ="binomial",alpha=0,lambda=0) # train a logistic model
```

We compute the test error as follows.

```
logistic_test_predicted_y<-predict(logistic_model,peng_test_x%>%
                                     as.matrix(),type="class")%>%as.integer()

logistic_test_error<-mean(abs(logistic_test_predicted_y-peng_test_y)) # test error

logistic_test_error
```

```
## [1] 0.01449275
```

What have we covered today?

- We introduced the concept of a linear classifier.

- We investigated generative linear discriminant analysis (LDA) model:
    - LDA models the joint probability distribution with Gaussians.
    - The maximum likelihood parameters can be estimated analytically.

- We investigated the discriminatve logistic regression:
    - Logistic regression models the class-conditional model with a sigmoid linear model.
    - The maximum likelihood parameters must be estimated iteratively.

Thanks for listening!

henry.reeve@bristol.ac.uk

Include EMATM0061 in the subject of your email.