COMS21202: Symbols, Patterns and Signals
Review - Part 1

Dima Damen
Dima.Damen@bristol.ac.uk

Bristol University, Department of Computer Science
Bristol BS8 1UB, UK

May 9, 2019

What is Data?

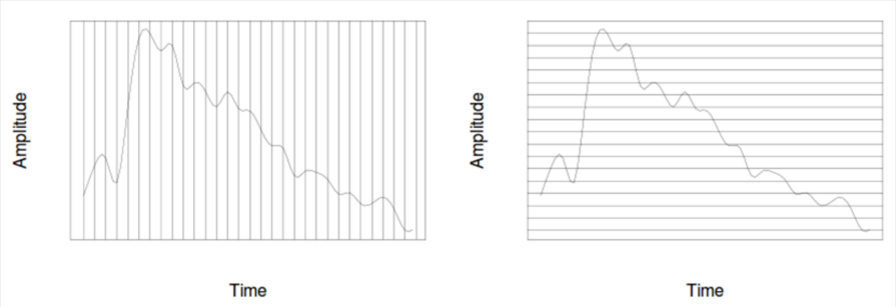
Dima Damen
Dima.Damen@bristol.ac.uk

Data Acquisition - Analogue to Digital Conversion

Analogue to Digital conversion involves

1. Sampling
2. Quantisation

e.g. Audio Signal - 1D



Distance

- Distance is measure of separation between data.
- Can be defined between single-dimensional data, multi-dimensional data or data sequences.
- Distance is important as it:
  - enables data to be ordered
  - allows numeric calculations
  - enables calculating similarity and dissimilarity
- Without defining a distance measure, almost all statistical and machine learning algorithms will not be able to function.

Distance

A valid distance measure $D(a, b)$ between two components a and b has properties

- non-negative: $D(a, b) > 0$
- reflexive: $D(a, b) = 0 \iff a = b$
- symmetric: $D(a, b) = D(b, a)$
- satisfies triangular inequality: $D(a, b) + D(b, c) \geq D(a, c)$

Covariance Matrix

In three dimensions,

$$\Sigma = \frac{1}{N-1} \sum_i \begin{bmatrix} (v_{i1}-\mu_1)^2 & (v_{i1}-\mu_1)(v_{i2}-\mu_2) & (v_{i1}-\mu_1)(v_{i3}-\mu_3) \\ (v_{i1}-\mu_1)(v_{i2}-\mu_2) & (v_{i2}-\mu_2)^2 & (v_{i2}-\mu_2)(v_{i3}-\mu_3) \\ (v_{i1}-\mu_1)(v_{i3}-\mu_3) & (v_{i2}-\mu_2)(v_{i3}-\mu_3) & (v_{i3}-\mu_3)^2 \end{bmatrix}$$

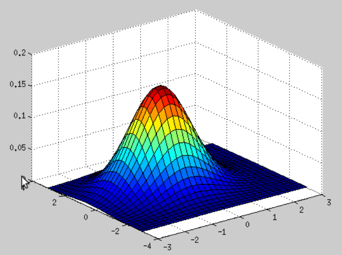
Covariance matrix is always

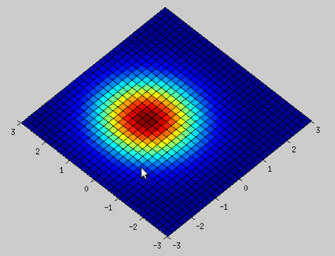
- square and symmetric
- variances on the diagonal
- covariance between each pair of dimensions is included in non-diagonal elements

Normal Distribution - Multi-dimensional

For multi-dimensional normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ in M dimensions, the probability density function (pdf) can be calculated as

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^M |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})} \tag{1}$$

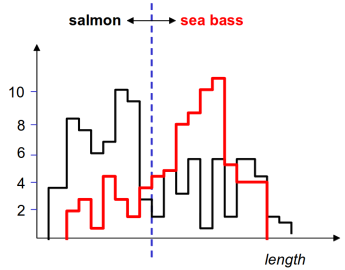




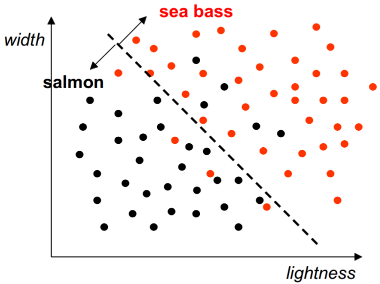
WARNING: **Σ is the capital letter of σ, not the summation sign!**

Model Parameters

- Models are defined in terms of parameters (one or more)
- These may be empirically obtained e.g. by trial and error
- or from training data by tuning or training the model



one parameter needed $x = t$

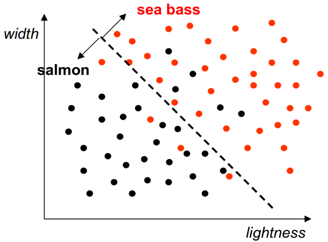


two parameters needed

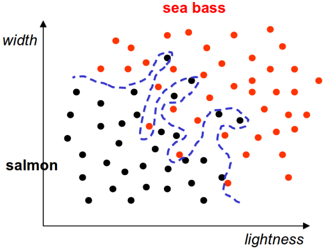
$y = mx + c$

Generalisation vs. Overfitting

- **Simpler models** often give good performance and can be more **general**
- **highly complex models** over-fit the training data



two parameters needed
$y = mx + c$



A large number of parameters needs to be tuned

Another Fish Problem

Data: a set of data points $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ where x_i is the length of fish i and y_i is the weight of fish i.

Task: build a model that can predict the weight of a fish from its length

Model Type: assume there exists a polynomial relationship between length and weight

Model Complexity: assume the relationship is linear
$weight = a + b * length$

$$y_i = a + bx_i \tag{2}$$

Model Parameters: model has two parameters *a* and *b* which should be estimated.

- *a* is the y-intercept
- *b* is the slope of the line

General Least Squares - matrix form

- Matrix formulation also allows least squares method to be extended to **polynomial fitting**
- For a polynomial of degree $p+1$

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_p\ x_i^p$$

General Least Squares - matrix form

- Solved in the same manner

$$\mathbf{y}_{(N\times 1)} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \mathbf{X}_{(N\times(p+1))} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^p \end{bmatrix}, \mathbf{a}_{((p+1)\times 1)} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}$$

$$\mathbf{a}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\,\mathbf{y}$$

where $(\mathbf{X}^T\mathbf{X})$ is a $(p+1)\times(p+1)$ square matrix

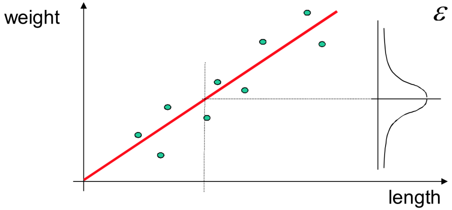
Back to Fish - Continuous

$$\textit{weight} = a \times \textit{length} + \epsilon$$

This is a model with one parameter, apart from the uncertainty

We can assume, for example, that ϵ is $\mathcal{N}(0, \sigma^2)$

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



Dima Damen
Dima.Damen@bristol.ac.uk

Maximum Likelihood Estimation - General

- Maximum Likelihood Estimation (MLE) is a common method for solving such problems

$$\begin{aligned}\theta_{MLE} &= \mathit{arg\ max}_\theta\, p(D|\theta) \\ &= \mathit{arg\ max}_\theta\, \ln p(D|\theta) \\ &= \mathit{arg\ min}_\theta\, -\ln p(D|\theta)\end{aligned}$$

MLE Recipe

1. Determine θ, D and expression for likelihood $p(D|\theta)$
2. Take the natural logarithm of the likelihood
3. Take the derivative of $\ln p(D|\theta)$ w.r.t. θ. If θ is a multi-dimensional vector, take partial derivatives
4. Set derivative(s) to 0 and solve for θ

Probabilistic Model - Ex2

Example

Given a coin, you were assigned the task of figuring out whether the coin will land on its head or tails. You were asked to build a probabilistic model (i.e. with confidence)

- Use binomial distribution for likelihood

$$\theta_{ML} = \frac{D}{N}$$

  where D is the number of success (i.e. heads)
- Use Gaussian distribution for likelihood

$$\theta_{ML} = \frac{1}{N} \sum_{i=1}^{N} d_i$$

  where $d_i = 1$ if success (i.e. heads) or $d_i = 0$ if failure (i.e. tails)
- same answer, different view

Probabilistic Model - Likelihood and Prior

- MLE ignores any prior knowledge we may have about θ
- If we have prior knowledge about values that θ is likely to have, then we can built this into MLE

$$\theta_{ML} = arg\, max_{\theta}\, p(D|\theta)\, p(\theta)$$

- This is known as **Maximum a Posteriori (MAP)** estimation

Note

- Use this lecture for revision NOT for studying!