Aprenentatge Automàtic 1

GCED

Lluís A. Belanche

belanche@cs.upc.edu

Soft Computing Research Group
Departament de Ciències de la Computació (Computer Science Department)

Universitat Politècnica de Catalunya - Barcelona Tech

2019-2020

LECTURE 1: Introduction to Machine Learning

What is this course about?

Motivation:

- We want **human-like** behaviour in **machines** (specially, in computers!): adaptable (flexible), fast, reliable and automatic

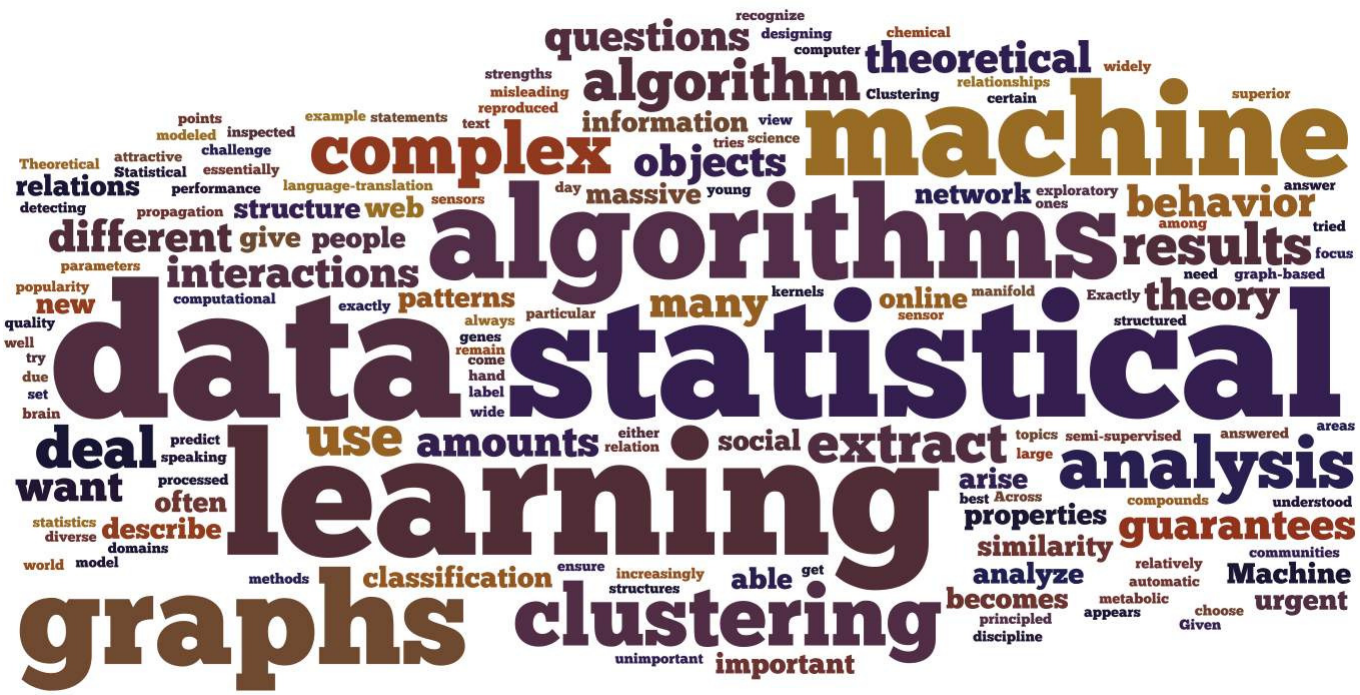
- **Biological systems** deal with imprecision, partial truth, uncertainties, noise, contradictions, ... mostly in a **data-driven** fashion (think of a baby learning to walk or talk)

- They also make **predictions** in intricate ways by own behavioral models (learnt by experience & almost impossible to verbalize)

Very difficult to achieve by direct programming (*brittleness*)

What is this course about?

Machine learning (ML) is a field that lies at the intersection of statistics, probability, computer science, and optimization. The main goal is to explore **automatic methods for inferring models from data** (*e.g.*, finding structure, making predictions).



Examples of learning tasks

SUPERVISED LEARNING uses labeled data

Classification: predicting a class (or category) to each example (e.g., document classification); note multi-label, probabilistic generalizations

Regression: predicting a real value for each example (e.g., prediction of ph concentration); note multi-variable generalization

UNSUPERVISED LEARNING does not use (or have) data labels

Clustering: discovering homogeneous groups in data (clusters)

Dimensionality reduction: finding lower-dimensional data representations

Density estimation: estimating the probabilistic mechanism that generates data

Novelty detection: finding anomalous/novel/outlying data

SEMI-SUPERVISED LEARNING uses partly labeled data

Ranking: ordering examples according to some criterion (e.g., web pages returned by a search engine).

Reinforcement: delayed rewarding (e.g., finding the way out in a maze)

TRANSFER LEARNING learning in a new task through the transfer of knowledge from a related task that has already been learned.

Introduction to Machine Learning

A system (living or not) learns if it uses *past* experience to improve *future* performance:

1. **Acquiring** more knowledge (or more abilities) with time, and

2. **Reorganizing** this knowledge such that some problems are solved:

   a) in a more *efficient* way (using less resources) or

   b) in a more *effective* way (higher performance standards)

Introduction to Machine Learning

I have an idea ...

Let the **machine** ...

1. **learn** the information contained in a data sample (the experience);
2. **build** a model that summarizes the regularities in the sample, and
3. **use** it to answer future queries

$\longrightarrow$ This is **Machine Learning**

Machine Learning in context

Machine Learning has strong bridges to other disciplines:

Statistics: inferential statistics, distribution and sampling theory, mathematical statistics

Data Mining: very large data bases, interest in high-level knowledge

Mathematics: optimization, numerical methods, asymptotics, ...

Algorithmics: correctness, complexity, ...

Artificial Intelligence: general aims at "intelligent" (?) behaviour

Machine Learning in context

Substantial relation to **Multivariate Statistics** (MVS):

- Many classical techniques in MVS are linear in nature: PCA, logistic regression, ridge and linear regression, Fisher's discriminant analysis, Canonical-correlation analysis, Factor Analysis, PLS, ...

- Many classical techniques in ML are non-linear: neural networks, kernel methods, random forests, ...

- Often the goals and problems are similar, and the techniques can often be rooted in the same theories

- Modern MVS is lagging behind in the analysis of complex data

Useful probability and statistics facts

The Central Limit Theorem (CLT) is one of the workhorses in statistics. If $X_1, \ldots, X_n$ are i.i.d. r.v.s (each having the same distribution) with $\mathbb{E}[X_i] = \mu$ and $Var(X_i) = \sigma^2$, then the sample mean

$$\frac{X_1 + \ldots + X_n}{n}$$

approximately has a $\mathcal{N}(\mu, \frac{\sigma^2}{n})$ distribution as $n \to \infty$.

- This holds true no matter what form the distribution of the X_i is
- How large n must be before the approximation is "good" depends on this distribution

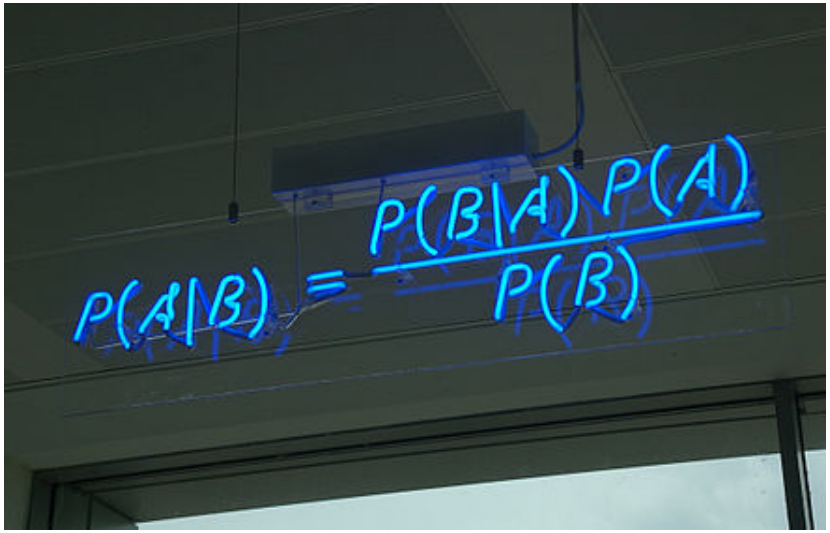
Useful probability and statistics facts

The product rule is a fundamental decomposition in probability. If $X_1, \ldots, X_n$ have a joint probability distribution $p(X_1, \ldots, X_n)$, then we can factorize the distribution as:

$$p(X_1, \ldots, X_n) = p(X_1) \prod_{i=2}^{n} p(X_i | X_1, \ldots, X_{i-1})$$

- There is no (even partial) independence assumption
- The number of terms is always linear in the number of variables

Useful probability and statistics facts



Useful probability and statistics facts

The **Bayes formula** (aka Bayes theorem or Bayes rule):

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_i P(A|B_i)P(B_i)}$$

In many random experiments, some specific event A occurs only when some other events $B_1, \ldots, B_n$ are met ("causes"), of which we know their probabilities. Assume we have already performed the experiment and that we know the probability $P(A|B_i)$.

The formula allows us to compute $P(B_i|A)$, i.e., the probability of the "cause" B_i given the "consequence" A of this cause.

This useful and modern interpretation is due to Laplace.

Useful probability and statistics facts

The **Bayes formula** in a data analysis context:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{\int_{\Theta} P(D|\theta)P(\theta)\,d\theta}$$

where θ is a random vector or random variable (e.g., the parameter(s) in a distribution or statistical model), and D is the available data, related to θ.

Notion of **likelihood**, **prior**, **posterior** and **unconditional** (expected likelihood) distributions:

$P(\theta)$: prior probability, confidence in θ prior to observing D

$P(D|\theta)$: likelihood, probability of observing data D if parameters are θ

$P(D)$: expected likelihood of observing data D

$P(\theta|D)$: posterior probability, confidence in θ after observing D

The Bayes formula

Frequentist view:

1. The parameters θ are constant (but unobserved)

2. The data is variable (that is, just a sample)

3. Probability understood as long-run frequency of event

4. Parameter **estimation** of a single θ; its uncertainty is given by the distribution of all possible datasets

The Bayes formula

Bayesian view:

1. The parameters θ are random variables

2. The data is constant (only a single dataset D)

3. Probability as statement of confidence or belief

4. Parameter **inference**: $P(\theta|D)$, probability that parameters are θ, given observed data D; its uncertainty is given by the full posterior distribution

A glimpse at Bayesian inference

Bayes formula for densities

$$\pi_{\mathsf{POST}}(\theta|data) = \frac{\pi_{\mathsf{LIK}}(data|\theta) \cdot \pi_{\mathsf{PRIOR}}(\theta)}{\int_{\Theta} \pi_{\mathsf{LIK}}(data|\theta) \cdot \pi_{\mathsf{PRIOR}}(\theta)\, d\theta}$$

in case θ is a continuous random vector.

The function $L(\theta) := \pi_{\mathsf{LIK}}(data|\theta)$ is called a likelihood function. The denominator –which is equal to $\pi(data)$– is the expected likelihood and acts as a normalization constant; it ensures that:

$$\int_{\Theta} \pi_{\mathsf{POST}}(\theta|data)\, d\theta = 1$$

Once we have the posterior distribution of the parameters, we can use standard statistical machinery to make probabilistic statements (inference) ...

A glimpse at Bayesian inference

Conjugacy

Definition. Suppose a prior distribution $\pi_{\mathsf{PRIOR}}(\theta)$ belongs to a class of parameterized distributions Π. Then the distribution is said to be **conjugate** wrt a likelihood $\pi_{\mathsf{LIK}}(\cdot|\theta)$ if the posterior distribution $\pi_{\mathsf{POST}}(\theta|\cdot)$ is also in Π.

Remember $\pi_{\mathsf{POST}}(\theta|\cdot) \propto \pi_{\mathsf{LIK}}(\cdot|\theta)\pi_{\mathsf{PRIOR}}(\theta)$.

Examples: Gaussian is conjugate to Gaussian, Beta is conjugate to Binomial.

A glimpse at Bayesian inference

Using the posterior

Once we have the posterior distribution of the parameters, we obtain many interesting information:

- Calculate a credible interval (Bayesian CI)
- Compute the MAP (the value of θ that maximizes the posterior)
- Compute the MMSE (the value of θ that minimizes the MSE)
- Draw observations for θ
- Compute the expected valued of the posterior
- Compute the predictive distribution

In contrast, without the Bayesian machinery we can only compute the ML (the value of θ that maximizes the likelihood)

Example 1: clinical test

- Consider a clinical problem where we need to decide if a patient has a particular medical condition on the basis of an imperfect test
  - Someone with the condition may go undetected (false-negative)
  - Someone free of the condition may yield a positive result (false-positive)
- Nomenclature
  - The true-negative rate P(NEG|¬COND) of a test is called its SPECIFICITY
  - The true-positive rate P(POS|COND) of a test is called its SENSITIVITY
- Problem
  - Assume a population of 10,000 with a 1% prevalence for the condition
  - Assume that we design a test with 98% specificity and 90% sensitivity
  - Assume you take the test, and the result comes out POSITIVE
  - What is the probability that you have the condition?

Example 1

Solution: We take the "has condition" (COND) situation as the positive class to build the confusion matrix (not to be confused with a positive test). The **test** is the data.

$$
\begin{aligned}
P(COND|POS) &= \frac{P(POS|COND)P(COND)}{P(POS)} \\
&= \frac{P(POS|COND)P(COND)}{P(POS|COND)P(COND) + P(POS|\neg COND)P(\neg COND)} \\
&= \frac{0{,}90 \times 0{,}01}{0{,}90 \times 0{,}01 + 0{,}02 \times 0{,}99} \\
&= \frac{5}{16}
\end{aligned}
$$

Example 2: The biased coin

- Tossing a (possibly biased) coin
- The chance of observing k heads and $n-k$ tails in n trials is

$$\binom{n}{k} p^k (1-p)^{n-k}$$

- Here the parameter is $\theta = p$ (probability of getting a head in a single coin toss)
- The numbers k and $n-k$ are the observed data D

Example 2

Solution:

1. Specify your prior!

   a) $P(p = 0{,}6) = 0{,}8$ & $P(p = 0{,}5) = 0{,}2$

   b) $p \sim Unif(0, 1)$

   c) $P(p = 0{,}5) = 1$ & $P(p \neq 0{,}5) = 0$

2. Likelihoods of observing D as $k = 4, n - k = 6$:

   a) $P(D|p = 0{,}6) \approx 0{,}111$

   b) $P(D|p = 0{,}5) \approx 0{,}205$

3. Posteriors ...

Introduction to Machine Learning

Inductive bias

Complete the series! 2, 4, 6, 8, ...

Answer 1: 132 (model 1: $f(n) = n^4 - 10n^3 + 35n^2 - 48n + 24$)

Answer 2: 10 (model 2: $f(n) = 2n$)

How can we rule out the more complex one? (and many others)

1. Supply more "training" data: 2, 4, 6, 8, 10, 12, 14, ...

2. Regularize: add a penalty to higher-order terms

3. Reduce the hypothesis space (e.g. restrict to quadratic models)

So what do we do?

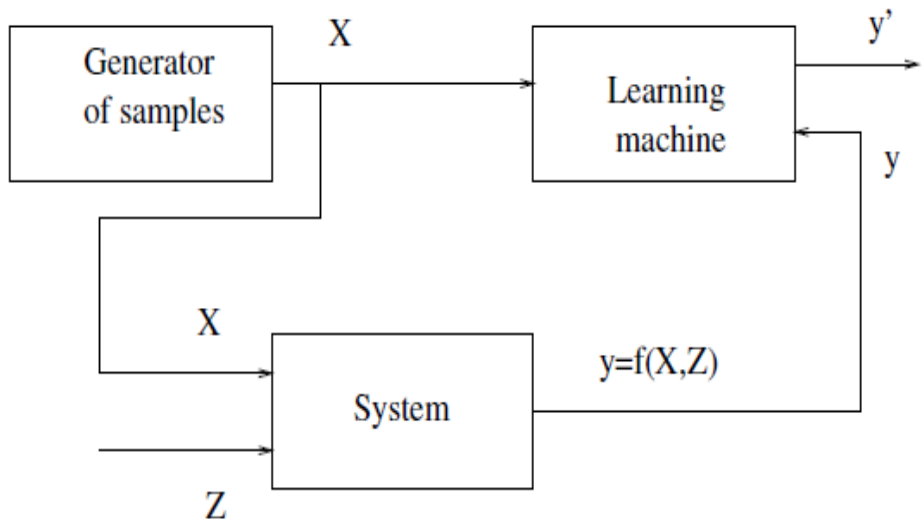
Moral (more formal): Based on training data $\mathcal{D}$ *only*, there is no means of choosing which function f is better (generalization is not "guaranteed")

Consequence: we must add control to the "fitting ability" of our methods (complexity control)

$$\boxed{\text{true error } (f) \leq \text{training error } (f) + \text{complexity of } f}$$

Introduction to Machine Learning

Formulation



`X` are the measured variables

`Z` are the non-measured variables

`y` is the true function

`y'` ($\widehat{y}$) is the modeled function

Introduction to Machine Learning

The Rosetta stone

Machine Learning	Statistics
model	model
parameter/weight	parameter/coefficient
train	fit
learn	infer/estimate
regression	regression
classification	discrimination
clustering	clustering/classification
inputs/features/variables	independent variables explanatory variables predictors
outputs/targets	dependent variables response variables
instances/examples	individuals/observations
error/loss function training/empirical error true/generalization error	fit criterion, deviance resubstitution/in-sample error predictive, out-sample error

Careful with other words: transaction (means observation in DDBB), sample (means dataset in MVS), attribute (means variable in AI), ...

Introduction to Machine Learning

Prediction vs. Inference

Prediction: produce a good estimate for the predicted variable

Inference:

1. Which predictors actually affect the predicted variable?

2. How strong are these dependencies?

3. Are these relationships positive or negative?

Introduction to Machine Learning

Example: Direct mailing

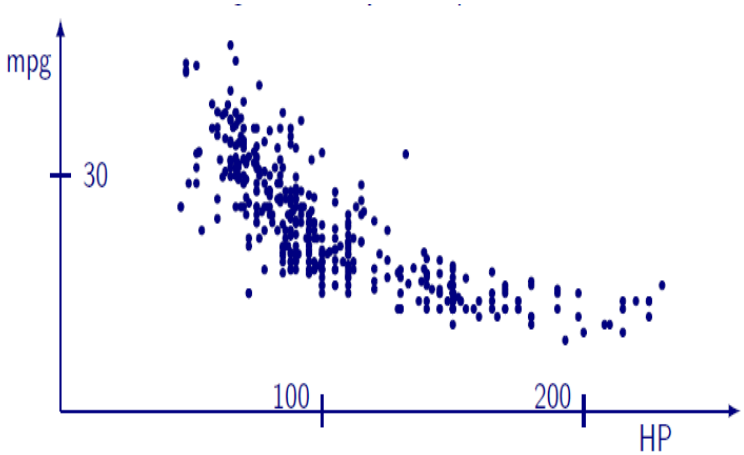
Predicting how much money an individual will donate (the response) based on observations from $90,000$ people on which we have recorded over 400 different characteristics (the predictors)

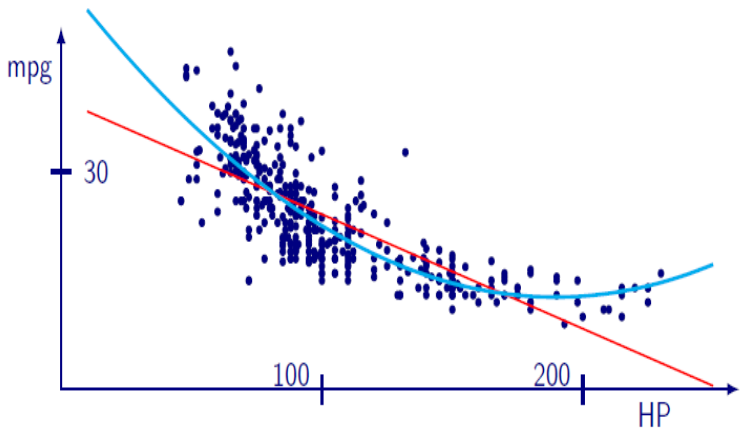
What do we pretend?

1. For a given individual should I send out an e-mail (yes/no)?
2. What is the probability that a specific individual will donate?
3. What is the expected donation for a specific individual?
4. What are the characteristics more strongly linked to donation?
5. How much increase in donation is associated with a given increase in a specific predictor?

Introduction to Machine Learning

The regression task





- Predict some quantitative outcome subject to probabilistic uncertainty
- Example: predict gas mileage (`mpg`) of a car as a function of horsepower (`HP`)

(`auto-mpg` data set from UCI Machine Learning Repository)

Questions related to the regression task

1. How do we express the regression problem as an statistical problem?

2. How do we express the regression problem as an optimization problem?

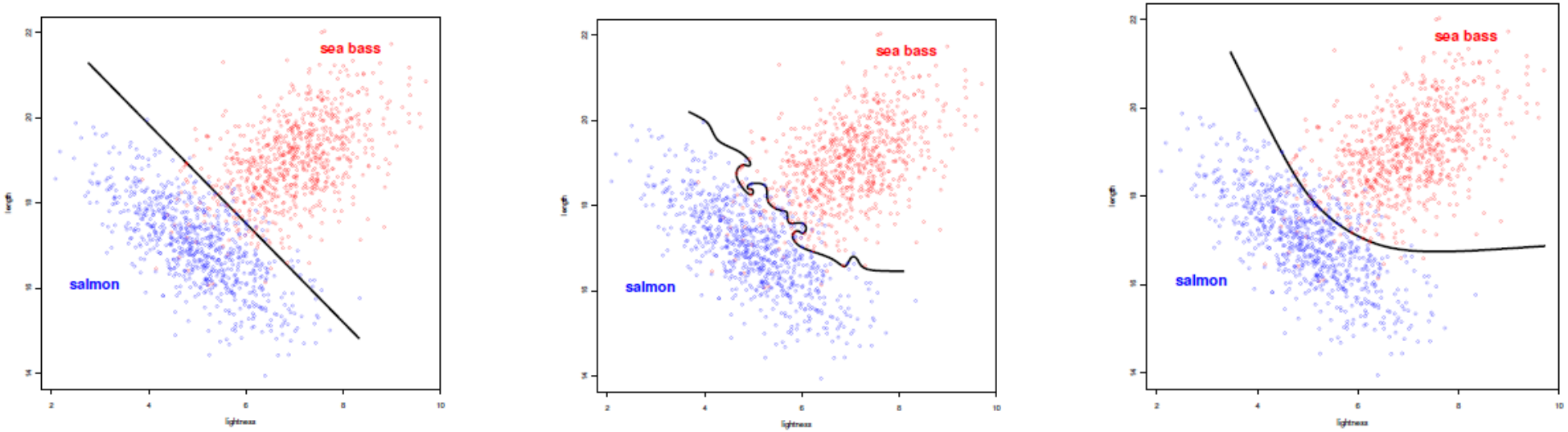
3. Is there always a solution? Is it unique? What is the optimal solution?

4. What is the best achievable error? How do we measure generalization error?

5. How do we express the uncertainty in the solution?

Introduction to Machine Learning

The classification task



The **goal** is to obtain a model based on available **training data** (*known* examples) with high classification accuracy on unseen *unknown* examples (**test data**), i.e. achieving good **generalization**

Questions related to the classification task

1. How do we express the classification problem as an statistical problem?

2. How do we express the classification problem as an optimization problem?

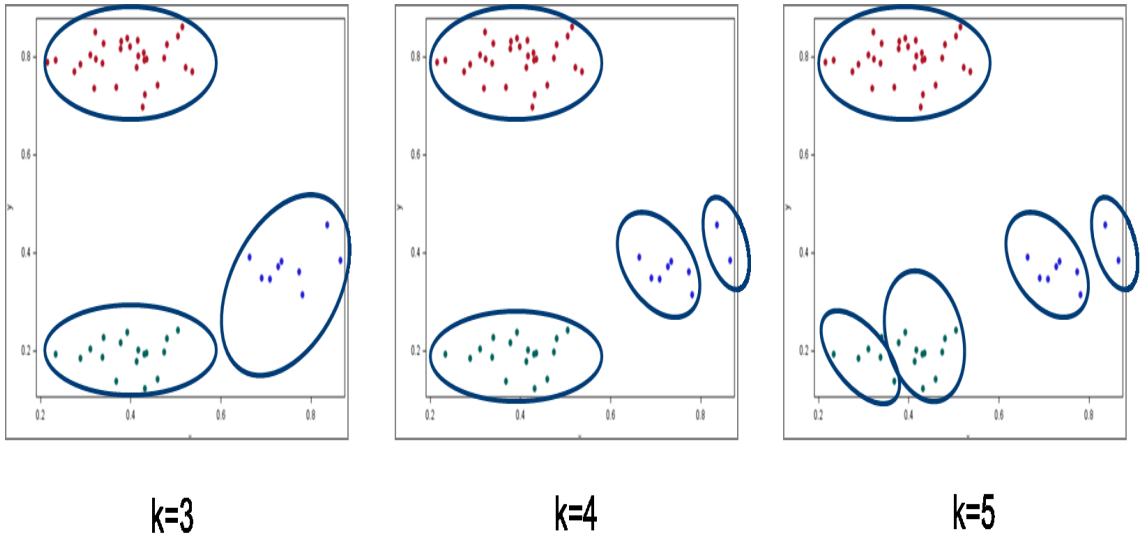
3. Is there always a solution? Is it unique? What is the optimal solution?

4. What is the best achievable error? How do we measure generalization error?

5. How do we express the uncertainty in the solution?

Introduction to Machine Learning

The clustering task



There is a general difficulty with clustering in the large subjectivity of the task.

Questions related to the clustering task

1. How do we estimate the "right" number of clusters? Is there a "right" number of clusters?

2. How do we specify the cluster shapes? Do we have to do so?

3. How do we express the clustering problem as an statistical problem?

4. How do we express the clustering problem as an optimization problem?

5. Is there always a solution? Is it unique? What is the optimal solution?

6. What is the best achievable error? How do we measure generalization error?

7. How do we express the uncertainty in the solution?

Why are these tasks stochastic?

Is there a common reason?

We have a (complete) input data object $(\boldsymbol{x}, \boldsymbol{z})$ and a output data object $\boldsymbol{y}$.

1. The true relation is $f_c : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$, that is $f_c(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{y}$.

2. When we measure data about f_c, we measure only the $\boldsymbol{x}$ portion of the input variables. Therefore, the relation between $\boldsymbol{x}$ and $\boldsymbol{y}$ becomes stochastic.

Setting up the tasks

There are (at least) two ways of setting up these tasks formally:

Optimization view :

$$\min_{\theta \in \Theta} E(\theta) := \frac{1}{n} \sum_{i=1}^{n} l(y_i, f_\theta(x_i)) + \Omega(f_\theta)$$

training error of f_θ + complexity of f_θ
(empirical risk) + (regularizer)

Statistics (both Bayesian and frequentist) view :

Use Bayes formula to compute $P(\theta|data)$ and choose one according to this (posterior) distribution

Many times these two views can yield the same results (which is good!). An example would be LSQ $\equiv$ MaxLik + Gaussian.

Setting up the tasks

The most general description of the data generation mechanism is in terms of the pdf $p(\boldsymbol{x}, y)$ in the joint input-output space $\leftarrow$ the key to generalization!

$$p(\boldsymbol{x}, y) = p(y|\boldsymbol{x}) \cdot p(\boldsymbol{x}), \text{where } p(\boldsymbol{x}) = \int p(y, \boldsymbol{x})\, dy$$

Some techniques use $p(\boldsymbol{x})$, others do not ... the important pdf is $p(y|\boldsymbol{x})$.

What is a ML algorithm/technique?

A ML algorithm gets a dataset D and returns a model of D (a representation of D that either gives structure to D or that allows to make predictions on unseen observations), together with an estimation of the model quality.

The algorithm itself typically determines the model space and the loss function.

Why are linear models so nice?

We will begin our analyses with linear models/techniques

A model is linear when –up to an invertible mapping– it is a linear function of its parameters

1. Analytically tractable: closed-form solutions or fast convergent iterative methods for the solution

2. Unique solution (no local optima)

3. Highly interpretable

4. Amenable to inference (see this same lecture)

5. User-defined fitting ability, via the basis functions

6. Regularization

Why are linear models so nice?

General form of a linear model

A model is linear when –up to an invertible mapping– it is a linear function of its parameters:

$$f(\boldsymbol{x};\theta) = g\left(\theta_0 + \sum_{i=1}^{h} w_i \phi_i(\boldsymbol{x})\right)$$

1. the set of ϕ_i functions are called basis functions (constitute a feature map)
2. g is a strictly monotonic function (in NNs, this is called an activation function)

On data pre-processing

Each problem requires a different approach in what concerns data cleaning and preparation. This pre-process is very important because it can have a deep impact on performance; it can easily take you a significant part of the time.

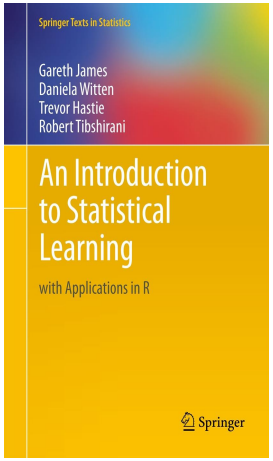
1. treatment of lost values (missing values)
2. treatment of anomalous values (outliers)
3. treatment of incoherent or incorrect values
4. coding of non-continuous or non-ordered variables
5. possible elimination of irrelevant or redundant variables (feature selection)
6. creation of new variables that can be useful (feature extraction)
7. normalization of the variables (*e.g.* standardization)
8. transformation of the variables (*e.g.* correction of serious skewness and/or kurtosis)

Non-standard data (images, audio, text, ...) may need completely *ad hoc* treatments

Recommended reading: introductory

- A free online version of *An Introduction to Statistical Learning, with Applications in R* by James, Witten, Hastie and Tibshirani (Springer, 2013) is available from January 2014.

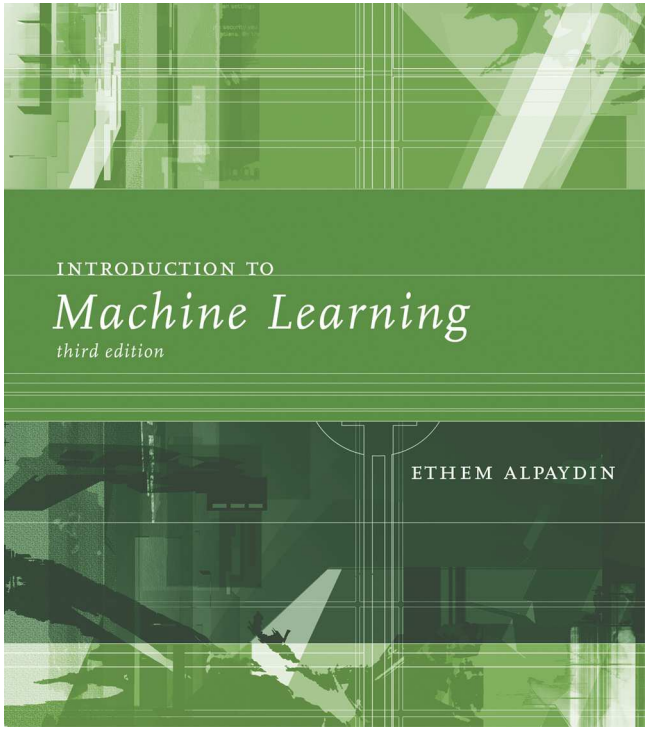
- Springer has agreed to this, so no need to worry about copyright. However, you may not distribute printed versions of this pdf file.



`http://www-bcf.usc.edu/~gareth/ISL/`

Recommended reading: intermediate

- *Introduction to Machine Learning (3rd Ed.)*, by E. Alpaydin (The MIT Press, 2009)
- There are several editions (the latest, the better)



https://mitpress.mit.edu/books/introduction-machine-learning-0

Recommended reading: standard level

Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006
`http://research.microsoft.com/~cmbishop/PRML`

Pattern Classification (2nd Ed.), Richard 0. Duda and Peter E. Hart and David G. Stork, Wiley-Interscience, 2001.
`http://rii.ricoh.com/~stork/DHS.html`

The Elements of Statistical Learning (10th edition) Hastie, Tibshirani and Friedman (2009). Springer-Verlag. `http://statweb.stanford.edu/~tibs/ElemStatLearn/`

Learning from data: concepts, theory, and methods (2nd Ed.). Cherkassky, V.S., Mulier, F. John Wiley, 2007.

Machine Learning: A Probabilistic Perspective. Kevin P. Murphy. MIT Press, 2012.
`https://www.cs.ubc.ca/~murphyk/MLbook/`

On the best programming environment

Python, R, MATLAB/Octave, Java, SQL, C/C++, Julia, ...

Programming Langauge	Standouts	Setbacks	Key Libraries	Execution Speed	Learning curve	Data Analytics Capabilities	Graphical Capabilities	Tools (IDEs, Plugins, etc)	Community Support	Integration with External apps	Job market	Score
R	Open source; Good for statistical analysis and data processing; Huge collection of algorithms available as packages; Visualization support;	Steep learning curve; obscure commands	gbm, RTextTools, dplyr, zoo, ggplot2, caret	3	1	5	4	4	4	3	4	28
Python	Open source; Easy to learn; All the benefits of general-purpose programming language; Big Data ready;	Speed of execution; needs to handle library dependancies if migrated from 2.x to 3.x	Scikit-learn, Pandas, matplotlib, NumPy, SciPy, theano, nltk	4	4	3	3	3	5	5	5	32
MATLAB	Good with mathematical processes, complex matrix operations; Broad range of machine learning, signal processing and image processing libraries as toolboxes;	Proprietary; Lack of good open source ecosystem; Difficulty when the data can't be represented in matrices	Statistics and Machine Learning; Image Processing; Signal Processing; Optimizatio n; Wavelet;	2	3	4	4	5	3	2	2	25
Octave	Open source; Good for numerical computations; Built with MATLAB compatibility; Known as Clone of Matlab; Good for building preliminary models;	Lack of interoperability with external data sources - csv, databases, etc	libsvm, shogun, liblinear, ltfat, vlfeat	2	2	3	2	2	3	2	1	17
Julia	Open source; Desinged to handle numerial and scientific computing; Good performance;Ability to call C, Python functions;	Relatively new programming language; doesn't have much to offer in the way of extensive libraries	MLBase, MLUtils, MLKernels, Clustering, MachineLear ning	5	3	4	2	2	2	3	1	22

Making the best out of R

- R is an open-source software for statistical computing, data analysis and publication-quality graphics and a very usable programming language (mix of imperative, OO and functional)

  Get R from `http://cran.r-project.org/`

- RStudio is a friendly IDE for R (Windows, Mac, and Linux)

  Get RStudio from `http://www.rstudio.com/`

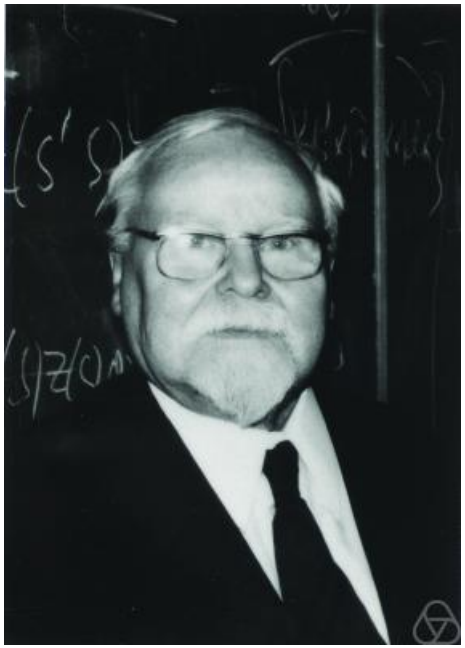
- R has a very active community and dozens of very useful packages:

  `https://cran.r-project.org/web/views/MachineLearning.html`
  `https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-pac`
  `https://awesome-r.com/#awesome-r-machine-learning`
  `https://github.com/ujjwalkarn/DataScienceR`
  `https://www.tidyverse.org/`

Relevant characters for this lecture

Thomas Bayes
(1702-1761)

Pierre Simon Laplace
(1749-1827)



Andrey N. Tikhonov
(1906-1993)