# Monte Carlo Methods

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya
Study Abroad

*jordi.villa@uvic.cat*

course 2023-2024

# Índex



FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Preliminary note

The material in these slides is strongly based on [1]. 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:

- Crude Monte Carlo.
- Monte Carlo Integration.

# Estimating expectation

Suppose we want to compute the expectation for a random variable $Y$:

$$\mathbb{E}Y = \mu = \begin{cases} \int yf(y)\,\mathrm{d}y & (\mathrm{continuouscase}) \\ \sum yf(y) & (\mathrm{discretecase}) \end{cases}$$

Many time knowing $f(y)$ is not possible ($Y$ may be a function of several other random variables).

ith Crude Monte Carlo (CMC) you can approximate $\mu$ by simulating many independent copies $Y_i, \ldots, Y_N$ of $Y$ and then take their sample mean as an estimator of $\mu$.

# First integration with CMC I

Imagine we want to estimate the value of a given integral $I = \int_{-\pi}^{\pi} \cos x \, \mathrm{d}x$. By the Average Value Theorem, we know that:

$$< f(x) >= \frac{1}{b-a} \int_a^b f(x) \, \mathrm{d}x$$

from which we obtain

$$\int_a^b f(x) \, \mathrm{d}x = (b-a) < f(x) >$$

Certainly:

# First integration with CMC II

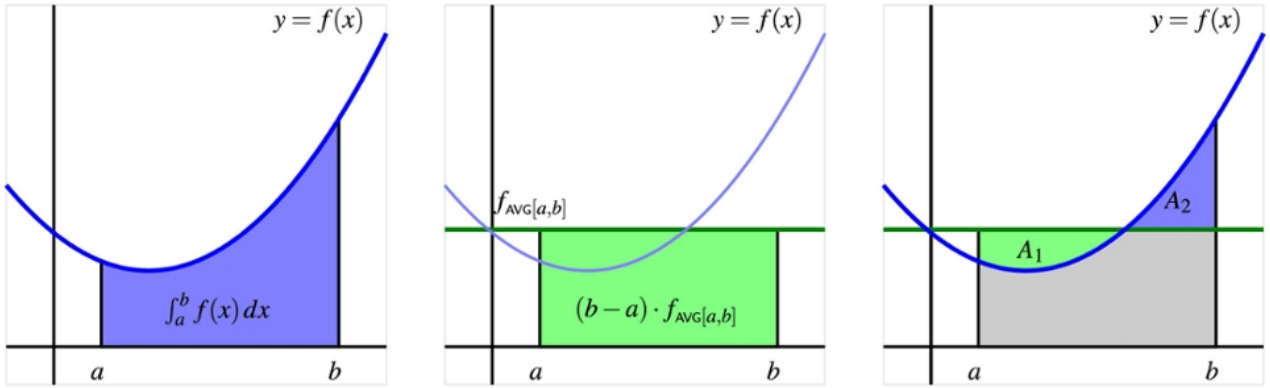


Figure 1: Visual inspection of the Average Value Theorem.

So, we can device an algorithm that takes values of the function and obtain their average to estimate the value of the requested integral.

# First integration with CMC III

```
from scipy import random
import numpy as np

a = -np.pi
b = np.pi
N = 1000

ar = np.zeros(N)

for i in range (len(ar)):
    ar[i] = random.uniform(a,b)

integral = 0.0

def f(x):
```

# First integration with CMC IV

```
    return np.cos(x)

for i in ar:
    integral += f(i)

ans = (b-a)/float(N)*integral

print ("Approx to the integral by CMC: {}.".format(ans
```

# Central limit theorem

In many situations, for independent and identically distributed (iid) random variables, the sampling distribution of the standardized sample mean tends towards the standard normal distribution even if the original variables themselves are not normally distributed.

Thus, $\bar{Y}$ approximately has a $\mathcal{N}(\mu, \sigma^2/N)$ distribution for large $N$, provided that $\mathrm{Var}\, Y < \infty$.

Thus we can construct an approximate $(1-\alpha)$ confidence interval for $\mu$:

$$\left(\bar{Y} - z_{1-\alpha/2}\frac{S}{\sqrt{N}}, \bar{Y} + z_{1-\alpha/2}\frac{S}{\sqrt{N}}\right)$$

where $S$ is the sample standard deviation of $Y_i$ and $z_\gamma$ is the $\gamma$-quantile of the $\mathcal{N}(0,1)$ distribution. Estimated standard error: $S/\sqrt{N}$; estimated relative error: $S/(\bar{Y}\sqrt{N})$.

FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Algorithm for CMC

**Algorithm 1:** CMC for iid

**Input:** Random variable $Y \sim f$, sample size $N$, confidence level $1-\alpha$.

**Output:** Point estimate and approximate $(1-\alpha)$ confidence interval for $\mu = \mathbb{E}Y$.

1 Simulate $Y_1, \ldots, Y_N \overset{iid}{\sim} f$

2 $\bar{Y} \leftarrow \frac{1}{N}\sum_{i=1}^{N} Y_i$

3 $S^2 \leftarrow \frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \bar{Y})^2$

4 **return** $\bar{Y}$ *and conf. interval* $\left(\bar{Y} - z_{1-\alpha/2}\frac{S}{\sqrt{N}}, \bar{Y} + z_{1-\alpha/2}\frac{S}{\sqrt{N}}\right)$

FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Monte Carlo integration

Consider the complicated integral

$$\mu = \int_{-\infty}^{infty} \int_{-\infty}^{infty} \int_{-\infty}^{infty} \sqrt{|x_1 + x_2 + x_3|} e^{-(x_1^2+x_2^2+x_3^2)/2} \, \mathrm{d}x_1 \, \mathrm{d}x_2 \, \mathrm{d}x_3$$

Defining $Y = |X_1 + X_2 + X_3|^{1/2}(2\pi)^{3/2}$ with $X_1, X_2, X_3 \overset{iid}{\sim} \mathcal{N}(0,1)$, we can write $\mu = \mathbb{E}Y$. Use the code here to test the calculation. If you want to learn more about CLT and confidence intervals, a good start can be found here.

# Polynomial regression. Original data.

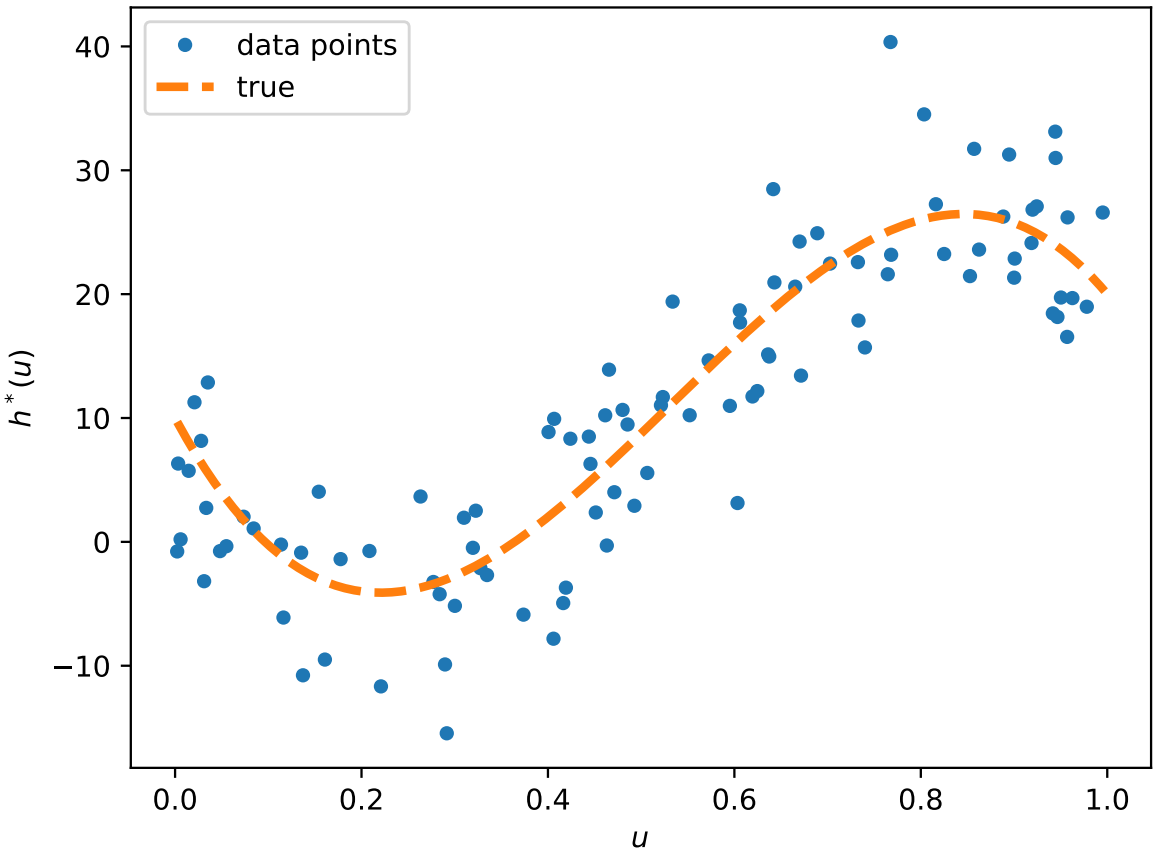


Figure 2: Training data and the optimal polynomial prediction function $h^*$[1]

# Polynomial regression. Estimating the generalization risk

The code here shows how to estimate how good is the graph below.

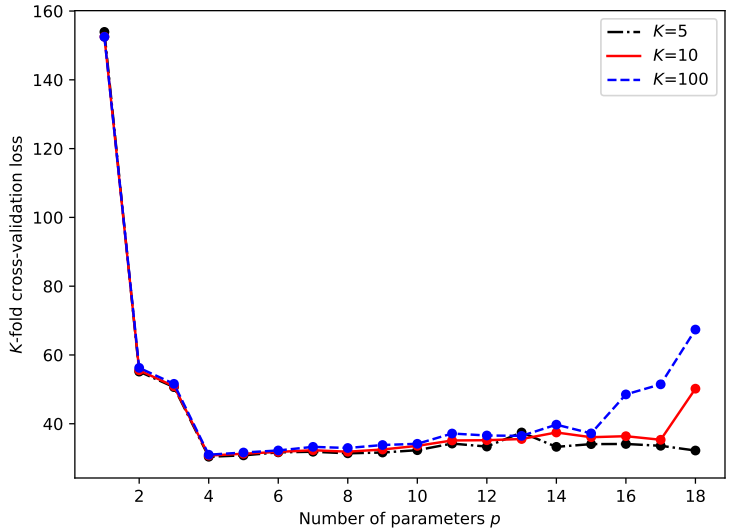


Figure 3: K-fold cross-validation for the polynomial regression[1].


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.

# Annex: detailed notation I

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

$$g\colon \mathcal{X} \to \mathcal{Y}$$
$$\boldsymbol{x} \mapsto \hat{\boldsymbol{y}} = g(\boldsymbol{x})$$

## Definition

Dataset $S = \{\boldsymbol{z}_i\}_{i=1}^n = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ 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).

# Annex: detailed notation II

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

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

## Definition

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

$\boldsymbol{\theta}$ denotes the set of parameters of $f_{\boldsymbol{\theta}}$.
If there exists a target function, it is denoted by $f^*$or $f : \mathcal{X} \to \mathcal{Y}$ satisfying $\boldsymbol{y}_i = f^*(\boldsymbol{x}_i)$ for $i = 1, \ldots, n$.

FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Annex: detailed notation III

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

$$\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z}) = \frac{1}{2}(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{y})^2,$$

where $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$. $\ell(f_{\boldsymbol{\theta}}, \boldsymbol{z})$ can also be written as

$$\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})$$

for convenience.
(In the case of a classification, $\ell(f_{\boldsymbol{\theta}}, \boldsymbol{y}) = \mathbb{1}\{y \neq \hat{\boldsymbol{y}}\}$)
We will see other useful loss functions ({em cross entropy} or *hinge* loss functions) later in this course.
It is unlikely that a mathematical function $g \equiv f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$ would be able to make accurate predictions of all possible pairs $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Annex: detailed notation IV

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

$$L_S(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i). \tag{1}$$

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

$$L_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{D}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}), \tag{2}$$

where $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ follows the distribution $\mathcal{D}$.
(In the case of a classification, we denote $L_{\mathcal{D}}(g) \equiv L_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{P}_{\mathcal{D}}[f_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq \boldsymbol{y}]$ 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 $g^* := \operatorname{argmin}_g \mathbb{E}_{\mathcal{D}} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})$

FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

# Annex: detailed notation V

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

**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 $\boldsymbol{X} = \boldsymbol{x}$.*

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

$$Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$$

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