Introduction to machine learning
Principles of Machine Learning

Mathilde Mougeot

ensIIE & ENS Paris-Saclay, France

2025

Principles of Machine Learning

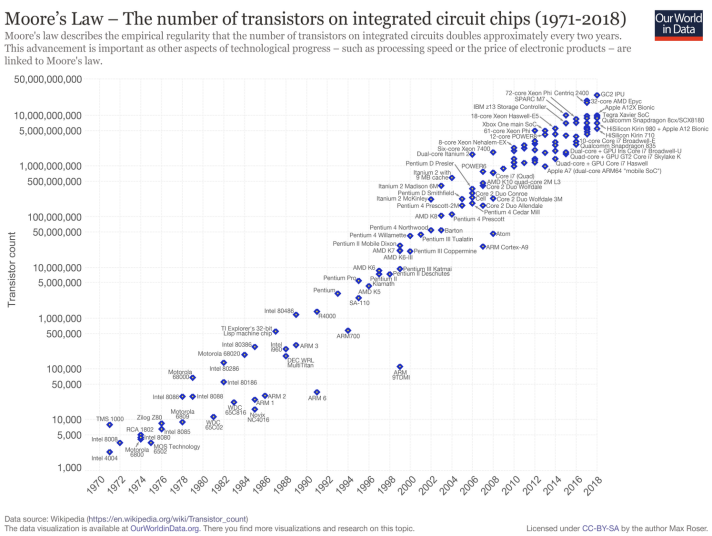
Introduction

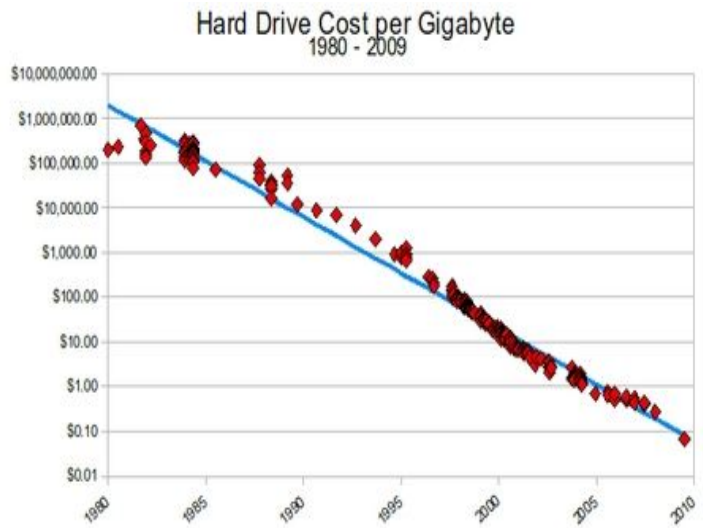
Introduction

- Machine Learning (ML), data science, and statistics are fields that describe how to learn from, and make *predictions* about data.
- The availability of big datasets is a hallmark of modern science where data analysis has become an important component of many areas.
- *ML is very broad and interdisciplinary*, drawing on ideas and intuitions from many fields including statistics, computational neuroscience, and physics.

Why study Machine Learning ?

- This *big data* revolution has been spurred by an exponential increase in computing power and memory commonly known as Moore's law.
- This increase in our computational ability has been accompanied by new techniques for analyzing and learning from large datasets.
- Data scientists and ML engineers in industry use concepts and tools developed for ML to gain insight from large datasets.





What is Machine Learning ?

- Classical statistics is primarily concerned with how to use data to **estimate** the value of an unknown quantity.
- Machine Learning is a subfield of artificial intelligence with the goal of developing algorithms capable of learning from data automatically.
- Therefore, techniques in **ML tend to be more focused on prediction** rather than estimation.
- Methods from ML tend to be applied to more complex high-dimensional problems than those typically encountered in a classical statistics course

What is Machine Learning ?

- In two words...
  - x observable quantity related to some parameter θ of a model $p(X|\theta)$
  - A dataset $\mathcal{D}$ is built from observations from experiment(s) ($X_{\mathcal{D}}$ matrix)
  - Data are used to fit the model, to compute the estimated parameter $\hat{\theta}$, that provides the best explanation of the data.

$$\hat{\theta} = \arg\max_{\theta}\{p(X_{\mathcal{D}}|\theta)\}$$

- **Estimation problems** are concerned with the accuracy of $\hat{\theta}$ whereas **Prediction problems** are concerned with the ability of the model to predict new observations i.e., the accuracy of $p(X|\hat{\theta})$.
- Although the goals of estimation and prediction are related, they often lead to different approaches

Machine learning issues

ML can be divided into three broad categories :

- **Supervised learning** concerns learning from *labeled data*. Common supervised learning tasks include classification and regression.
- **Unsupervised learning** is concerned with *finding patterns and structure* in unlabeled data. Examples of unsupervised learning include clustering, dimensionality reduction, and generative modeling.
- **Reinforcement learning** concerns agents which learn by interacting with an environment and changing its behavior to maximize its reward.

Ingredients of a Supervised machine learning problem.

1. The dataset $\mathcal{D} = \{(x_i, y_i),\ 1 \leq i \leq n\} = (X, y)$.
   X is a matrix of independent variables and y is a vector of dependent variables.

2. The model $f(x, \theta)$.
   Function : $f : x \to f(x) = y$ of the parameters θ.
   f is used to predict an **output** from a vector of **input** variables.

3. The cost function $C(y, f(x, \theta))$.
   Allows us to quantify how well the model performs on the observations y.

4. Learning the model means finding the value of θ that minimizes the cost function.

   A commonly used cost function is the squared error. Minimizing the squared error cost function is known as the method of least squares, and is typically appropriate for experiments with Gaussian measurement errors.

Machine Learning Recipes

ML researchers and data scientists follow standard recipes to obtain models that are useful for prediction problems.

1. Randomly divides (at least) the dataset $\mathcal{D}$ into two mutually exclusive groups $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ called the training and test sets .
2. Training. The model is fit by minimizing the cost function using only the data in the training set $\hat{\theta} = \arg\min_\theta C(y_{\text{train}}; f(X_{\text{train}}, \theta))$.
3. Predictive power. The performance of the model is evaluated by computing the cost function using the test set $C(y_{\text{test}}; f(X_{\text{test}}, \theta))$

Machine Learning Recipes

- The Train and the Test errors :
  In-sample error $E_{\text{in}} = C(y_{\text{train}}; f(X_{\text{train}}, \theta))$.
  Out-sample error $E_{\text{out}} = C(y_{\text{test}}; f(X_{\text{test}}, \theta))$.
  We often have $E_{\text{out}} \geq E_{\text{in}}$

- Cross-validation Splitting the data into mutually exclusive training and test sets provides an unbiased estimate for the predictive performance of the model.

- Several candidates ML models need to be compared because ML problems involve inference about complex systems where the exact form of the mathematical model that describes the system is unknown.

- Model selection The model that minimizes this out-of-sample error E_{out} is chosen as the best model.

Choosing a ML model

The goal of Machine Learning is to obtain a model that is useful for prediction not a model that provides the best explanation for the current observations.

- The discrepancy between E_{in} and E_{out} becomes more and more important, as the complexity of our data, and the models used to make predictions, grows.
- As the number of parameters in the model increases, ML works in high-dimensional spaces. The *curse of dimensionality* ensures that many phenomena that are absent or rare in low-dimensional spaces become generic.
- The nature of distance changes in high dimensions.
- The ability to predict depends on the number of data points, the noise, and the prior knowledge of the system.

The modeling issues : the framework

- We consider a probabilistic process that assigns :
  y_i to an observation x_i.
- The data are generated by drawing samples from the equation :

$$y_i = f(x_i) + \epsilon_i$$

$f(x_i)$ is some fixed but possibly unknown function,
$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, Gaussian, uncorrelated noise variable,
ϵ_i, are iid, $1 \leq i \leq n$.

The modeling issues : ML models.

- $f_\alpha(x, \theta_\alpha)$ a family of functions :
  α represents *the model class* used to model the data and make prediction without knowing the function $f(x)$.

- Illustration with the polynomial regression families of order p :
  - ML Model : $f_{\theta_p(x)} = \theta + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p$
    $x \to z = (x, x^2, x^3, \ldots, x^p)$,
    z provides a multivariate transformation of the initial x.
    z is usually denoted by **features** (feature vector of size p).
    Different p parameters provide *different complexities*.
- DataSet of observations : $\mathcal{D}_n = \{(x_i, y_i),\ 1 \leq i \leq n,\ x_i \in \mathbb{R},\ y_i \in \mathbb{R}\}$.

- **Learning step.** optimization procedure
  $\hat{\theta} = \arg\min_{\theta_p} E(\mathcal{D}_n, f_{\theta_p}) = \sum_i (y_i - f_{\theta_p}(x_i))^2$

- **Predictive power.** The effectiveness of each model is evaluated on a different dataset, the test dataset.

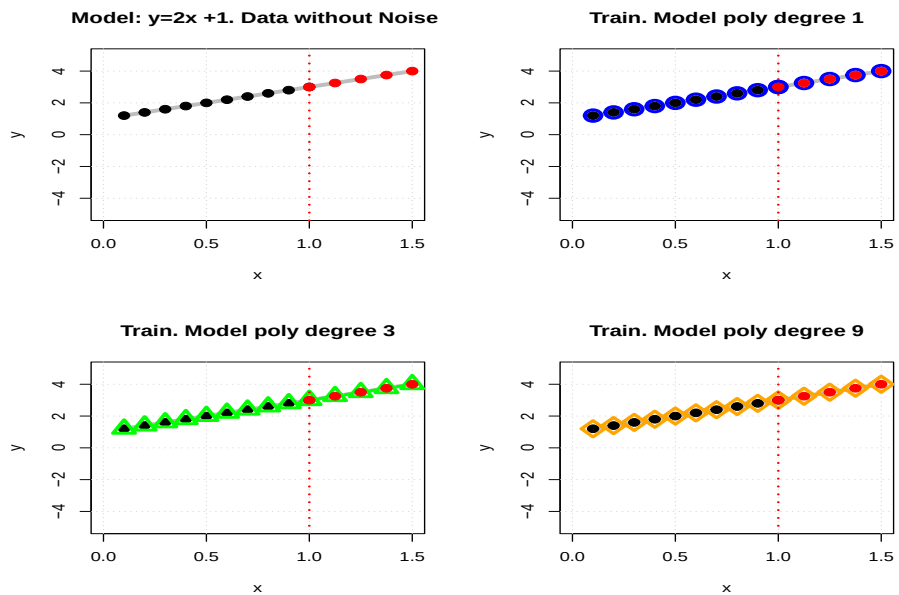
The modeling issues : ML models.

Impact of the choice of the ML model (here polynomial).

Unknown function : $y = 2x + 1$,

3 ML models $x \to z = (1, x, x^2, x^3, \ldots, x^p)$ with $p = 1$, $p = 3$, $p = 9$

Training 10 obs. with no noise (black points). Test data set (5 red points).



$\to$ With no noise, all the ML models perform well on the Test data.

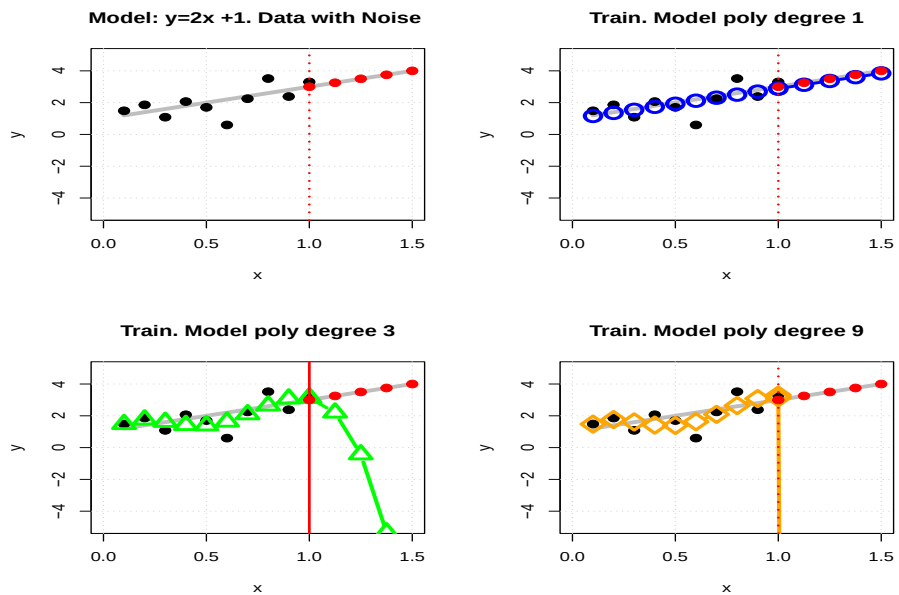
The modeling issues : ML models.

Impact of the choice of the ML model and the training data set.

Unknown function : $y = 2x + 1$,

3 ML models $x \to z = (1, x, x^2, x^3, \ldots, x^p)$ with $p = 1$, $p = 3$, $p = 9$

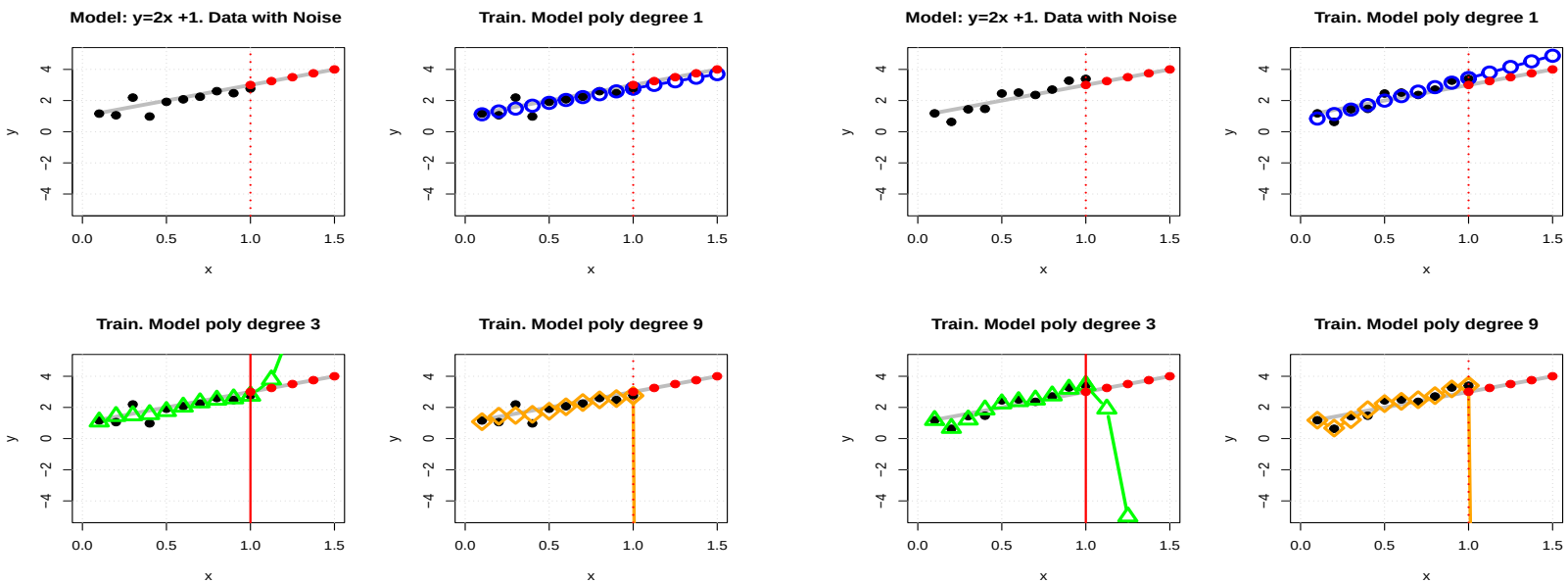
Training 10 obs. with noise (black points). Test data set (5 red points).



$\to$ With noise in the training, the less complex/ contrained ML model performs much better for new data (outside the learning domain).

The modeling issues : ML models.

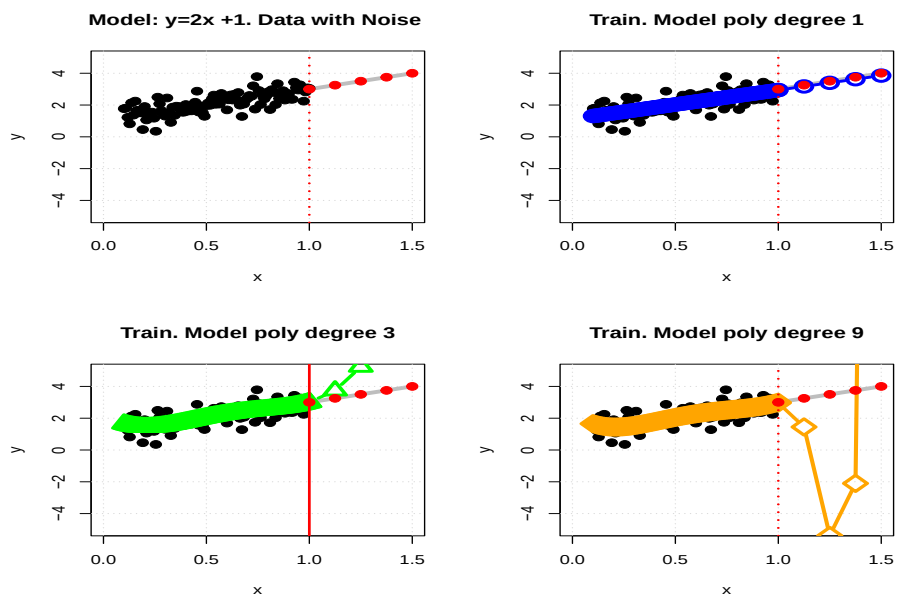
Impact of the choice of the ML model and the training data set.
Unknown function : $y = 2x + 1$, Training 10 obs. with noise (black points).
3 ML models $x \to z = (1, x, x^2, x^3, \ldots, x^p)$ with $p = 1$, $p = 3$, $p = 9$
Test data set (5 red points).



$\to$ With noise in the training, the less complex/ contrained ML model performs much better for new data (outside the learning domain).

The modeling issues : ML models.

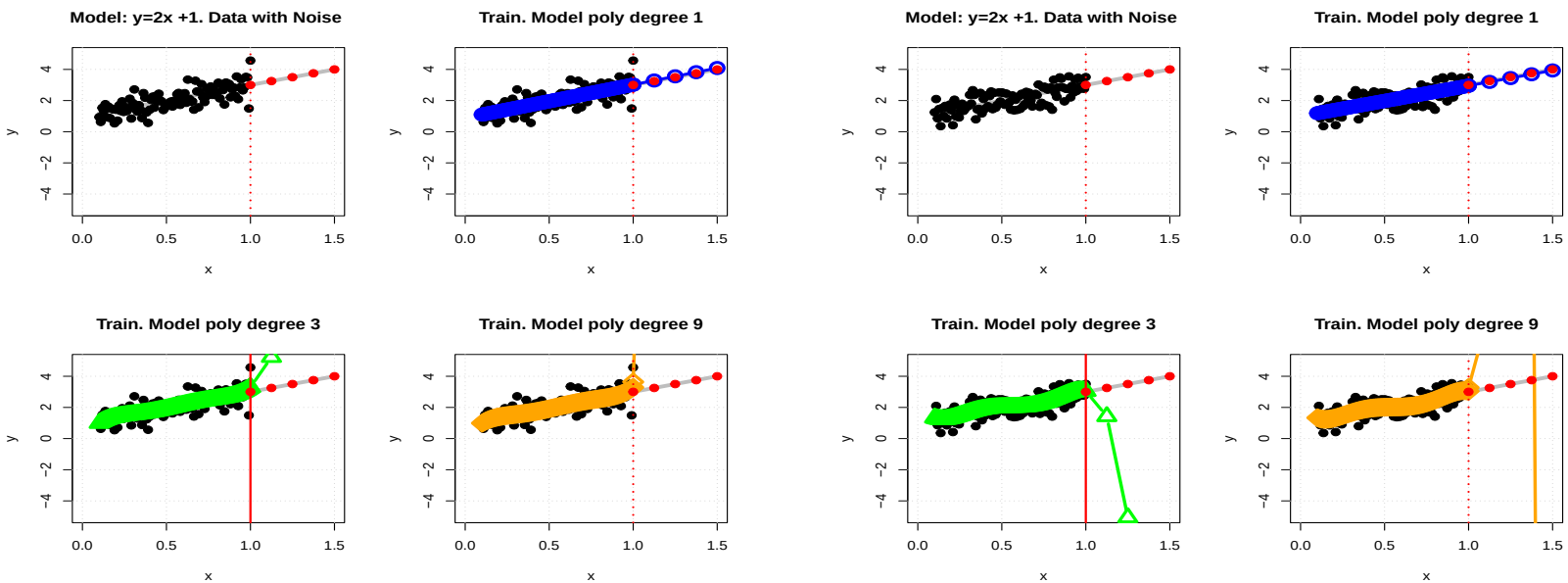
Impact of the choice of the ML model, the size of the training data set.
Unknown function : $y = 2x + 1$, Training 100 observations with noise (black points).
3 ML models $x \to z = (1, x, x^2, x^3, \ldots, x^p)$ with $p = 1$, $p = 3$, $p = 9$
Test data set (5 red points).



$\to$ With more data in the training set, the parameters of the correct model are better estimated (less variations for various training data set)

The modeling issues : ML models.

Impact of the choice of the ML model, the size of the training data set.
Unknown function : $y = 2x + 1$, Training 100 obs. with noise (black points).
3 ML models $x \to z = (1, x, x^2, x^3, \ldots, x^p)$ with $p = 1$, $p = 3$, $p = 9$
Test data set (5 red points).



$\to$ With more data in the training set, the parameters of the correct model are better estimated (less variations for various training data set)

The modeling issues : ML models.

- At "small" sample sizes, noise can create fluctuations in the data that look like genuine patterns.
- Simple models (like a linear function) cannot represent complicated patterns in the data, so they are forced to ignore the fluctuations and to focus on the larger trends.
- Complex models with many parameters can capture both the global trends and noise-generated patterns at the same time. and can be tricked into thinking that the noise encodes real information.
- This problem is called **overfitting** and leads to drop-off predictive performance.

Universal messages

- Fitting is not predicting. Fitting existing data well is fundamentally different from making predictions about new data.
- Using a complex model can result in overfitting. Overfitting degrade the predictive performance of the model, the ability to predict on new data.
- Simple models can be better at prediction than complex models due to the bias-variance tradeoff. It takes less data to train a simple model than a complex one.
- Interpolation vs extrapolation. It is difficult to generalize beyond the situations encountered in the training data set

An Introduction to Statistical learning theory

Data, model and complexity

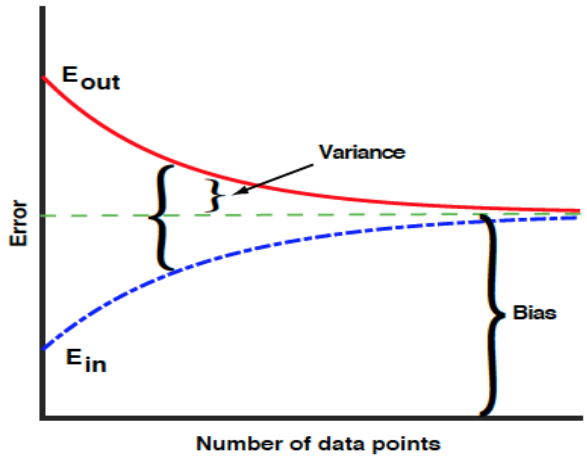
Considering :

- an unknown function $y = f(x)$
- A hypothesis set $\mathcal{H}$ consisting of all considering functions in the domain of f. $\mathcal{H}$ may be infinite.
  This important choice usually depends on the intuition about the problem of interest.
- A data set of supervised observations : $\mathcal{D}_n = \{(x_i, y_i)\ 1 \leq i \leq n\}$

The goal of ML is

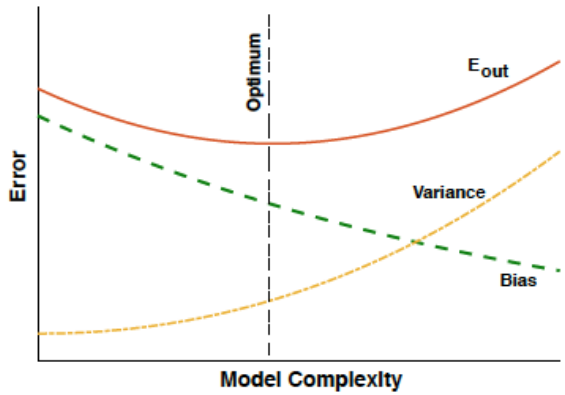
- to select a function from the hypothesis set $\mathcal{H}$ that approximates $f(x)$ as best as possible.
- to find $h \in \mathcal{H}$, such that $h \simeq f$ in some mathematical sense
- with a finite set of observations...

Impact of the size of the data set.



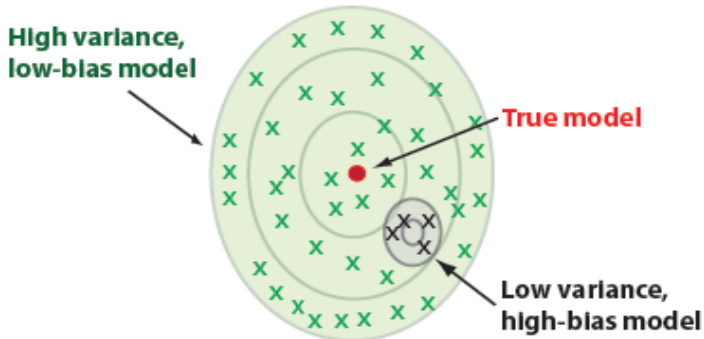
- The bias represents the best the ML model could do if we had an infinite amount of training data.
- In the infinite data limit the in-sample and out-of-sample errors must approach the same value, which is called the bias of the model.

Impact of model complexity.



- The bias is a property of the kind of functions, or model class, we are using to approximate $f(x)$.
- The more complex the model is, the smaller is the bias.
- To get the best predictive power, one should minimize the out-of sample error, E_{out}, rather than the bias.

Impact of biais and variance



- More complex models need a larger amount of training data
- Because of noise, the fluctuations in the learned models (variance) will be much larger for the more complex model than simpler models.
- Depending on the amount of training data, it may be more favorable to use a less complex, high-bias model to make predictions.

The biais-Variance decomposition

The central principal of ML, the

biais and variance trade-off.

- More complex models need a larger amount of training data
- Because of noise, the fluctuations in the learned models (variance) will be much larger for the more complex model than simpler models.
- Depending on the amount of training data, it may be more favorable to use a less complex, high-bias model to make predictions.

Bias-Variance Dilemma

- $\mathcal{H} = \{\text{measurable functions } \mathcal{X} \to \mathcal{Y}\}$, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$
- **Best solution** $f^* = \arg\min_{f \in \mathcal{H}} \mathcal{R}(f)$ with $\mathcal{R}(f) = \mathbb{E}_{\mathcal{X}}[f - f^*]^2$
- Class $h \subset \mathcal{H}$ of functions
- **Ideal target in h.** $f_h^* = \arg\min_{g \in h} \mathcal{R}(g)$ with $\mathcal{R}(g) = \mathbb{E}_{\mathcal{X}}[g - f_h^*]^2$
- **Estimate in h.** $\hat{f}_h$ obtained with some procedure thanks to bf data

Approximation error and estimation error (Bias/Variance)

$$\mathcal{R}(\hat{f}_h - f^*) = \mathcal{R}(f_h^* - f^*) + \mathcal{R}(\hat{f}_h - f_h^*) + \sigma^2$$

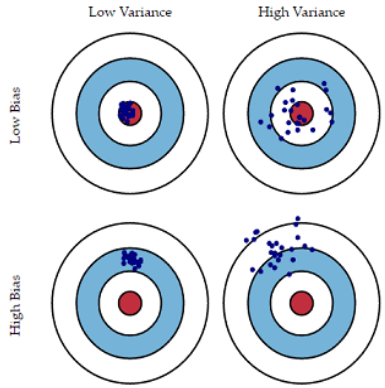
- $\mathcal{R}(f_h^* - f^*)$: Approximation error : large if the model h is not well chosen
- $\mathcal{R}(\hat{f}_h - f_h^*)$: Estimation error : large if the model is complex !

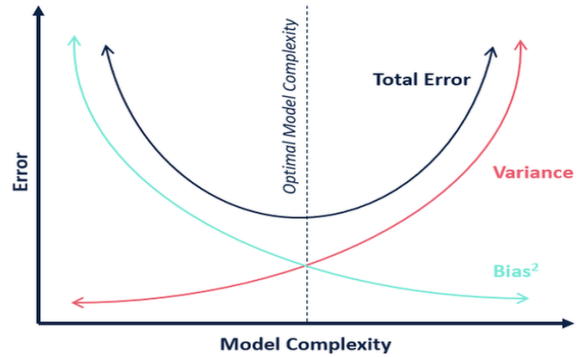
Bias-Variance Dilemma

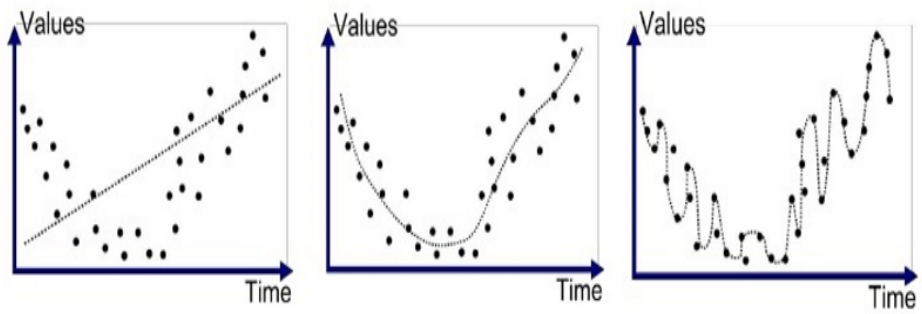
$$\mathbb{E}[(y - \hat{f}_h(x))^2] = \text{Bias}[\hat{f}_h(x)]^2 + \text{Var}[\hat{f}_h(x)] + \sigma^2$$

- $Bias[\hat{f}_h(x)]^2 = \mathbb{E}[\hat{f}_h - f(x)]^2$
- $Variance[\hat{f}(x)]^2 = \mathbb{E}[(\hat{f}_h(x) - \mathbb{E}[\hat{f}_h(x)])^2]$
- σ^2 variance of the noise

Impact of the choice of a model

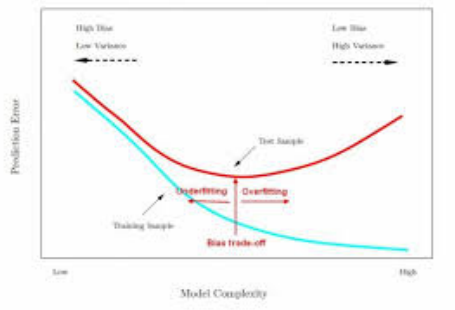






Under-Fitting vs Over-fitting Issue

Different behaviors for different model complexity.



- Low complexity model are easily learned but approximation may remain large (Under-fit).
- High complexity model may contains a good ideal target but the one learned can be bad due to a high variance (Over-fit).

Bias-Variance trade-off $\iff$ avoid overfitting and underfitting

Empirical Risk minimization

In practice, one replaces the minimization of the average loss by the minimization of the empirical loss.

- Empirical Risk

$$\mathcal{R}_n(g) = \frac{1}{n} \sum_{i=1}^{i=n} \ell(Y_i, g(x_i))$$

- Empirical Risk minimizer over a model $h \in \mathcal{H}$

$$\hat{g}_h = \arg\min_{g \in h} \{\mathcal{R}_n(g)\}$$

Cross-validation

- Generalization is the goal of supervised learning
- A trained classifier has to be generalizable. It must be able to work on other data than the training dataset
- Generalizable means "works without over fitting"
- This can be achieved using cross-validation
- There is no machine learning without cross-validation at some point !
- In the case of penalization, we need to choose a penalization parameter C that generalizes.

Cross-validation

- Cross-validation :$\mathcal{D}_n = \mathcal{D}_{\text{Train}} \oplus \mathcal{D}_{\text{Test}}$
  - $\mathcal{D}_{\text{Train}}$ calibration of the parameters of model (model selection)
  - $\mathcal{D}_{\text{Test}}$ Performance evaluation

  Remark :
  $\rightarrow$ possible bias for a given Train or Test set on the performances, depending on the chosen data.
  $\rightarrow$ The data are often chosen at random.

- K-Fold cross validation

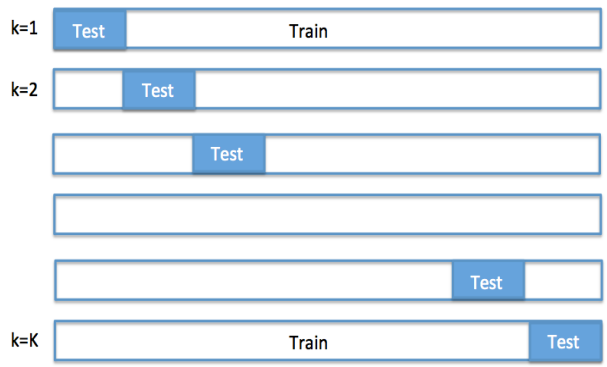
- Leave One Out
  $K = n$, n number of available observations (small data set !)

Kfold cross validation.

Kfold : K repetitions with K different data sub sets for Train and Test procedures. $\mathcal{D}_n = \{(x_i, y_i)\ i = 1, \ldots, n\ , y_i \in \mathcal{X},\ x_i \in \mathbb{R}^p\}$

For k, $1 \leq k \leq K$:

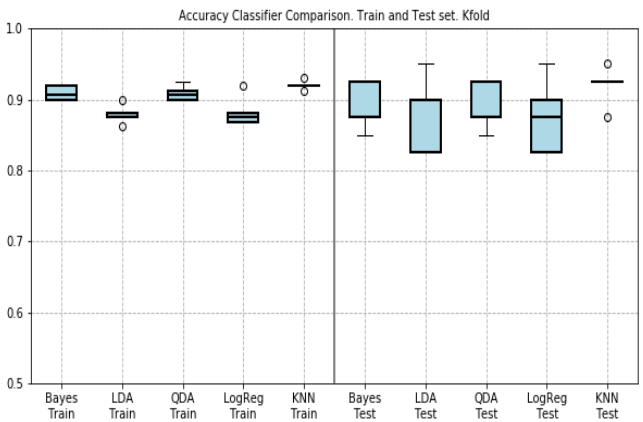
- $\mathcal{D}_n = \mathcal{D}_{\mathrm{T}rain_k} \oplus \mathcal{D}_{\mathrm{T}est_k}$
- $\mathcal{D}_{\mathrm{T}est_k} = \mathcal{D}_n \ominus \mathcal{D}_{\mathrm{T}rain_k}$

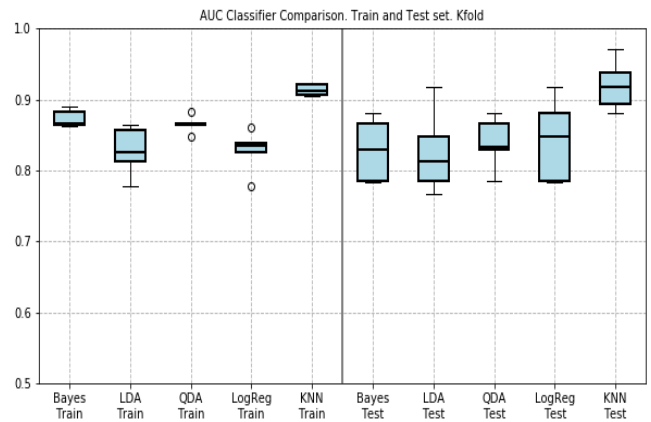


- The Train and Test performances are studied over the K repetitions (boxplot)
- The model which minimized the Test averaged performance is *in general* selected.

Kfold cross validation

Kfold (cross validation) Performances for several models computed for Train and Test data bases.





Curse of high dimension

Distance between observations

–	X^1	X^2	...	X^j	...	X^d
1	x_{11}		...	x_{1j}		x_{1d}
2						
...						
$\rightarrow i$	x_{i1}		...	x_{ij}		x_{id}
...						
n	x_{n1}		...	x_{nj}		x_{nd}

- For two observations (x_i, x_k), $x_i \in \mathbb{R}^d$, $x_k \in \mathbb{R}^d$

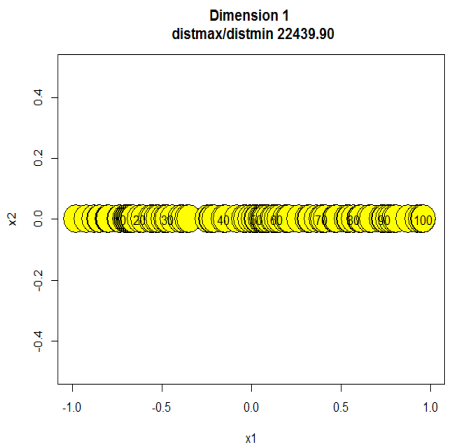
$\rightarrow$ Euclidian distance ℓ_2 between two observations

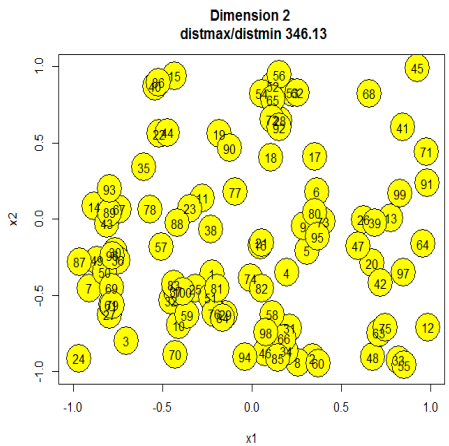
$$||x_i - x_k||_{\ell_2} = \sqrt{\sum_{j=1}^{d} (x_i(j) - x_k(j))^2}$$

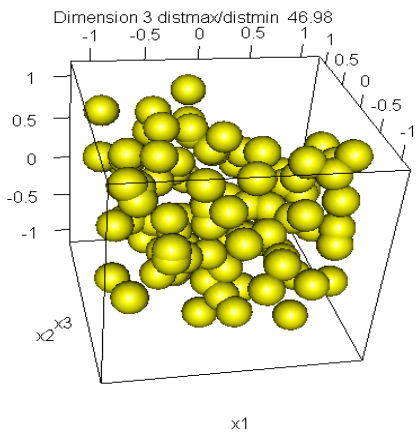
Dimension curse

- Evaluation of the distance between two observations in dimension d
- Illustrations : $n = 100$ observations uniformly distributed, 1, 2, 3, ...
- Indicator : $\frac{max_{i \neq j} ||x_i - x_k||_{\ell_2}}{min_{i \neq j} ||x_i - x_k||_{\ell_2}}$

$d = 1$ | $d = 2$ | $d = 3$





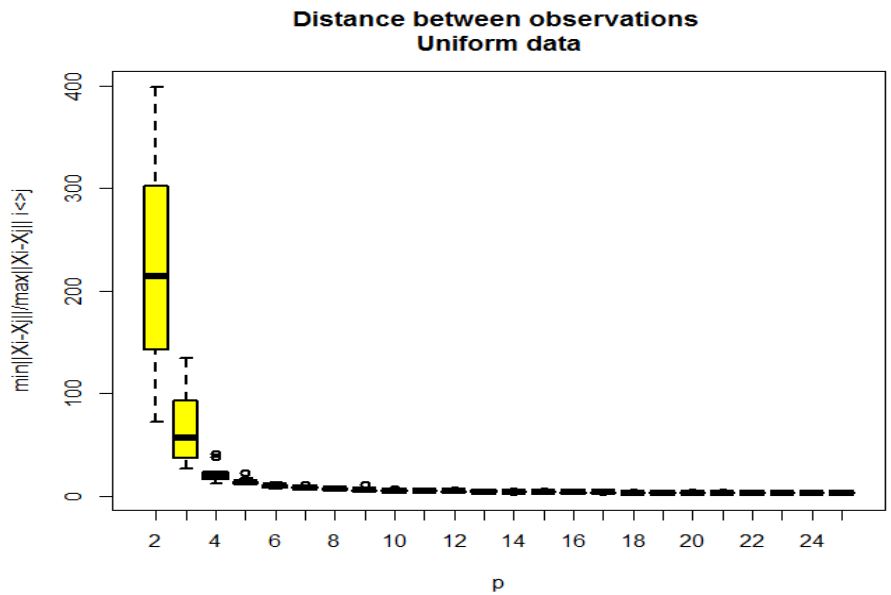


22 435 | 346 | 47

Dimension curse

Ratio study $\frac{max_{i \neq j} ||x_i - x_j||_{\ell_2}}{min_{i \neq j} ||x_i - x_j||_{\ell_2}}$ function of the dimension d

Illustration : $n = 100$ observations uniformly distributed ($K = 100$ repetitions)



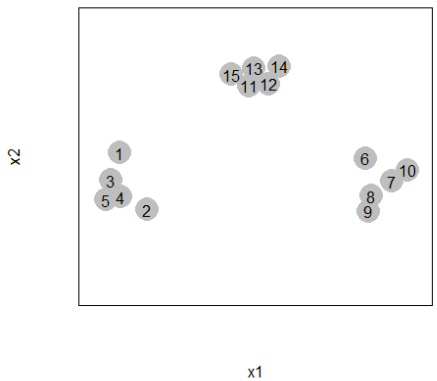
$\rightarrow$ The value of the ratio tends to ~ 1 when d increases.

$\rightarrow$ The euclidian distance loses its discrimination ability in high dimension

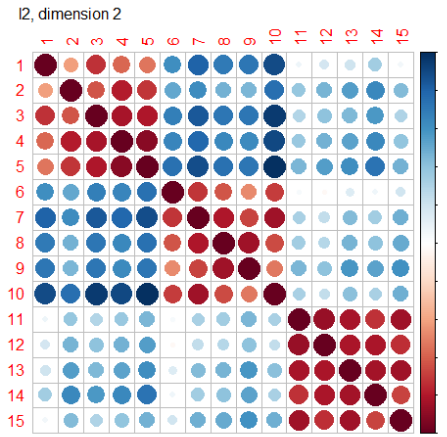
$\rightarrow$ Serious problem especially for segmentation tasks...

Data segmentation (d=2)

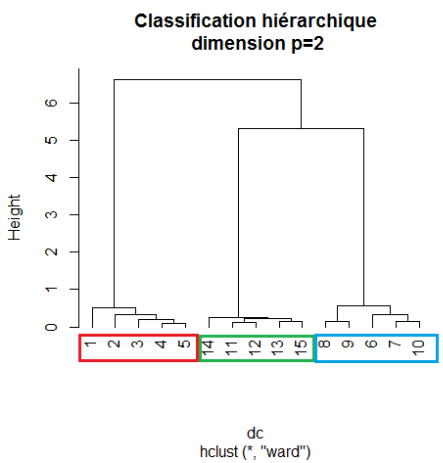
Observations



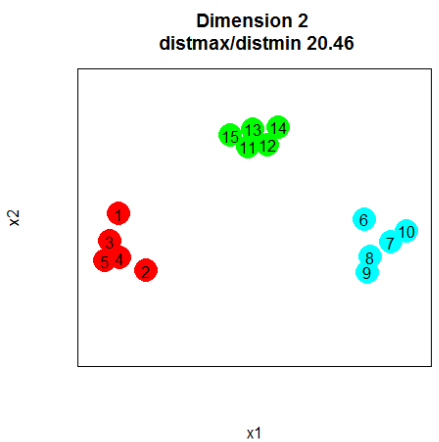
distance matrix



HAC



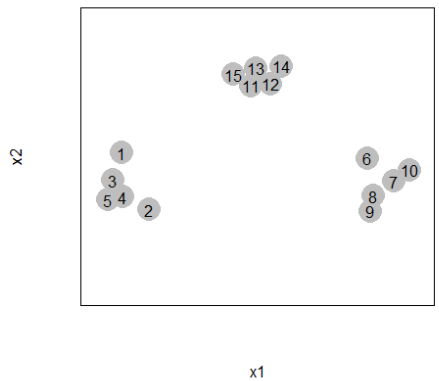
3 classes Clustering



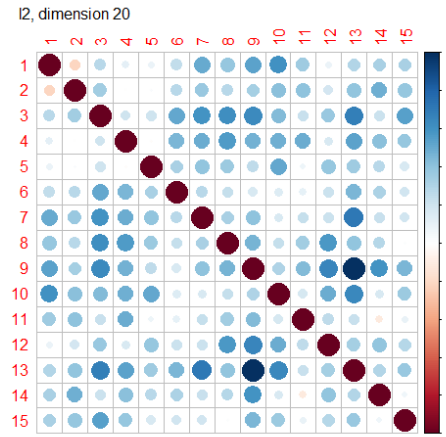
Data segmentation (d=20)

data are embedded in a high dimensional space $d = 20 = 2 + 18$

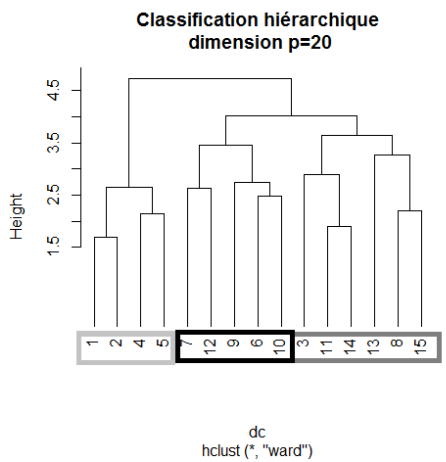
Observations



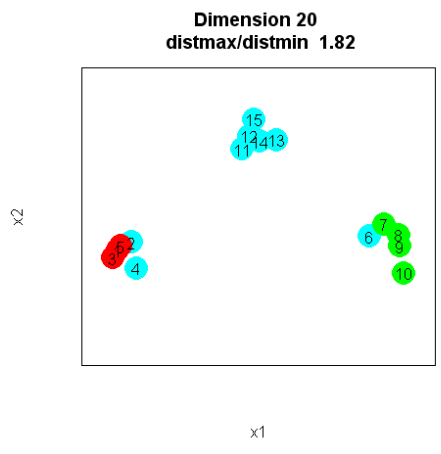
distance matrix



HAC



3 classe Clustering



Dimension reduction

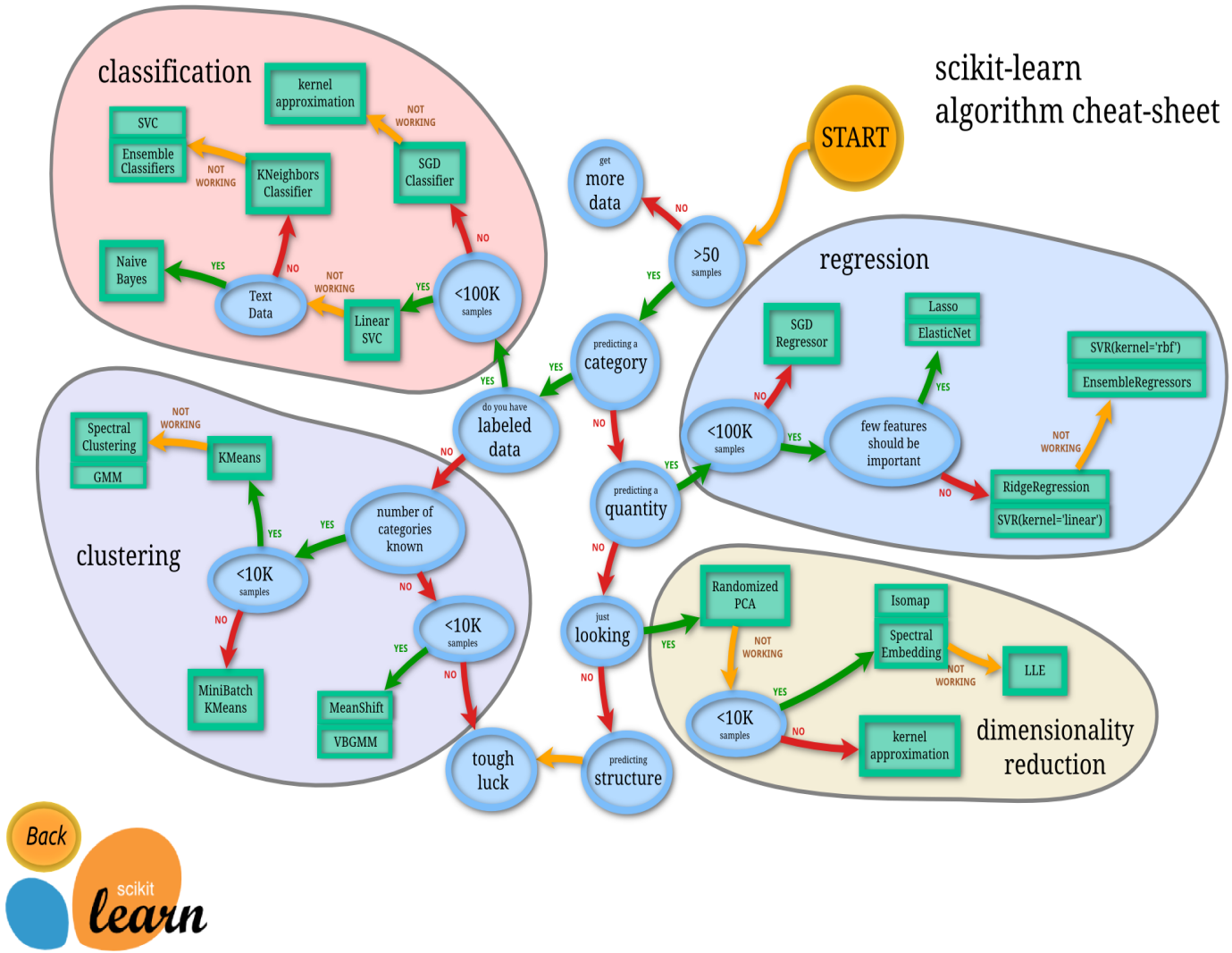
Find good representations of the data initially coded in large dimensions

- Features : a small number of discriminant features
  based on data expertise or automatic extraction.
- Compress Sensing : sparse representation (S) of x based on a linear combinaison of p vectors.
- Manifold estimation : x is represented in a low-dimensional space using the Laplacian eigenvectors on the variety, estimated from a graph of neighborhoods using the examples

$\rightarrow$ Mathematical tools at the interface of harmonic analysis, geometry, probability and statistics.

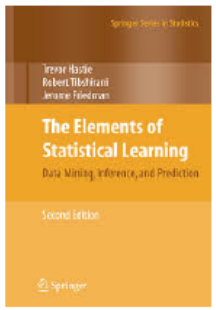
Machine learning models

Some Machine learning models...



References

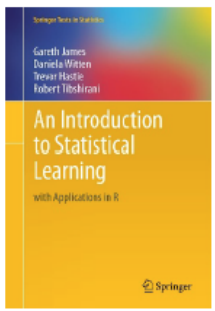
References



.The Elements of Statistical Learning Theory

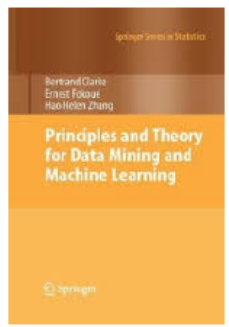
Par Hastie, Tibshirani, Friedman

http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html



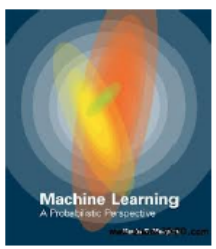
.An Introduction to Statistical Learning with Applications in R

Par Witten, Hastie, Tibshirani



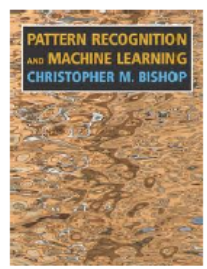
.Principles and Theory for Data Mining and Machine Learning

Par Clarke, Fokoué, Zhang



.Machine Learning: A Probabilistic Perspective

Par K. Murphy



.Pattern Recognition and Machine Learning

Par C. Bishop