

Machine Learning 1

Introduction

Master 1 Computer Science

Yohann de Castro, Aurélien Garivier

2019-2020



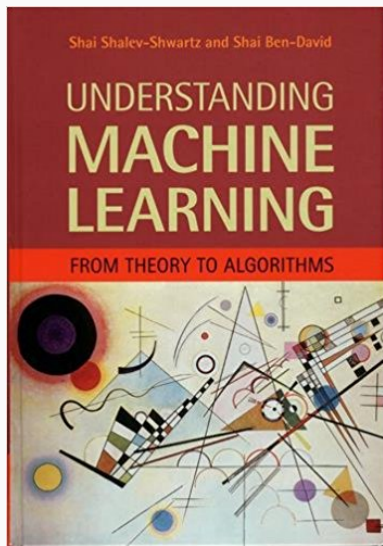
Table of contents

1. Before we start
2. What is Machine Learning?
3. The Learning Models
4. Machine Learning Methodology
5. Statistics 101

Before we start

Outline

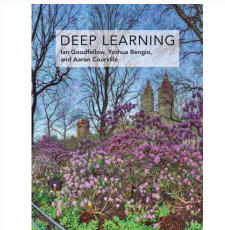
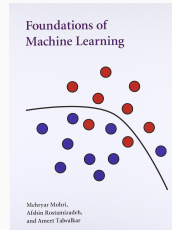
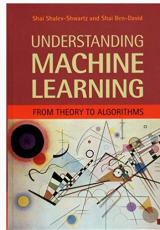
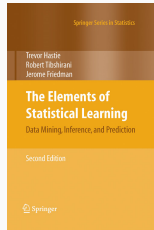
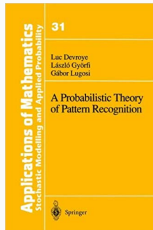
- 1. 01.17 Introduction to ML, Statistics 101
- 2. 01.24 Clustering
- 3. 01.31 Dimensionality Reduction: PCA, random projections
- 4. 02.07 Supervised Learning, Nearest Neighbors
- 5. 02.14 Bias-Variance Tradeoff, CART
- 6. 02.21 Ensemble methods: Boosting, Bagging, Random Forests
- 7. 02.28 Empirical risk minimization, Linear Separators
- 03.06 holidays
- 8. 03.13 Structural Risk Minimization, Kernels, Regularization
- 9. 03.20 Neural networks and stochastic gradient descent
- 10. 03.27 Online Learning
- 11. 04.03 Revisions
- 12. 04.10 free
- 13. 04.17 Final Exam



General introduction to Machine Learning theory, by two leading researchers of the field.

Covers a good part of the content of this course (other references will be provided for specific topics).

Additional References



- 50% final exam
- 50% exercises and project; bonus for scribes

Project: a "challenge-like" data science problem (to be presented later).
By groups of 4-5.

What is Machine Learning?

Why Machine Learning?

Actualité

Yann LeCun, Geoffrey Hinton et Yoshua Bengio reçoivent le prix Turing

Par Stéphane Nachez - 27 mars 2019



LE MACHINE LEARNING PROVOQUE UNE CRISE DANS LE DOMAINE DE LA SCIENCE

✎ Bastien L. | 19 février 2019 | Analytics, Data Analytics, Intelligence artificielle | 1 commentaire

Le Machine Learning est en train de provoquer une grave crise de reproductibilité dans le domaine de la science. C'est ce qu'affirme la statisticienne Genevera Allen de la Rice University dans le cadre de la conférence AAAS Annual Meeting.

De plus en plus de chercheurs utilisent le **Machine Learning** pour analyser des données et y détecter des tendances. Cependant, dans le cadre de la conférence scientifique AAAS Annual Meeting, la statisticienne Genevera Allen de la Rice University a tenu à tirer la sonnette d'alarme. Selon elle, le Machine Learning est en passe de provoquer **une crise de reproductibilité dans le domaine de la science.**

SHARE SPECIAL VIEWPOINTS



Machine Learning for Science: State of the Art and Future Prospects

Eric Mjølhus, Dennis DeCoste
* See all authors and affiliations

Science 14 Sep 2019
Vol. 365, Issue 6402, pp. 2051-2055
DOI: 10.1126/science.1263.0037.2019

Article Figures & Data Info & Metrics eLetters PDF

Abstract

Recent advances in machine learning methods, along with successful applications across a wide variety of fields such as planetary science and bioinformatics, promise powerful new tools for practicing scientists. This viewpoint highlights some useful characteristics of modern machine learning methods and their relevance to scientific applications. We conclude with some speculations on near-term progress and promising directions.

PUBLIC RELEASE: 15-FEB-2019

Can we trust scientific discoveries made using machine learning?

Rice U. expert: Key is creating ML systems that question their own predictions

RICE UNIVERSITY

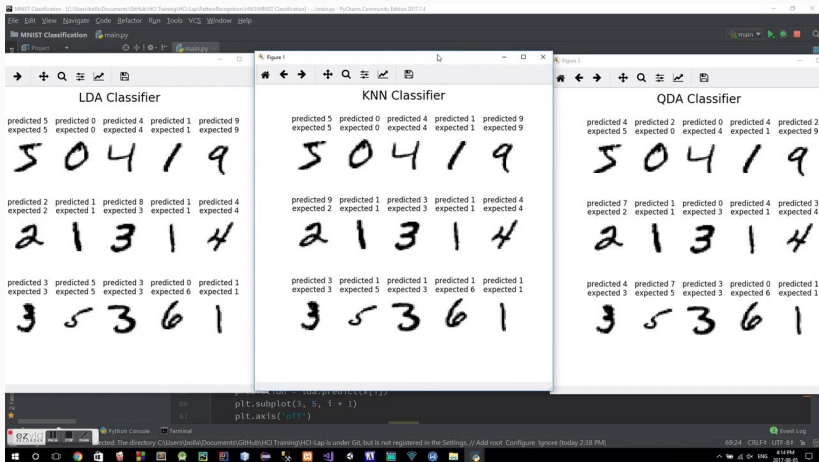
What is Machine Learning?

- Algorithms operate by building a model from **example** inputs in order to make data-driven **predictions or decisions**...
- ...rather than following strictly static program instructions: useful when designing and programming explicit algorithms is unfeasible or poorly efficient.

Within Artificial Intelligence

- evolved from the study of pattern recognition and computational learning theory in artificial intelligence.
- AI: emulate cognitive capabilities of humans (big data: humans learn from abundant and diverse sources of data).
- a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

Example: MNIST dataset



Machine Learning (ML): Definition

Arthur Samuel (1959)

Field of study that gives computers the ability to learn without being explicitly programmed

Tom M. Mitchell (1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .

Machine Learning: Typical Problems

- spam filtering, text classification
- optical character recognition (OCR)
- search engines
- recommendation platforms
- speech recognition software
- computer vision
- bio-informatics, DNA analysis, medicine
- etc.

For each of this task, it is possible but very inefficient to write an explicit program reaching the prescribed goal.

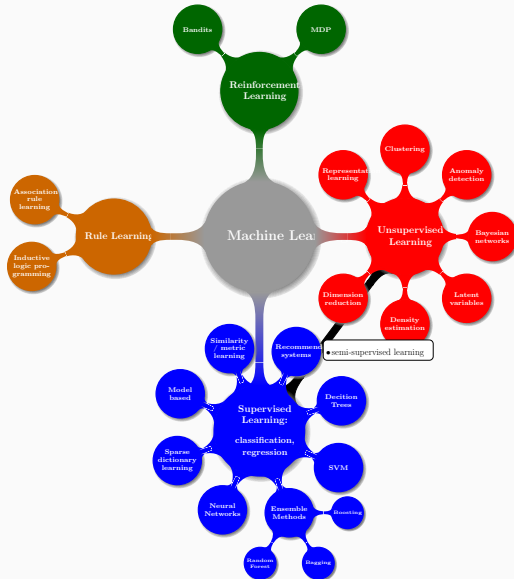
It proves much more succesful to have a machine infer what the good decision rules are.

What is Statistical Learning?

- = Machine Learning using statistics-inspired tools and guarantees
 - Importance of **probability**- and **statistics**-based methods
→ **Data Science** (Michael Jordan)
 - **Computational Statistics**: focuses in prediction-making through the use of computers together with statistical models (ex: Bayesian methods).
 - **Data Mining** (unsupervised learning) focuses more on exploratory data analysis: discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.
 - Machine Learning has more **operational** goals
Ex: ~~consistency~~ → oracle inequalities
Models (if any) are *instrumental*.
ML more focused on *correlation*, less on *causality* (now changing).
 - Strong ties to **Mathematical Optimization**, which furnishes methods, theory and application domains to the field

The Learning Models

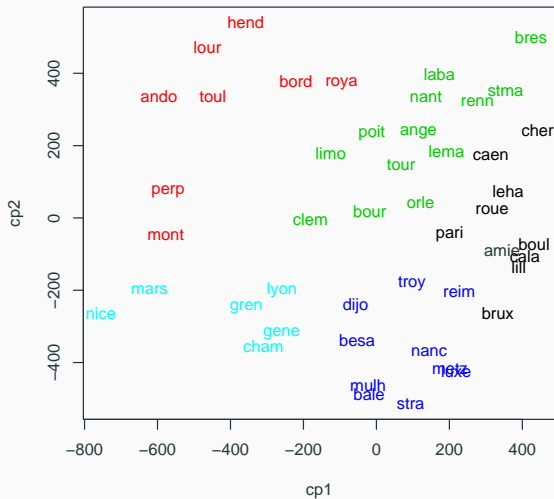
What ML is composed of



Unsupervised Learning

- (many) observations on (many) individuals
- need to have a simplified, structured overview of the data
- *taxonomy*: untargeted search for *homogeneous clusters* emerging from the data
- Examples:
 - customer segmentation
 - image analysis (recognizing different zones)
 - exploration of data

Example: representing the climate of cities



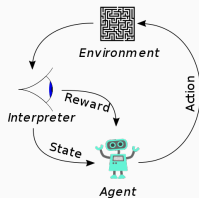
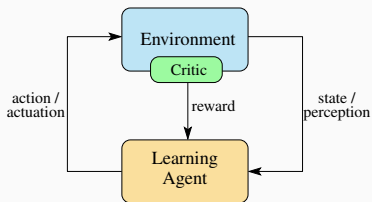
Supervised Learning

- Observations = pairs (X_i, Y_i)
- Goal = learn to *predict* Y_i given X_i
- Regression (when Y is continuous)
- Classification (when Y is discrete)

Examples:

- Spam filtering / text categorization
- Image recognition
- Credit risk ranking

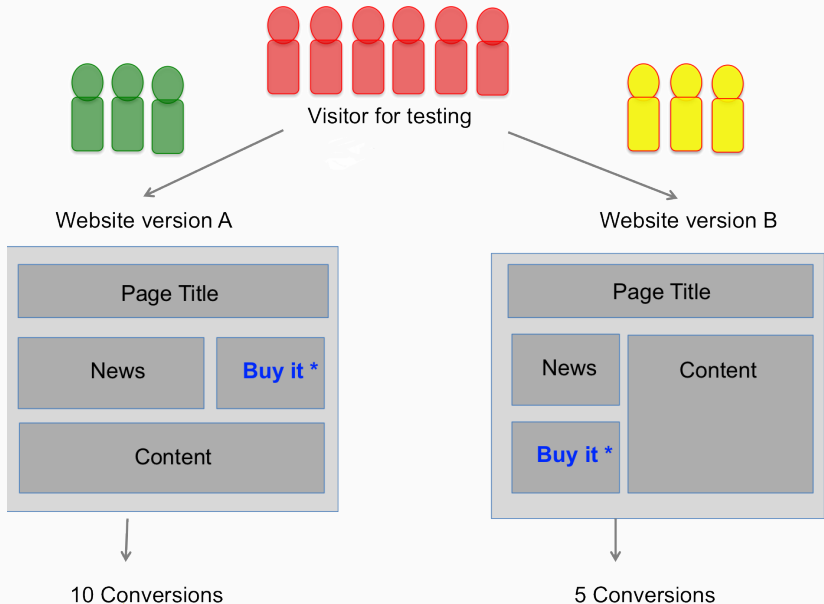
Reinforcement Learning



[Src: https://en.wikipedia.org/wiki/Reinforcement_learning]

- area of machine learning inspired by behaviourist psychology
- how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
- Model: random system (typically : Markov Decision Process)
 - agent
 - state
 - actions
 - rewards
- sometimes called approximate dynamic programming, or neuro-dynamic programming

Example: A/B testing



Machine Learning Methodology

n -by- p matrix X

- n examples = points of observations
- p features = characteristics measured for each example

Questions to consider:

- Are the features centered?
- Are the features normalized? bounded?

In `scikitlearn`, all methods expect a 2D array of shape (m, p) often called

`X (n_samples, n_features)`

- Inside R: package datasets
- Inside scikitlearn: package sklearn.datasets
- UCI Machine Learning Repository



- Challenges: Kaggle, etc.

The big steps of data analysis

1. Extracting the data to expected format
2. Exploring the data
 - detection of outliers, of inconsistencies
 - descriptive exploration of the distributions, of correlations
 - data transformations

 - learning sample
 - validation sample
 - test sample
3. For each algorithm: parameter estimation using training and validation samples
4. Choice of final algorithm using testing sample, risk estimation

Machine Learning tools: R

Activities | Studio | Mon, 19 Feb 2024 16:10

File Edit Code View Plots Session Build Debug Tools Help

```

264 plot(TT, RR[,1], type="l", lwd=4, lty=lty[s], log="x", xlim=c(min(TT), max(TT)), ylim = c(0, max(RR[
265 #arrrow(TT, RR[,1-2^SD[1],]/sqrt(Nmc), TT, RR[,1+2^SD[1],]/sqrt(Nmc), length=0.1, lwd=4, col=3, log
266 #for (j in 1:nstrategies){
267 lines(TT, RR[,1, lwd=4, lty=lty[j])
268 f <- lsm(NMC, TT=10000) ~ log(TT*(1+10000))
269 lines(TT*(1+500), rcoefficients[1] + rcoefficients[2]*log(TT*(1+500)), col="red", lty=lty[j])
270 slopes[j] <- rcoefficients[2]^4
271 #arrrow(TT, RR[,1-2^SD[1],]/sqrt(Nmc), TT, RR[,1+2^SD[1],]/sqrt(Nmc), length=0.1, lwd=4, col=2, l
272 }
273 #for (h in c(0.5, 1, 2, 4)){ lines(TT, h*log(TT^h+2^4)/d, col="red", lwd=2) }
274
275 #plot(TT, pRR[,j,]/log(TT), type="l", lwd=4, lty=lty[s], log="x", xlim=c(min(TT), max(TT)), ylim = c
276 #arrrow(TT, RR[,1-2^SD[1],]/sqrt(Nmc), TT, RR[,1+2^SD[1],]/sqrt(Nmc), length=0.1, lwd=4, col=3, log
277 #for (j in 2:nBstrategies){
278 # lines(TT, pRR[,j,]/log(TT), lwd=4, lty=lty[j])
279 #arrrow(TT, RR[,1-2^SD[1],]/sqrt(Nmc), TT, RR[,1+2^SD[1],]/sqrt(Nmc), length=0.1, lwd=4, col=3, l
280 #}
281
282 names <- c("FB-ETC", "SPRT-ETC", "BAI-ETC", "D-UCB", "UCB") #??-UCB"
283 order <- c(1,3,5,2,4)
284 legend(50, 80, slopej[order, function(k) paste(names[k], "\t", round(slopes[k], 2))], lty = ltyj[order
285
286
287 # T <- c(2.5, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 600, 800, 1000, 1500, 2000, 3000, 5000, 8000, 10000)/d^2; Nmc
288 # T <- c(2.5, 10, 20, 30, 50, 100, 150, 200, 300, 400, 500, 600, 800, 1000)/d^2; Nmc <- 10000; createData(T, Nmc);
289 # T <- c(10, 20, 20^2); Nmc <- 1000; createData(T, Nmc);
290 # T <- c(400, 800, 1500)/d^2; Nmc <- 10000; createData(T, Nmc);
  
```

```

Console | View | Top Level | 6 Script
[1] 0.91119 0.00303 1.00000 0.00039 0.00618
[1] 10000
[1] 72.72482 34.05975 53.29412 27.32934 28.02340
[1] 0.00027 0.50208 1.00000 0.00023 0.00425
[1] 12500
[1] 77.44936 36.14412 56.56396 28.23063 49.41559
[1] 0.00719 0.00177 1.00000 0.00021 0.00361
[1] 15000
[1] 80.50309 36.92575 59.43862 29.01284 41.89409
[1] 0.00548 0.00148 1.00000 0.00013 0.00272
[1] 20000
[1] 85.12948 38.00491 63.51994 29.99955 44.16748
[1] 0.00435 0.00107 1.00000 0.00016 0.00210
[1] 25000
[1] 89.93376 39.54227 66.87458 30.88489 45.81386
[1] 0.00370 0.00083 1.00000 0.00012 0.00166
[1] 37500
[1] 96.90037 42.01918 72.95729 32.54084 49.32925
[1] 0.00236 0.00050 1.00000 0.00008 0.00127
[1] 50000
[1] 102.77196 42.75523 77.29644 33.54500 51.36300
[1] 0.00177 0.00030 1.00000 0.00008 0.00080
  
```

RStudio

Environment | History

Global Environment

Values

d	0.2
dUCB	List of 9
dETC	List of 9
dETCUG	List of 8
Fb	List of 4
DayData	Large List (17 elements, 524.5 Mb)
UCB	List of 7

Functions

createData	function (T, Nmc)
getData	function ()
loadData	function (data, data2, data3)
mergeResults	function (data1, data2, data3)
myZero	function (f, l, u)

Files | Plots | Packages | Help | Viewer

Zoom | Export

Project: (None)

Machine Learning tools: python

The image shows a Python IDE (Spyder) with a script for Kaplan-Meier survival analysis. The script defines a function `kaplan_meier` that takes `Delta` and `censorDate` as input. It calculates the survival function `S` and the NA estimate. The plot shows the NA estimate (blue line) and the survival function (black line) over time.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Aug 16 09:05:03 2016
4
5 @author: agarvis
6 """
7
8 import numpy as np
9 from random import random
10 import matplotlib as mpl
11 import matplotlib.pyplot as plt
12
13
14 def kaplan_meier(Delta, censorDate):
15     k = len(Delta)
16     atrisk = np.zeros(k) # atrisk before Delta[i]
17     surv = np.zeros(k) # dnt between Delta[i] and Delta[i+1]
18     for i in k:
19         s = 0
20         while j < k and x[i] + Delta[i] < censorDate and x[i] + Delta[i] < x[i+1]:
21             atrisk[j] += 1
22             s[j] + Delta[i+1] < x[i+1]:
23                 surv[j+1] += 1
24         s = np.concatenate([s, np.cumprod(survived/atrisk)])
25     return(s)
26
27 # atrisk: index shift of 1 - see kaplan_meier_2 which conforms to package
28
29
30 def plot_survival():
31     N = 10000
32     n = 2
33     x = [random() for k in range(N)]
34     y = [1 - random() for k in range(N)]
35     Delta = np.array([float(x)/N for i in range(N)])
36     print(Delta)
37
38     # plot: atrisk, Delta, now
39     # plot: hat(S)
40     # plot: hat(S) - Delta
41     # plot: hat(S) - y
42
43     # plot: surv - 0.85, 1, 0.1
44
45
46 def kaplan_meier(Delta, E):
47     I = np.arange(1, E)
48     T = [I for k in I]
49     e = [I for k in I]
50     n = len(I)
51     S = np.zeros(n)
52     H = np.zeros(n)
53     for k in range(n-1):
54         S[k+1] = S[k] + (n-k-1)/(n-k)
55         H[k+1] = H[k] + e[k]/(n-k)
```

Figure 2: A plot showing the NA estimate (blue line) and the survival function (black line) over time. The x-axis is labeled 'smetime' and ranges from 0.0 to 0.6. The y-axis ranges from 0.0 to 1.4. The NA estimate starts at 0.0 and increases to approximately 1.0 at time 0.6. The survival function starts at 1.0 and decreases to approximately 0.8 at time 0.6.

smetime	NA estimate	survival function
0.0	0.0	1.0
0.1	0.1	0.9
0.2	0.2	0.8
0.3	0.3	0.7
0.4	0.4	0.6
0.5	0.5	0.5
0.6	0.6	0.4

scikit-learn
Machine Learning In Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, ... — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Modules: grid search, cross validation, metrics, ... — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extraction, ... — Examples

News

On-going development: [What's new \(Changelog\)](#)

Community

About us See authors and contributing
More Machine Learning Find related

Who uses scikit-learn?

Knime, Weka and co: integrated environments

The screenshot displays the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Use training set' selected. The 'Classifier output' window displays the following summary:

```
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      144      96 %
Incorrectly Classified Instances     6       4 %
Kappa statistic                    0.94
Mean absolute error                 0.035
Root mean square error              0.035
```

The 'Tree View' window shows a decision tree structure:

```
graph TD
    Root((petalwidth)) -- "<= 0.6" --> Node1[Iris-setosa (50.0)]
    Root -- "> 0.6" --> Node2((petalwidth))
    Node2 -- "<= 1.7" --> Node3((petalength))
    Node2 -- "> 1.7" --> Node4[Iris-virginica (46.0/1.0)]
    Node3 -- "<= 4.9" --> Node5[Iris-versicolor (48.0/1.0)]
    Node3 -- "> 4.9" --> Node6((petalwidth))
    Node6 -- "<= 1.5" --> Node7[Iris-virginica (3.0)]
    Node6 -- "> 1.5" --> Node8[Iris-versicolor (3.0/1.0)]
```

The interface also includes a 'Result list' on the left, a 'Start' button, and a 'Visualize' section on the right showing a scatter plot of 'petalwidth (Num)' with 'Iris-versicolor' and 'Iris-virginica' classes.

Statistics 101

Statistical model

- Sample size: n
- Observation space: \mathcal{X}
- Statistical model = pair $(\mathcal{X}^n, \mathcal{P})$, where \mathcal{P} is a family of probability distributions on \mathcal{X}^n
- Observation: $(X_1, \dots, X_n) \sim P$ where $P \in \mathcal{P}$
- Parametric model : $\mathcal{P} = \{P_\theta : \theta \in \Theta \subset \mathbb{R}^d\}$
- Product model: $\mathcal{P} = \{Q^{\otimes n} : Q \in \mathcal{Q}\} \stackrel{param}{=} \{Q_\theta^{\otimes n} : \theta \in \Theta\}$
- Bernoulli model: parametric product model with $Q = \mathcal{B}(\theta), \Theta = [0, 1]$

Estimator

- Statistic = any function of (X_1, \dots, X_n) (and not θ !)
- Estimator of $g(\theta)$ = any statistic; a good estimator tries to be "close" to $g(\theta)$ "whatever the value of θ ."
- Ex: Bernoulli model: $\hat{\theta}_n = \bar{X}_n$, $\tilde{\theta}_n = X_1$, $\check{\theta}_n = 2(X_1 + \dots + X_{n/2})/n$
- Bias of T_n : $\theta \mapsto \mathbb{E}_\theta [T_n - g(\theta)]$
- Consistent: $T_n \xrightarrow{P} g(\theta)$ when $n \rightarrow \infty$
- Quadratic risk: $\theta \mapsto \mathbb{E}_\theta [(T_n - g(\theta))^2]$
- Minimax risk:

$$\inf_{T_n} \sup_{\theta \in \Theta} \mathbb{E}_\theta [(T_n - g(\theta))^2]$$

Minimax estimator: reaches the minimax risk

Definition: if $\theta = \phi\left(E_\theta[X_1], \dots, E_\theta[X_1^d]\right)$, then

$$\hat{g}_n = \phi\left(\frac{1}{n} \sum_{i=1}^n X_i, \dots, \frac{1}{n} \sum_{i=1}^n X_i^d\right)$$

Prop: if $E_\theta[X_1^d] < \infty$ and if ϕ is continuous, then \hat{g}_n is consistent

Ex: Bernoulli model $\theta = E[X_1] \rightarrow \hat{\theta}_n = \bar{X}_n$

More generally: if $g(\theta) = \mathbb{E}_\theta[X_1]$, then $\hat{g}_n = \bar{X}_n$

Remark: best constant guess = expectation

Ex: Gaussian model

Maximum Likelihood Estimator

Definition the likelihood function in a parametric model is

$$\ell(\theta, X_1, \dots, X_n) = \begin{cases} P_\theta(X_1, \dots, X_n) & \text{in a discrete model} \\ f_\theta(X_1, \dots, X_n) & \text{in a continuous model} \end{cases}$$

Definition The maximum likelihood estimator of θ is defined by

$$\hat{\theta}_n \in \arg \max_{\theta \in \Theta} \ell(\theta, X_1, \dots, X_n)$$

Ex: Bernoulli model:

$$\ell(\theta, X_1, \dots, X_n) = \prod_{i=1}^n p^{X_i} (1-p)^{1-X_i}$$

and $\hat{\theta}_n = \bar{X}_n$

Linear Regression

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i$$

with $\mathbb{E}[\epsilon_i] = 0$, $\text{Var}[\epsilon_i] = \sigma^2$ and (ϵ_i) independent.

Matrix form: $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, with $X_{i,0} = 1$, $\mathbf{X} = (X_{ij}) \in \mathcal{M}_{n,p+1}(\mathbb{R})$ and $\mathbf{Y} \in \mathbb{R}^n$ and ϵ random vector with range in \mathbb{R}^n .

Least Mean Square estimator:

$$\hat{\beta}_n = \arg \min_{\beta \in \mathbb{R}^{p+1}} \|\mathbf{Y} - \mathbf{X}\beta\| = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

if $\text{rank}(\mathbf{X}) = p + 1$.

- if $\epsilon_i \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$, the ML estimator is the LMS estimator and

$$\hat{\beta}_n \sim \mathcal{N}\left(\beta, \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}\right)$$

- simple regression: $p = 1$, $\hat{\beta}_{n,1} = \frac{\text{Cov}_n(\mathbf{X}, \mathbf{Y})}{\text{Var}_n(\mathbf{X})}$