

# Statistical Learning

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya  
Study Abroad

*jordi.villa@uvic.cat*

course 2023-2024

- 1 Introduction and scope
- 2 Introduction
- 3 Statistical modelling
- 4 Modelling data
- 5 Bibliography
- 6 Detailed notation

# Preliminary note

The material in these slides is strongly based on [?]. When other materials are used, they are cited accordingly.

Mathematical notation follows as good as it can a [good practices proposal](#) from the Beijing Academy of Artificial Intelligence.

# What to expect?

In this session we will discuss:

- Modelling data.
- Models with independent and identically distributed (iid) data.
- The modelling dilemma.
- Clustering as an example of unsupervised learning method.
- Loss function and risk.
- Polynomial regression.

# How is data analyzed and used?

**Statistical learning** interpret the model and quantify the uncertainty of the data.

**Machine learning** (or *data mining*) making predictions using large scale data.

The goals of modelling data are:

- to predict data, based on existing one;
- to discover unusual or interesting patterns in data.

# Types of machine learning

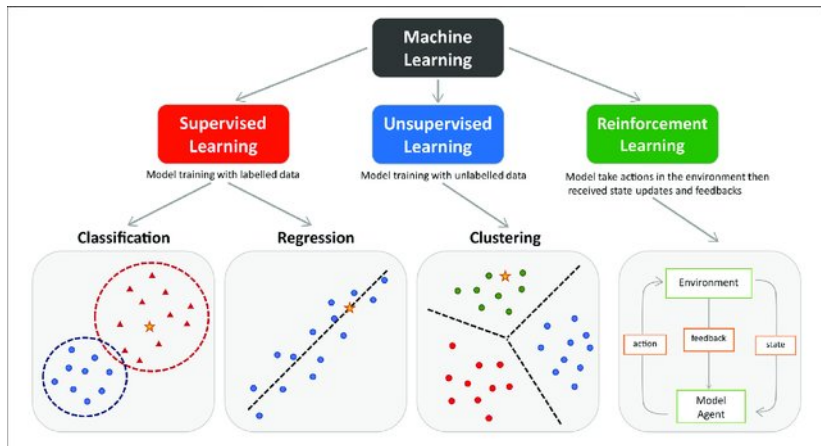


Figure 1: Different types of machine learning techniques[?]

# Supervised vs unsupervised

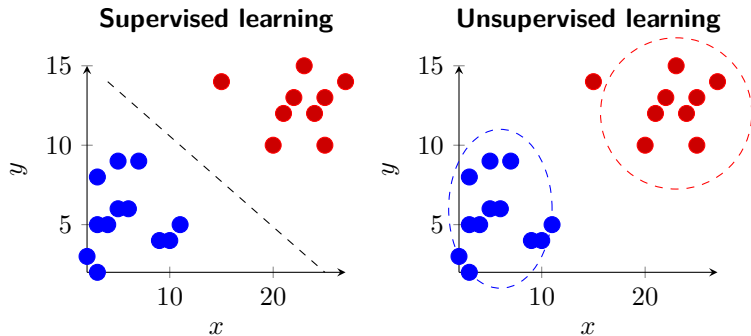


Figure 2: Supervised vs unsupervised ML

# Example of modelling data I

Imagine an unsupervised learning problem, with data represented by a vector  $\mathbf{x} = [x_1, \dots, x_p]^\top$ , a very general model is to assume that  $\mathbf{x}$  is the outcome of a random vector  $\mathbf{X} = [X_1, \dots, X_p]^\top$  with some unknown pdf  $f$ .

The model can be refined by assuming a specific form of  $f$ .



# Example of modelling data II

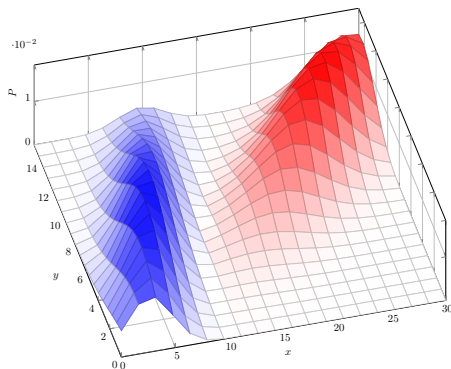


Figure 3: Some unknown pdf  $f$  from which data in Figure 3 was sampled.

# Example of method for unsupervised learning: K-means clustering

- 1 Specify the number of clusters  $K$
- 2 Randomly initialize the cluster centers (centroids)
- 3 Assign each data point to the closest centroid
- 4 recalculate the cluster centroids from the mean of the data points in the cluster
- 5 come back to step 3 if not converged

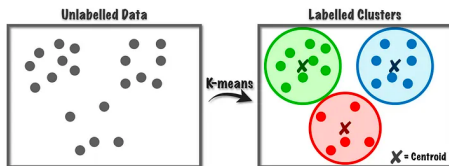
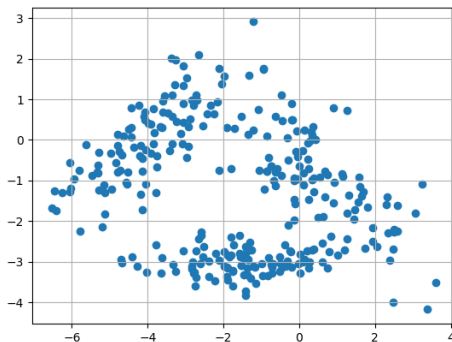


Figure 4: Clustering

# Example of unsupervised modelling

## Exercise 1      Unsupervised learning

Using the data in [this file](#), try to find 3 clusters using the K-means method.



# Tools to model data

**Function approximation** Model data with approximate and simple functions or maps.

**Optimization** Given a set of feasible mathematical models to the data, we may need to find the optimal one by fitting or calibrating a function to observed data.

**Probability and Statistics** Probability theory and statistical inference provides ways to quantify the uncertainty inherent in making predictions based on observed data.

## iid data

If we are given a sequence of data vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  one of the simplest possible models is to assume that the corresponding random vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  are independent and identically distributed (iid). We express this as:

$$\mathbf{x}_1, \dots, \mathbf{x}_n \stackrel{iid}{\sim} f$$

meaning that the random vectors form an iid sample from a pdf  $f$  or sampling distribution *Dist*.

This is the same as saying that knowing about one variable does not provide information about another variable.

# Independent data models

In independent data models, the joint density of the random vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  is the *product* of the marginal ones:

$$f_{\mathbf{x}_1, \dots, \mathbf{x}_n}(\mathbf{x}_1, \dots, \mathbf{x}_n) = f(\mathbf{x}_1) \cdots f(\mathbf{x}_n)$$

The function  $g()$ , our "model" for  $f()$ , is usually specified up to a small number of parameters, corresponding to :

- $\mathcal{N}(\mu, \sigma^2)$
- $\text{Bin}(n, p)$
- $\text{Exp}(\lambda)$

The parameters are typically obtained from the data.

# Modeling dilemma

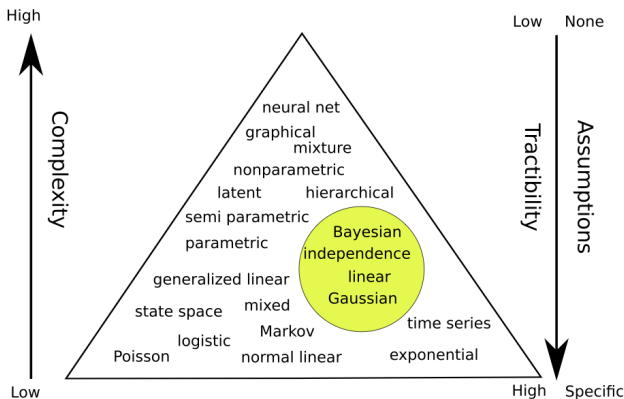


Figure 5: Complex models (very few of them) generally applicable but difficult to analyze. Simple models (a lot of options) very tractable but they do not describe well the data[?].

# Tradeoff

There exists a tradeoff between model tractability and applicability, as seen in Figure 5. Coming back to the example in page 8, the *training set*  $\tau = \{x_1, \dots, x_n\}$  is viewed as the outcome of  $n$  iid random variables  $x_1, \dots, x_n$  for some unknown pdf  $f$ .

**Goal:** to learn or estimate  $f$  from the finite training set.



# Tradeoff vs risk

Imagine the **unsupervised learning** framework shown before. We can specify a class (a collection) of pdfs that we will call  $\mathcal{G}_p$ :

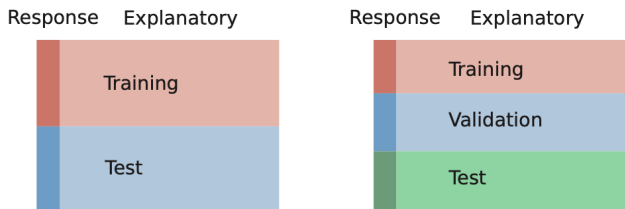
- We seek within  $\mathcal{G}$  the best approximation to the true model pdf  $f()$ , and we will call it  $g(|\Theta)$ .
- Such best approximation will minimize some calculated risk.

$$\text{Loss}(f(), g(|\Theta)) = \ln f() - \ln g(|\Theta)$$

with expected value, this is, the **risk** as

$$\ell(g) = \mathbb{E} \ln \frac{f()} {g(|\Theta)} = \int f() \ln \frac{f()} {g(|\Theta)} d$$

# Train-Test-Validate



**Figure 6:** Sometimes we use the second set of data for model validation. Then we need to use a third one for testing.

# Train-Test-Validate

To compare the predictive performance of various learners in  $\mathcal{G}$ , as measured by the test loss,

- we can use the same fixed training set  $\tau$  and test set  $\tau'$  for different learners, or
- if the overall data set is of modest size, we can perform the validation phase (model selection) on the training set only, using **cross-validation**.

# Polynomial regression. Original data.

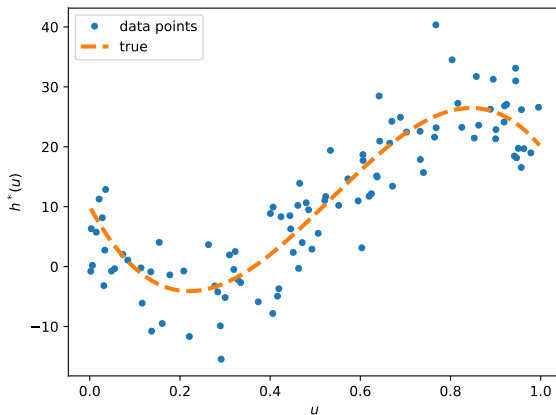


Figure 7: Training data and the optimal polynomial prediction function  $h^*$

# Polynomial regression. Fitting.

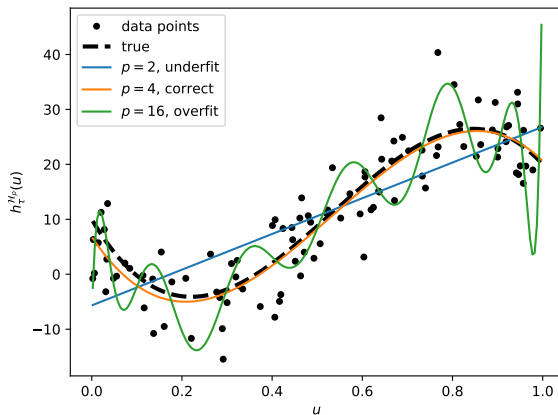


Figure 8: Fitted models for different orders of polynomial regressions[?]

# Polynomial regression. Error.

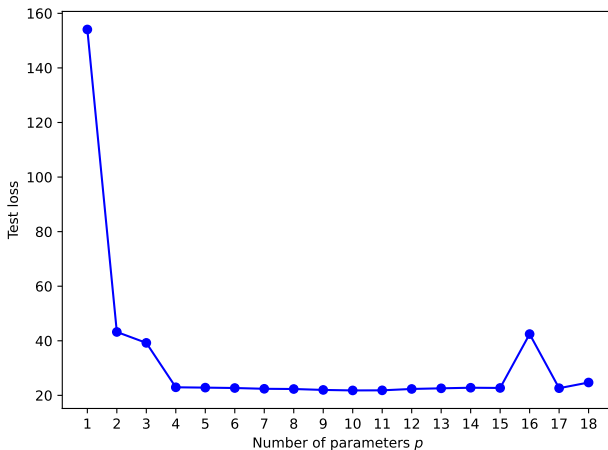
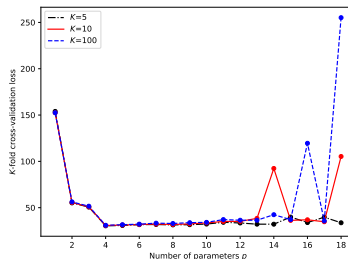
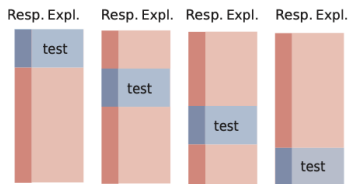


Figure 9: Fitted models for different orders of polynomial regressions[?]

# Polynomial regression. Cross validation.



**Figure 10:** a) Example of four-fold cross-validation, representing four copies of the same data set. The data in each copy is partitioned into a training set (pink) and a test set (blue). Darker columns are the response variable; and lighter ones the explanatory variables. b) K-fold cross-validation for the polynomial regression[?].



Dirk P. Kroese, Zdravko Botev, Thomas Taimre, and Radislav Vaisman.

*Data Science and Machine Learning: Mathematical and Statistical Methods.*

Machine Learning & Pattern Recognition. Chapman & Hall/CRC, 2020.



Junjie Peng, Elizabeth Jury, Pierre Dönnès, and Coziana Ciurtin.

Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges.

*Frontiers in Pharmacology*, 12, September 2021.



# Annex: detailed notation I

Given an input or *feature* vector , ML aims at predicting an output or *response* variable vector . In particular, we search for a mathematical *prediction function*  $g$  such that we can *guess* an approximation to  $\hat{y}$ :

$$\begin{aligned} g: \mathcal{X} &\rightarrow \mathcal{Y} \\ &\mapsto \hat{y} = g(x) \end{aligned}$$

## Definition

Dataset  $S = \{i\}_{i=1}^n = \{(x_i, y_i)\}_{i=1}^n$  is sampled from a distribution  $\mathcal{D}$  over a domain  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ .

$\mathcal{X}$  is the instance domain (a set),  $\mathcal{Y}$  is the label domain (a set), and  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  is the example domain (a set).

Usually,  $\mathcal{X}$  is a subset of  $\mathbb{R}^d$  and  $\mathcal{Y}$  is a subset of  $\mathbb{R}^{d_o}$ , where  $d$  is the input dimension,  $d_o$  is the output dimension.

## Annex: detailed notation II

$n = \#S$  is the number of samples. Without specification,  $S$  and  $n$  are for the training set.

- In *regression* problems,  $\mathbf{y}$  is a vector of real values.
- In *classification* problems,  $\mathbf{y}$  values lie within a finite set of  $c$  categories:  $y \in \{0, 1, \dots, c-1\}$ .

### Definition

A hypothesis space is denoted by  $\mathcal{H}$ . A hypothesis function is denoted by  $f_{\theta\theta}() \in \mathcal{H}$  or  $f(; \theta\theta) \in \mathcal{H}$  with  $f_{\theta\theta} : \mathcal{X} \rightarrow \mathcal{Y}$ .

$\theta\theta$  denotes the set of parameters of  $f_{\theta\theta}$ .

If there exists a target function, it is denoted by  $f^*$  or  $f : \mathcal{X} \rightarrow \mathcal{Y}$  satisfying  $y_i = f^*(x_i)$  for  $i = 1, \dots, n$ .

## Annex: detailed notation III

A loss function, denoted by  $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_+ := [0, +\infty)$ , measures the difference (or error) between a predicted label and a true label, e.g.,  $L^2$  loss:

$$\ell(f_{\theta\mathbf{0}}, y) = \frac{1}{2}(f_{\theta\mathbf{0}}() - y)^2,$$

where  $\mathbf{0} = (, )$ .  $\ell(f_{\theta\mathbf{0}}, y)$  can also be written as

$$\ell(f_{\theta\mathbf{0}}(), y)$$

for convenience.

(In the case of a classification,  $\ell(f_{\theta\mathbf{0}}, y) = \mathbb{1}\{y \neq \hat{y}\}$ )

We will see other useful loss functions (cross entropy or *hinge* loss functions) later in this course.

It is unlikely that a mathematical function  $g \equiv f_{\theta\mathbf{0}} : \mathcal{X} \rightarrow \mathcal{Y}$  would be able to make accurate predictions of all possible pairs  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ .

## Annex: detailed notation IV

So, we use a probabilistic approach here to empirical risk or training loss for a set  $S = \{(i, i)\}_{i=1}^n$  is denoted by  $L_S(\theta_0)$  or  $L_n(\theta_0)$  or  $R_n(\theta_0)$  or  $R_S(\theta_0)$ ,

$$L_S(\theta_0) = \frac{1}{n} \sum_{i=1}^n \ell(f_{\theta_0}(i), i). \quad (1)$$

The population risk or expected loss is denoted by  $L_{\mathcal{D}}(\theta_0)$  or  $R_{\mathcal{D}}(\theta_0)$

$$L_{\mathcal{D}}(\theta_0) = \mathbb{E}_{\mathcal{D}} \ell(f_{\theta_0}(), ), \quad (2)$$

where  $= (, )$  follows the distribution  $\mathcal{D}$ .

(In the case of a classification, we denote  $L_{\mathcal{D}}(g) = L_{\mathcal{D}}(\theta_0) = \mathbb{P}_{\mathcal{D}}[f_{\theta_0}() \neq ]$  and we say that  $g$  is a classifier.)

Because we are interested in minimizing the risk in our prediction, we are looking for the best possible  $g^* := \operatorname{argmin}_g \mathbb{E}_{\mathcal{D}} \ell(f_{\theta_0}(), )$

# Annex: detailed notation V

(In classification, we look for  $g^*() = \operatorname{argmax}_{y \in \{0,1,\dots,c-1\}} \mathbb{P}[Y = y \mid X = x].$ )

## Theorem

*For the squared-error loss  $\ell(y, \hat{y}) = (y - \hat{y})^2$ , the optimal prediction function  $g^*$  is equal to the conditional expectation of  $Y$  given  $X$ .*

which leads to write the random response  $Y$  as:

$$Y = g^*(X) + \varepsilon(X)$$

Note that such random deviation satisfies  $\mathbb{E}\varepsilon(X) = 0$