

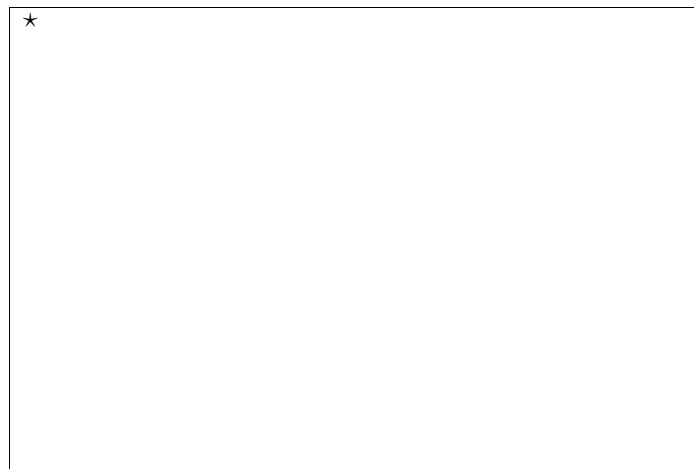
## Lecture 5 - *Linear Supervised Learning*

**OBJECTIVE: Linear regression** is a supervised learning task. It is of great interest because:

- Many real processes can be approximated with linear models.
- Linear regression appears as part of larger problems.
- It can be solved analytically.
- It illustrates many of the approaches to machine learning.

**Textbook:** Pages 41–49.

Given the data  $\{x_{1:n}, y_{1:n}\}$ , with  $x_i \in \mathbb{R}^d$  and  $y_i \in \mathbb{R}$ , we want to fit a hyper-plane that maps  $x$  to  $y$ .



Mathematically, the linear model is expressed as follows:

$$y_i = \theta_0 + \sum_{j=1}^d x_{ij} \theta_j$$

We let  $x_{i,0} = 1$  to obtain

$$y_i = \sum_{j=0}^d x_{ij} \theta_j$$

In matrix form, this expression is

$$Y = X\theta$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{10} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots \\ x_{n0} & \cdots & x_{nd} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_d \end{bmatrix}$$

If we have several outputs  $y_i \in \mathbb{R}^c$ , our linear regression expression becomes:

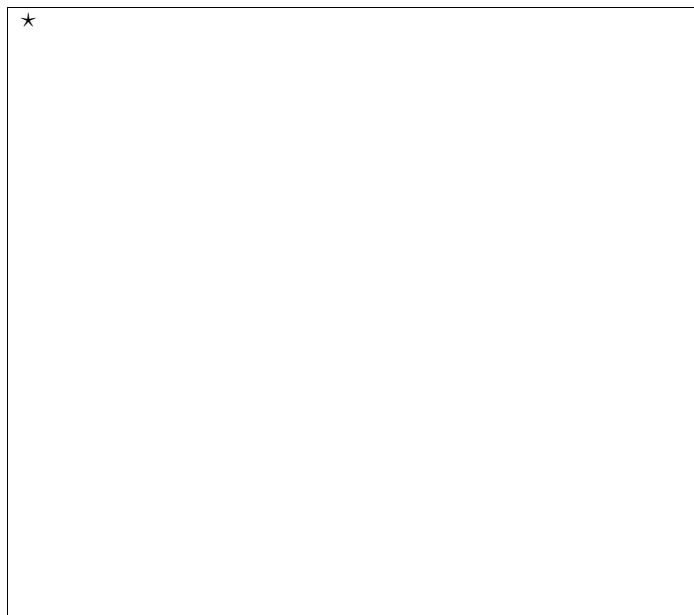


We will present several approaches for computing  $\theta$ .

## ◇ OPTIMIZATION APPROACH

Our aim is to minimise the quadratic cost between the output labels and the model predictions

$$C(\theta) = (Y - X\theta)^T(Y - X\theta)$$



We will need the following result from matrix differentiation:  $\frac{\partial A}{\partial \theta} = A^T$ .

★

$$\frac{\partial C}{\partial \theta} =$$

These are the **normal equations**. The solution (estimate) is:

$$\hat{\theta} =$$

The corresponding predictions are

$$\hat{Y} = HY =$$

where H is the “hat” matrix.

# ◇ GEOMETRIC APPROACH

★

$$X^T(Y - \hat{Y}) =$$

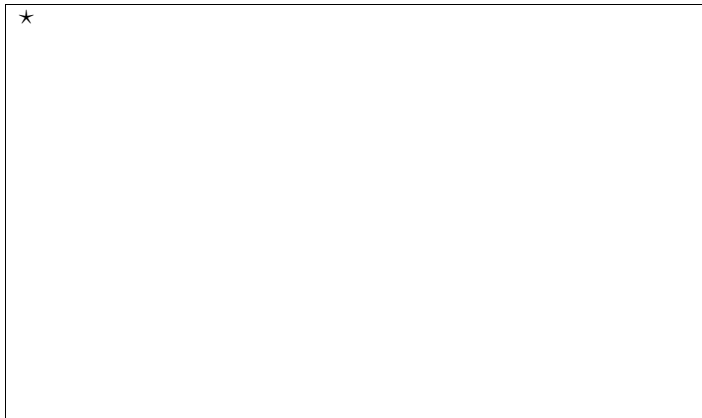
## ◇ PROBABILISTIC APPROACH

**Univariate Gaussian Distribution**

The probability density function of a Gaussian distribution is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$

where  $\mu$  is the mean or center of mass and  $\sigma^2$  is the variance.



Our short notation for Gaussian variables is  $X \sim \mathcal{N}(\mu, \sigma^2)$ .

**Multivariate Gaussian Distribution**

Let  $x \in \mathbb{R}^n$ . The pdf of an n-dimensional Gaussian is given by

$$p(x) = \frac{1}{2\pi^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

where

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} \mathbb{E}(x_1) \\ \vdots \\ \mathbb{E}(x_n) \end{pmatrix}$$

and

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{pmatrix} = \mathbb{E}[(X - \mu)(X - \mu)^T]$$

with  $\sigma_{ij} = \mathbb{E}[X_i - \mu_i)(X_j - \mu_j)^T]$ .

We can interpret each component of  $x$ , for example, as a feature of an image such as colour or texture. The term  $\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)$  is called the **Mahalanobis distance**. Conceptually, it measures the distance between  $x$  and  $\mu$ .

## Maximum Likelihood

If our errors are Gaussian distributed, we can use the model

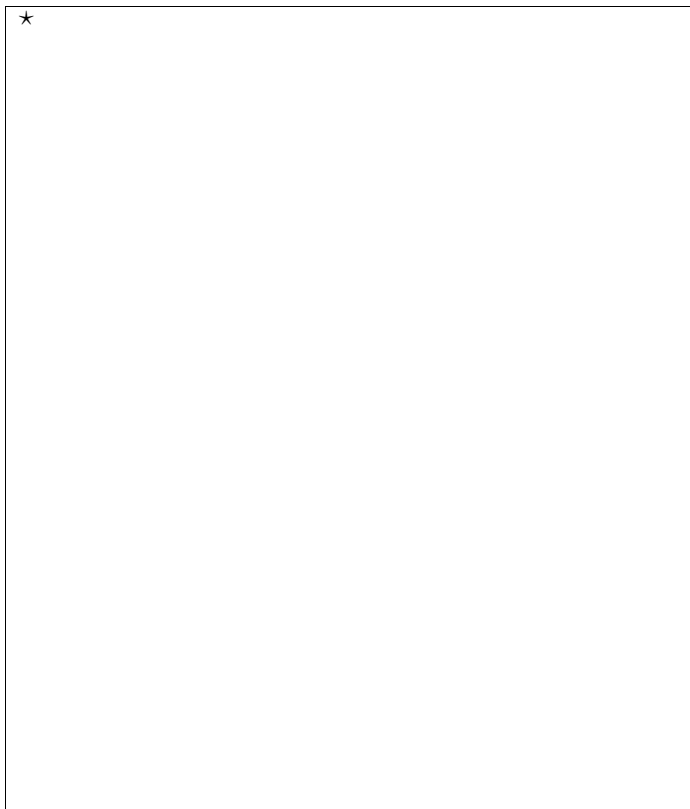
$$Y = X\theta + \mathcal{N}(0, \sigma^2 I)$$

Note that the mean of  $Y$  is  $X\theta$  and that its variance is  $\sigma^2 I$ . So we can equivalently write this expression using the probability density of  $Y$  **given**  $X$ ,  $\theta$  and  $\sigma$ :

$$p(Y|X, \theta, \sigma) = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}(Y-X\theta)^T(Y-X\theta)}$$

The maximum likelihood (ML) estimate of  $\theta$  is obtained by taking the derivative of the log-likelihood,  $\log p(Y|X, \theta, \sigma)$ . The idea of maximum likelihood learning is to maximise the likelihood of seeing some data  $Y$  by modifying the parameters  $(\theta, \sigma)$ .

The ML estimate of  $\theta$  is:



Proceeding in the same way, the ML estimate of  $\sigma$  is:

