Introduction to machine learning

With a focus on RS data

Laetitia Chapel[1]

Master CDE, 2020

[1]laetitia.chapel@irisa.fr

Syllabus

Machine learning module

Learning outcomes

- Understand the different ML problems and methods
- Design, for a given data analytic problem, the appropriate solution to be used
- Implement deep learning models within a standard framework

Module contents

- Principles of supervised learning and other ML paradigms
- Classification and regression
- Dimension reduction and feature selection
- Anomaly detection
- Training strategies and evaluation protocols
- Use of software libraries

Outline

Syllabus

What is machine learning?

A classification method: the k-nearest neighbors

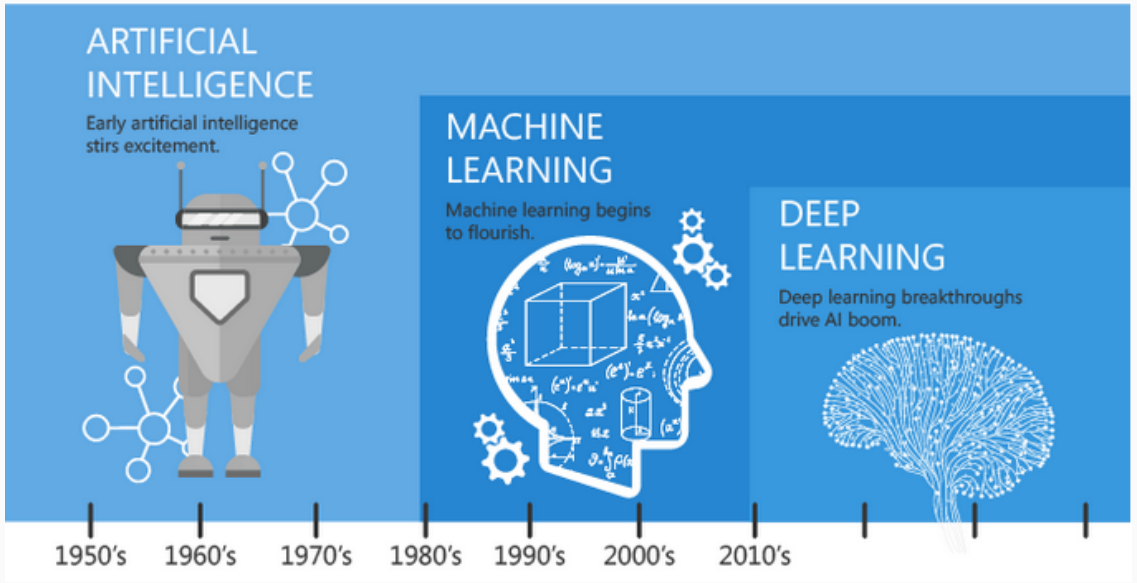
A ML dilemma: precision vs generalisation

Evaluation metrics

What is machine learning?

Definition

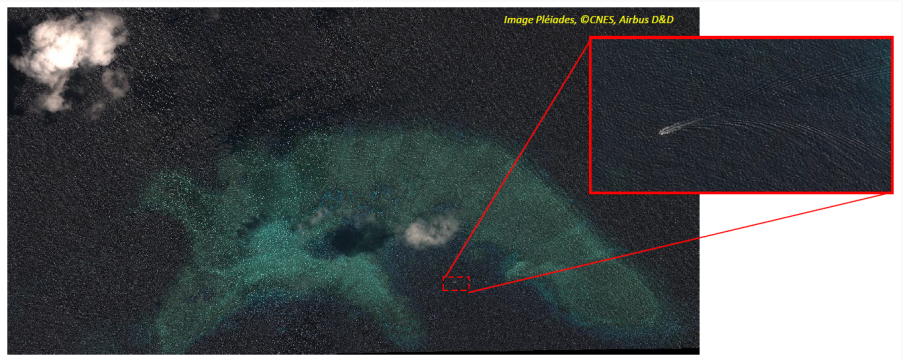
- Machine Learning algorithms enable the computers to learn from data, and even improve themselves, without being explicitly programmed (Arthur Samuel)
- Machine Learning is the study of algorithms that improve their performance P at some task T with experience E. A well-defined learning task is given by $\langle P, T, E \rangle$ (Tom Mitchell)



source:Linked In | Machine Learning vs Deep learning

When do we need machine learning?

- Human can not explain their expertise



- Facing a huge amount of data (25 peta bytes of data from sentinel satellites)
- Cost of the task is high (experts are often needed to label the scene)

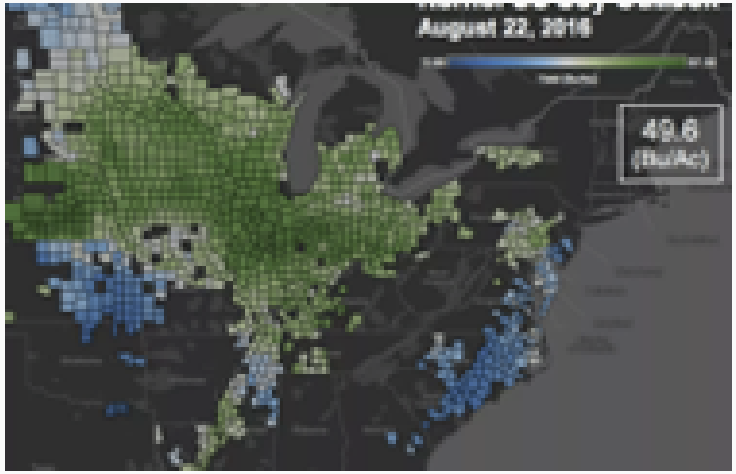
When do we need machine learning?

- Human can not explain their expertise
- Facing a huge amount of data (25 peta bytes of data from sentinel satellites)

- Cost of the task is high (experts are often needed to label the scene)

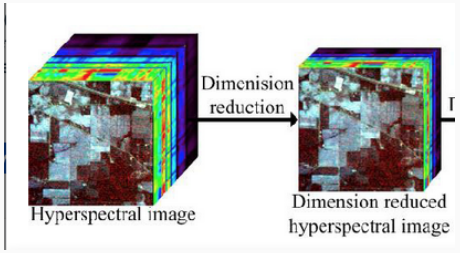
When do we need machine learning?

- Human can not explain their expertise
- Facing a huge amount of data (25 peta bytes of data from sentinel satellites)
- Cost of the task is high (experts are often needed to label the scene)



For which (RS) tasks do we need ML?

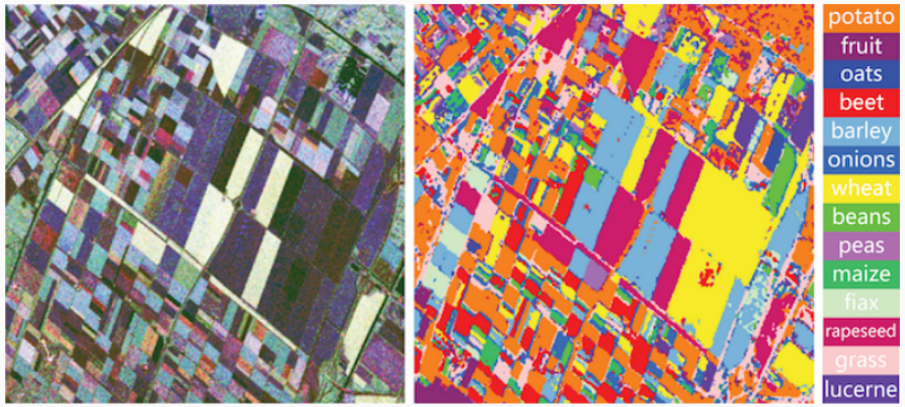
- Preprocess the data e.g. Dimension reduction techniques



- Build land use land cover map Classification task
- Detect objects of interest Retrieval
- Segmentation of an image Clustering
- Recognize a scene Classification task
- Detect change
- Fuse information

For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task

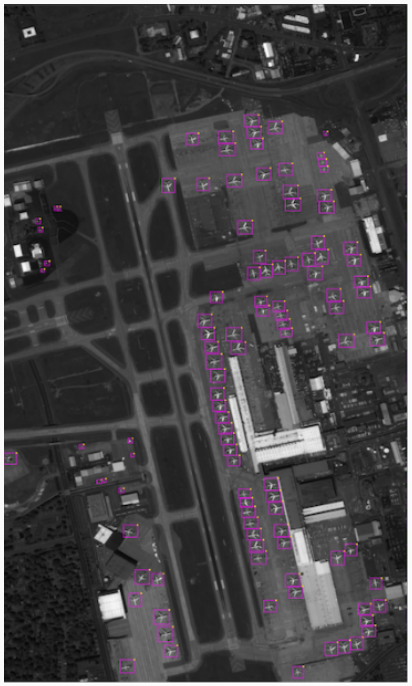


source:Deep Learning in Remote Sensing: A Review

- Detect objects of interest Retrieval
- Segmentation of an image Clustering
- Recognize a scene Classification task
- Detect change
- Fuse information

For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task
- Detect objects of interest Retrieval



source:Deep Learning in Remote Sensing: A Review

- Segmentation of an image Clustering
- Recognize a scene Classification task
- Detect change

For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task
- Detect objects of interest Retrieval
- Segmentation of an image Clustering

- Recognize a scene Classification task
- Detect change
- Fuse information

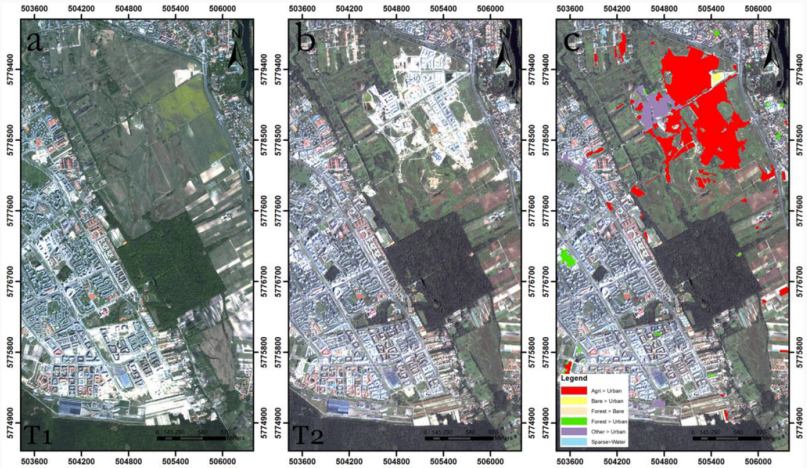
For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task
- Detect objects of interest Retrieval
- Segmentation of an image Clustering
- Recognize a scene Classification task

- Detect change
- Fuse information

For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task
- Detect objects of interest Retrieval
- Segmentation of an image Clustering
- Recognize a scene Classification task
- Detect change



source: Change Detection Algorithm for the Production of Land Cover Change Maps over the European Union Countries

- Fuse information

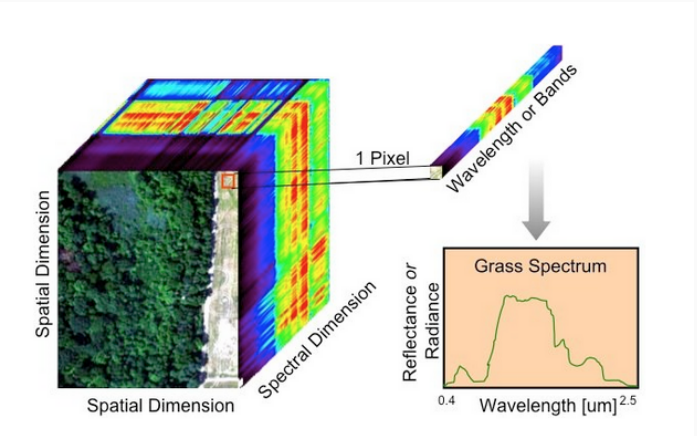
For which (RS) tasks do we need ML?

- Preprocess the data e.g. Dimension reduction techniques
- Build land use land cover map Classification task
- Detect objects of interest Retrieval
- Segmentation of an image Clustering
- Recognize a scene Classification task
- Detect change
- Fuse information



On which type of data do we apply ML?

- Vectors

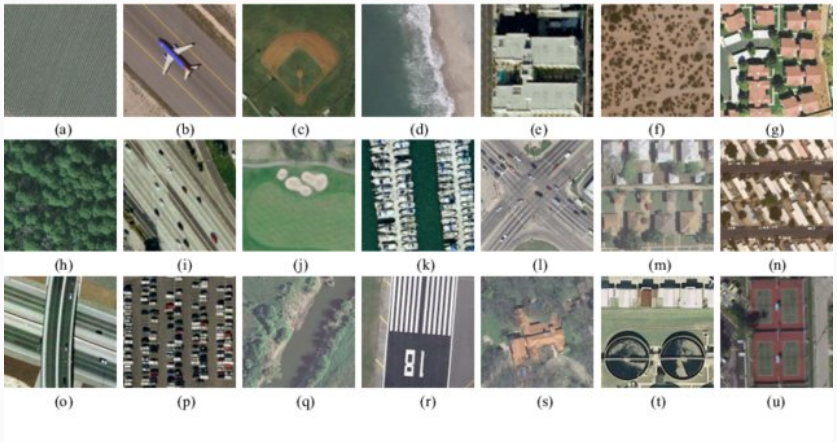


source: A manifold learning approach to target detection in high-resolution hyperspectral imagery

- Whole images
- Videos or time series
- Trees or graphs

On which type of data do we apply ML?

- Vectors
- Whole images

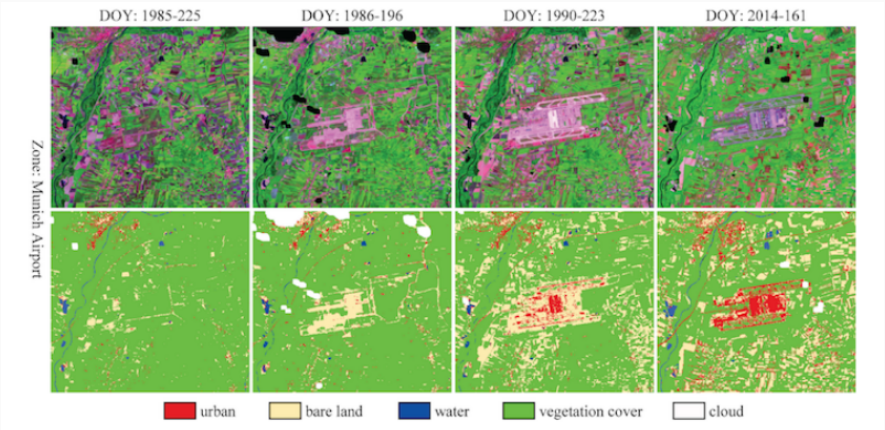


source: Merced dataset

- Videos or time series
- Trees or graphs

On which type of data do we apply ML?

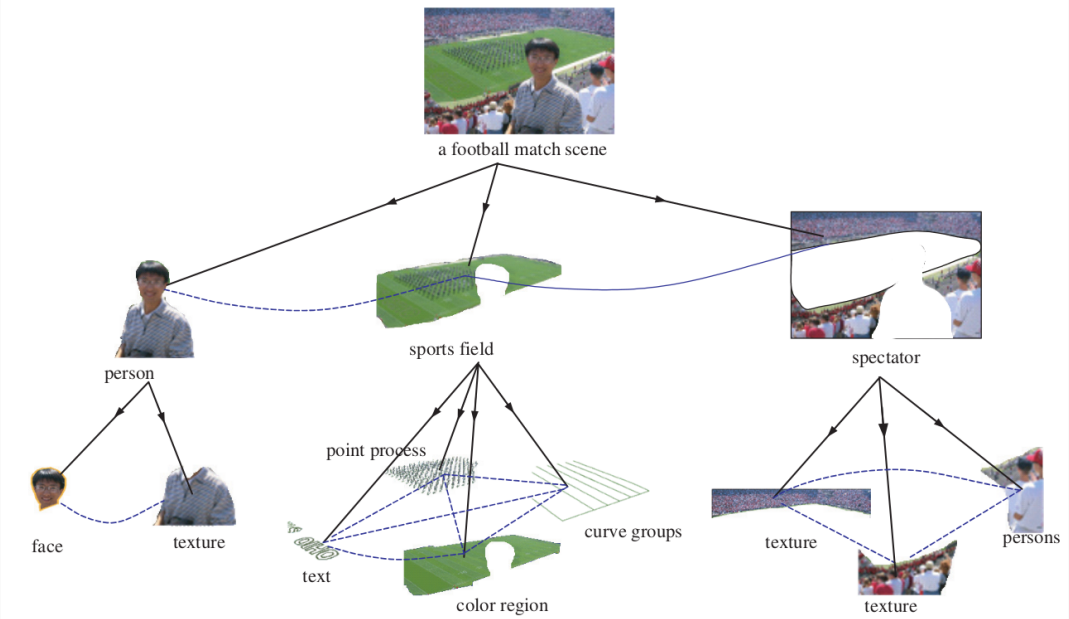
- Vectors
- Whole images
- Videos or time series



- Trees or graphs

On which type of data do we apply ML?

- Vectors
- Whole images
- Videos or time series
- Trees or graphs

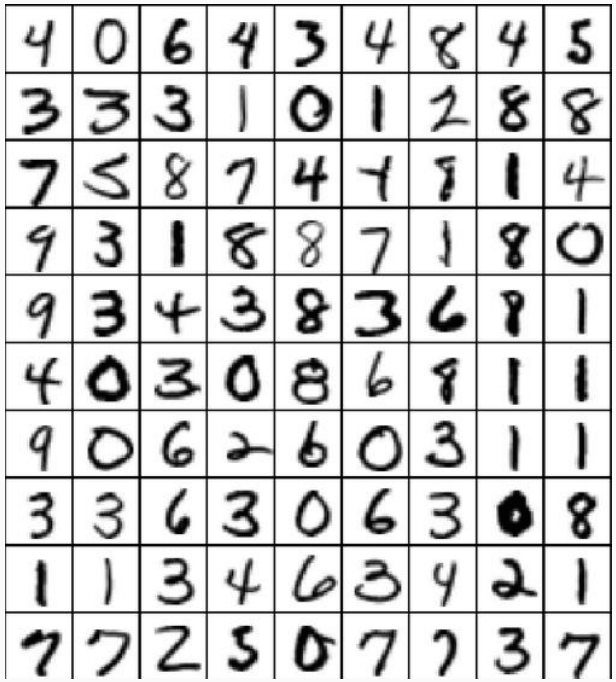


Characteristics of RS data

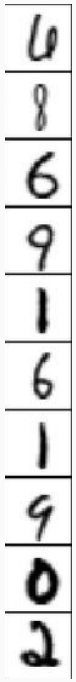
- often multi-modal (same scene described by different sensors)
- geo-located
- collected at several time instants
- (very very) big (both in variables and in size)

Key word of ML: generalisation

- Learn $\neq$ memorize



Learning set ($\mathbf{y} = [4, 0, 6, 4, \cdots, 7, 3, 3]$)



Test $\mathbf{y} = ?$

- Generalise to new data

Data

- A set of N points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \cdots, \boldsymbol{x}_N\} \in \mathcal{X} = \mathbb{R}^d$, where each point $\boldsymbol{x}_i$ contains d attributes (in this class) $\begin{bmatrix} x_{1i} \\ x_{2i} \\ \cdots \\ x_{di} \end{bmatrix}$
- Sometimes some labels $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix} \in \mathcal{Y}$

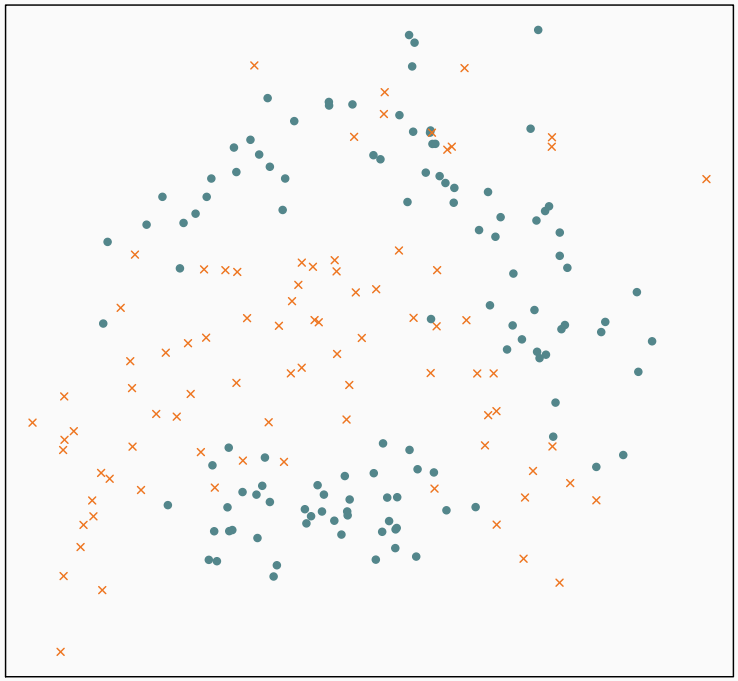
Different types of learning

- Supervised learning: labels are known
  Aim : find a function f such that $f : \mathcal{X} \to \mathcal{Y}$, $f(\boldsymbol{X}) = \boldsymbol{y}$.
  - if $\mathcal{Y} = \mathbb{R}$: regression problem
  - if $\mathcal{Y} \in \mathcal{S}$, with $\mathcal{S}$ a finite set: classification problem
  - if $\mathcal{S}$ contains 2 elements: binary classification
- Weakly supervised learning: few labels are known
- Unsupervised learning: the labels are unknown
  Aim: find the structure on the data
- Reinforcement learning: aiming at learning an action

A classification method: the k-nearest neighbors

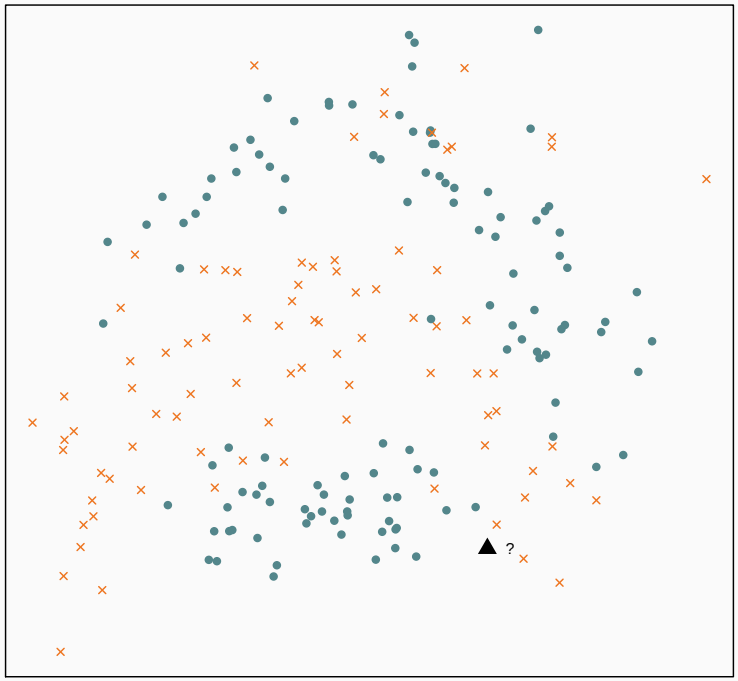
A simple example

- $N = 200$ points, 2 classes $\mathcal{Y} = \{\text{blue disks}, \text{orange cross}\}$



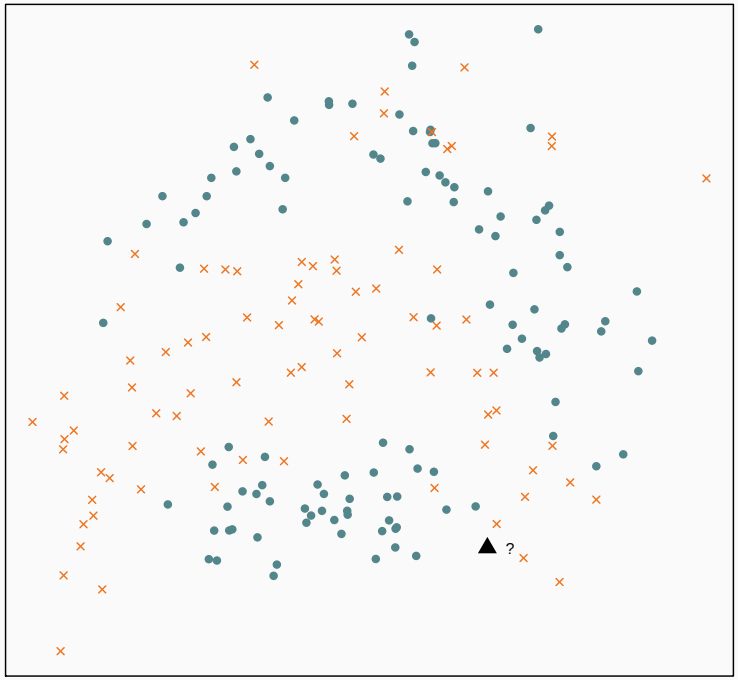
A simple example

- $N = 200$ points, 2 classes $\mathcal{Y} = \{\text{blue disks}, \text{orange cross}\}$



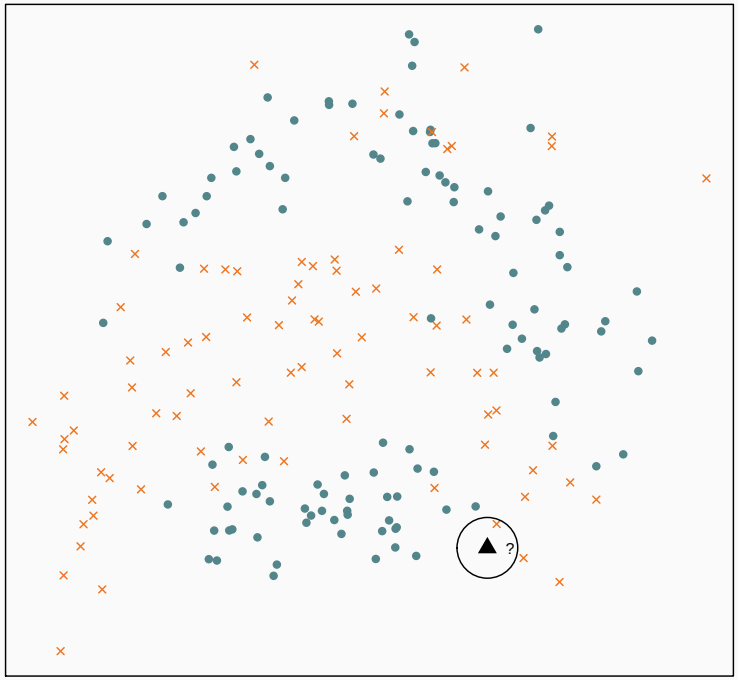
A simple example

- Overall idea: we look at the class of the k nearest neighbors



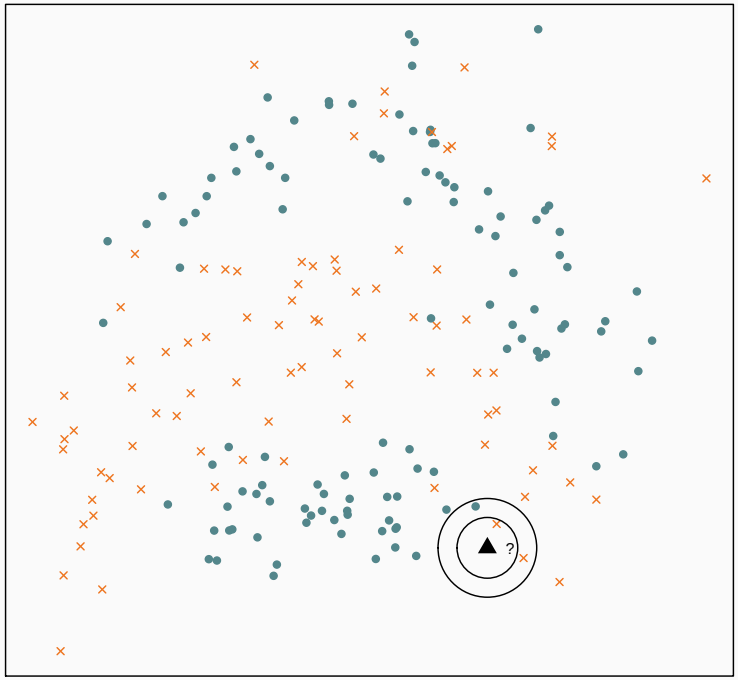
A simple example

- Overall idea: we look at the class of the k nearest neighbors



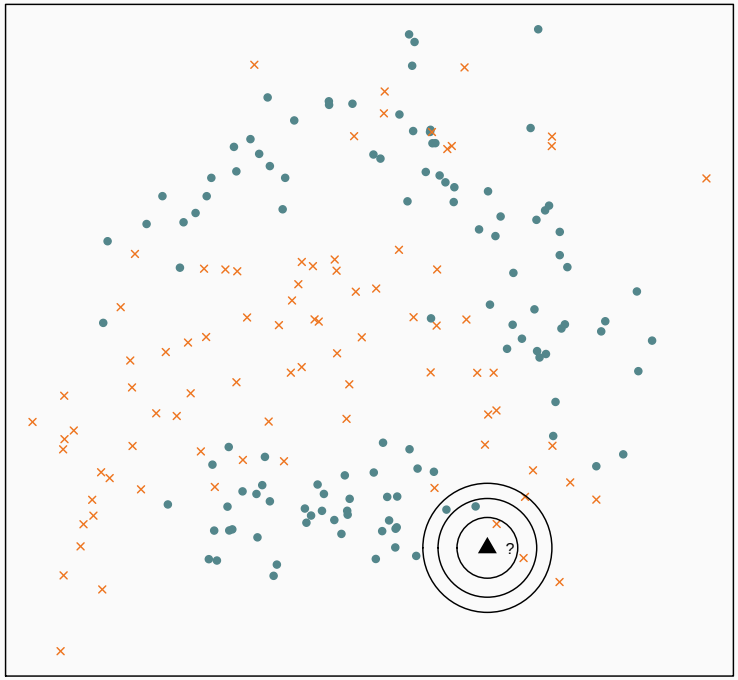
A simple example

- Overall idea: we look at the class of the k nearest neighbors



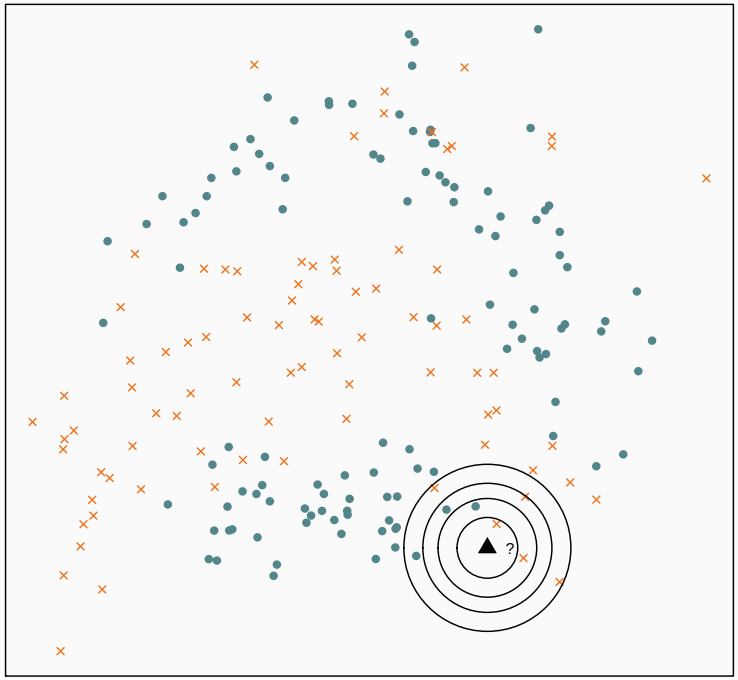
A simple example

- Overall idea: we look at the class of the k nearest neighbors



A simple example

- Overall idea: we look at the class of the k nearest neighbors



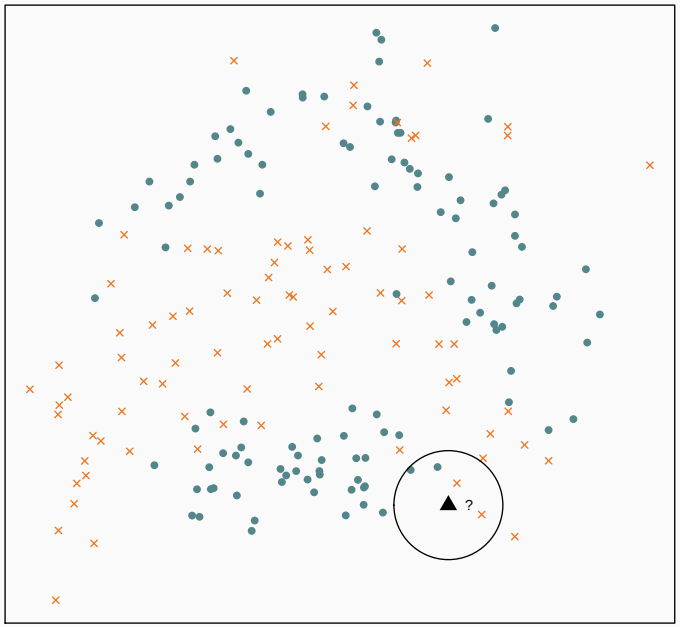
Algorithm

Be $\boldsymbol{x}$ a (new) point to classify

1. Compute $d(\boldsymbol{x}, \boldsymbol{x}_i)$, $\forall i \in 1, \cdots, N$
2. Sort $(j_i)_{i=1}^{N}$, $d(\boldsymbol{x}, \boldsymbol{x}_{j_i}) \leq d(\boldsymbol{x}, \boldsymbol{x}_{j_{i+1}})$
3. $\hat{y}$ = majority class among the k values $y_{j_1}, y_{j_2}, \cdots, y_{j_k}$

Algorithm

In case of ties

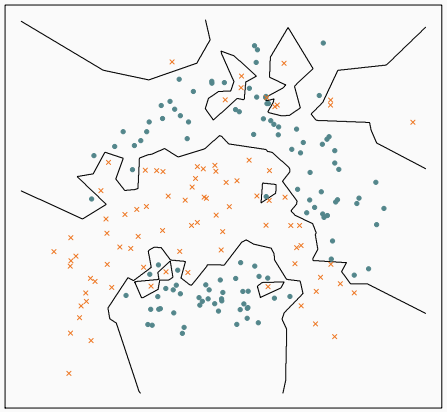


1. Raise the value of k of 1(may not solve the problem)
2. Draw at random the result Tirer au hasard la classe parmi les classes ambigües
3. Weigth the samples by their distances to the point x

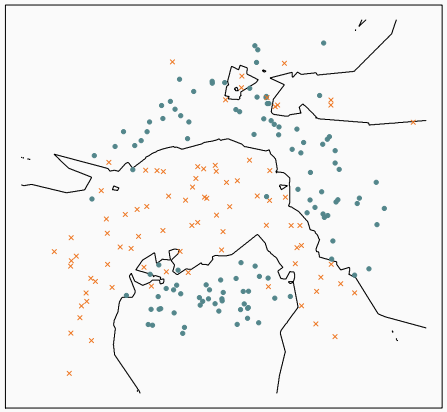
How to choose k

- small k
  - local decision
  - complex f
- large k
  - global decision;
  - simpler model
- What guarantees on the model?

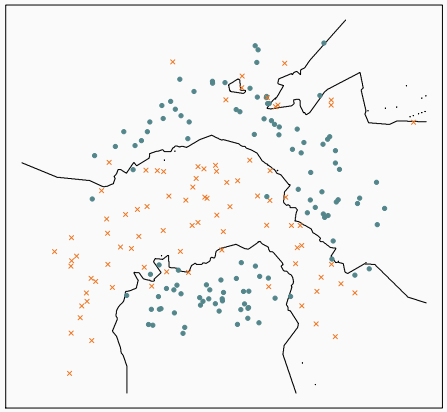
Influence of the k parameter



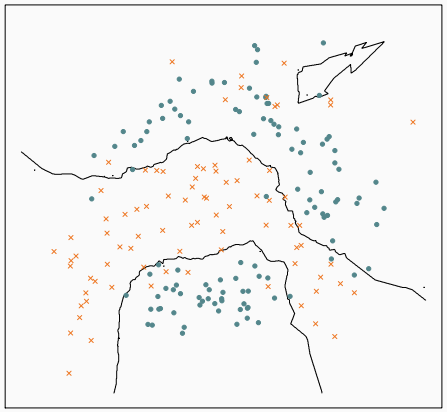
(a) $k = 1$



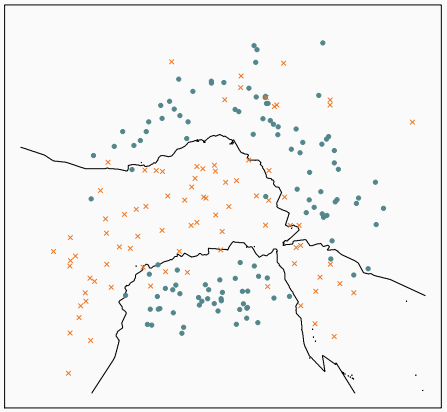
(b) $k = 3$



(c) $k = 5$



(d) $k = 9$

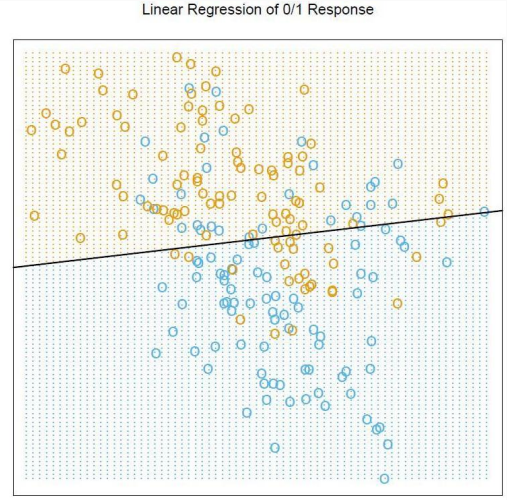


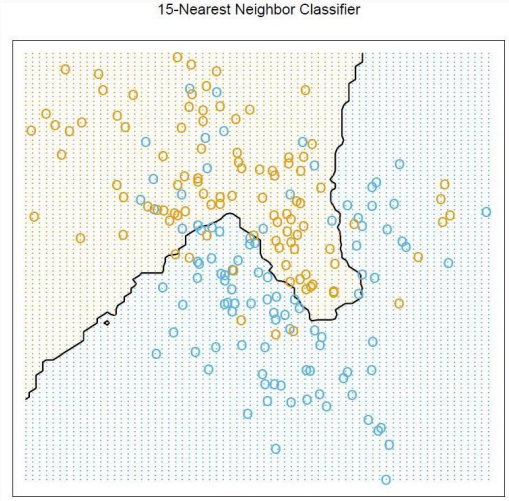
(e) $k = 15$

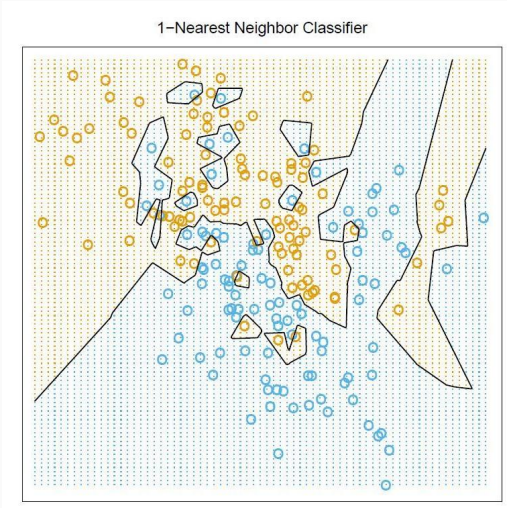
A ML dilemma: precision vs generalisation

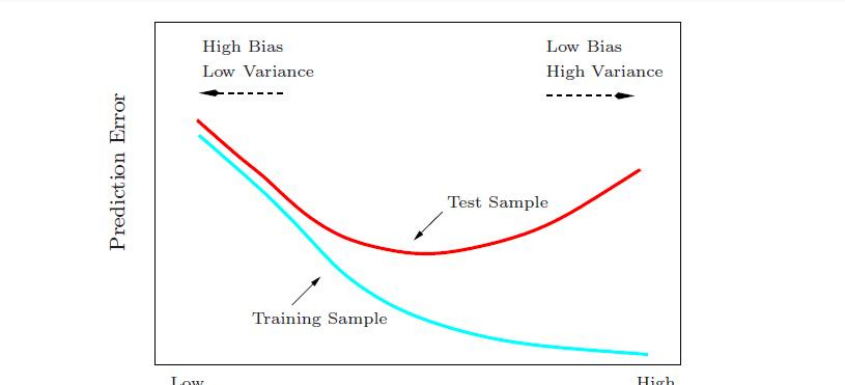
Generalisation vs precision

- The model has to be good when working on new (unseen) data (no learning by heart)









Bias-Variance Tradeoff

- bias error is an error from erroneous assumptions in the learning algorithm (e.g. model that is too simple) – underfitting
- variance is an error from sensitivity to small fluctuations in the training set – overfitting
- low variance results in high bias, low bias results in high variance
- Error = Bias + Variance + noise

A first evaluation criteria

- Remainder: solve a supervised learning problem $\Leftrightarrow$ find a function f such that $f : \mathcal{X} \to \mathcal{Y}$, $f(\boldsymbol{X}) = \boldsymbol{y}$
  $\Rightarrow$ evaluate the performances of the f function
  $\Rightarrow$ we define a loss function $\ell(y_i, f(\boldsymbol{x}_i))$ that computes the cost of taking decision $f(\boldsymbol{x}_i)$ when the correct label is y_i
- Linear regression $f(x) = y = \beta_0 + \beta_1 \cdot x_i$
  - Least square criteria: $\min_{\beta_0,\beta_1} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
  - $\ell(y_i, f(\boldsymbol{x}_i)) = (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
- Supervised classification: all the mistakes have the same weight

$$\ell(y_i, f(\boldsymbol{x}_i)) = \begin{cases} 0 \text{ si } y_i = f(\boldsymbol{x}_i) \\ 1 \text{ si } y_i \neq f(\boldsymbol{x}_i) \end{cases} \quad (1)$$

- In both cases, do we want to minimize $\frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i))$???

A first evaluation criteria

- Remainder: solve a supervised learning problem $\Leftrightarrow$ find a function f such that $f : \mathcal{X} \to \mathcal{Y}$, $f(\boldsymbol{X}) = \boldsymbol{y}$
  $\Rightarrow$ evaluate the performances of the f function
  $\Rightarrow$ we define a loss function $\ell(y_i, f(\boldsymbol{x}_i))$ that computes the cost of taking decision $f(\boldsymbol{x}_i)$ when the correct label is y_i
- Linear regression $f(x) = y = \beta_0 + \beta_1 \cdot x_i$
  - Least square criteria: $\min_{\beta_0, \beta_1} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
  - $\ell(y_i, f(\boldsymbol{x}_i)) = (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
- Supervised classification: all the mistakes have the same weight

$$\ell(y_i, f(x_i)) = \begin{cases} 0 \text{ si } y_i = f(x_i) \\ 1 \text{ si } y_i \neq f(x_i) \end{cases} \quad (1)$$

- In both cases, do we want to minimize $\frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i))$???

A first evaluation criteria

- Remainder: solve a supervised learning problem $\Leftrightarrow$ find a function f such that $f : \mathcal{X} \to \mathcal{Y}$, $f(\boldsymbol{X}) = \boldsymbol{y}$
  $\Rightarrow$ evaluate the performances of the f function
  $\Rightarrow$ we define a loss function $\ell(y_i, f(\boldsymbol{x}_i))$ that computes the cost of taking decision $f(\boldsymbol{x}_i)$ when the correct label is y_i
- Linear regression $f(x) = y = \beta_0 + \beta_1 \cdot x_i$
  - Least square criteria: $\min_{\beta_0, \beta_1} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
  - $\ell(y_i, f(\boldsymbol{x}_i)) = (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
- Supervised classification: all the mistakes have the same weight

$$\ell(y_i, f(\boldsymbol{x}_i)) = \begin{cases} 0 \text{ si } y_i = f(\boldsymbol{x}_i) \\ 1 \text{ si } y_i \neq f(\boldsymbol{x}_i) \end{cases} \tag{1}$$

- In both cases, do we want to minimize $\frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i))$???

A first evaluation criteria

- Remainder: solve a supervised learning problem $\Leftrightarrow$ find a function f such that $f : \mathcal{X} \to \mathcal{Y}$, $f(\boldsymbol{X}) = \boldsymbol{y}$
  $\Rightarrow$ evaluate the performances of the f function
  $\Rightarrow$ we define a loss function $\ell(y_i, f(\boldsymbol{x}_i))$ that computes the cost of taking decision $f(\boldsymbol{x}_i)$ when the correct label is y_i
- Linear regression $f(x) = y = \beta_0 + \beta_1 \cdot x_i$
  - Least square criteria: $\min_{\beta_0, \beta_1} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
  - $\ell(y_i, f(\boldsymbol{x}_i)) = (y_i - (\beta_0 + \beta_1 \cdot x_i))^2$
- Supervised classification: all the mistakes have the same weight

$$\ell(y_i, f(\boldsymbol{x}_i)) = \begin{cases} 0 \text{ si } y_i = f(\boldsymbol{x}_i) \\ 1 \text{ si } y_i \neq f(\boldsymbol{x}_i) \end{cases} \tag{1}$$

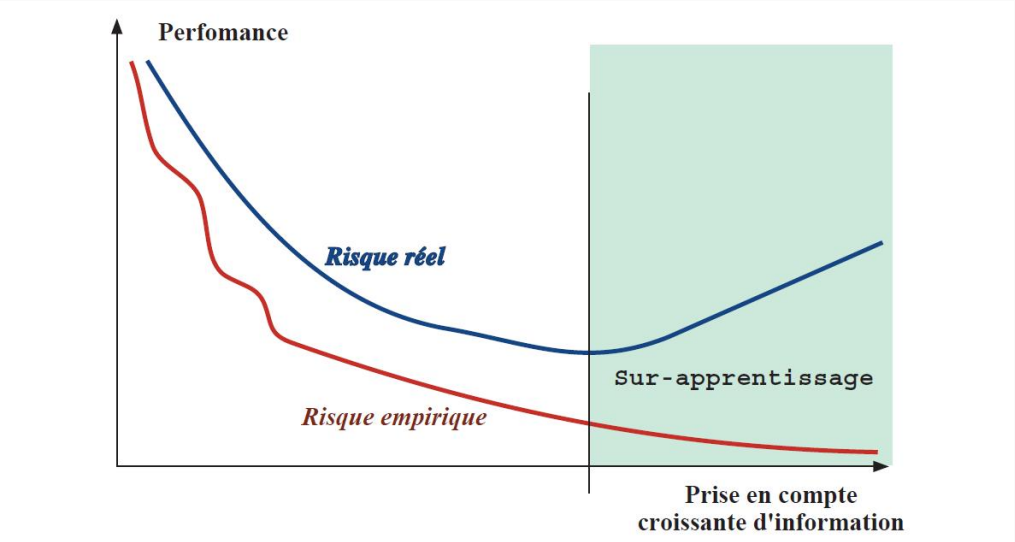
- In both cases, do we want to minimize $\frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i))$???

Risks

- The empirical risk (= computed on the data) – precision

$$\mathcal{R}_{\text{emp}} = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i)) \tag{2}$$

  is too optimistic and is not a good proxy for the actual risk $\mathcal{R}_{\text{actual}}$ (= computed on unseen data) – generalisation



- $\mathcal{R}_{\text{actual}} = \mathcal{R}_{\text{emp}} + \phi(d_{vc}(\text{complexity}, N))$

Measure of generalisation

Estimation of $\hat{\mathcal{R}}_{\text{actual}}$ of $\mathcal{R}_{\text{actual}}$

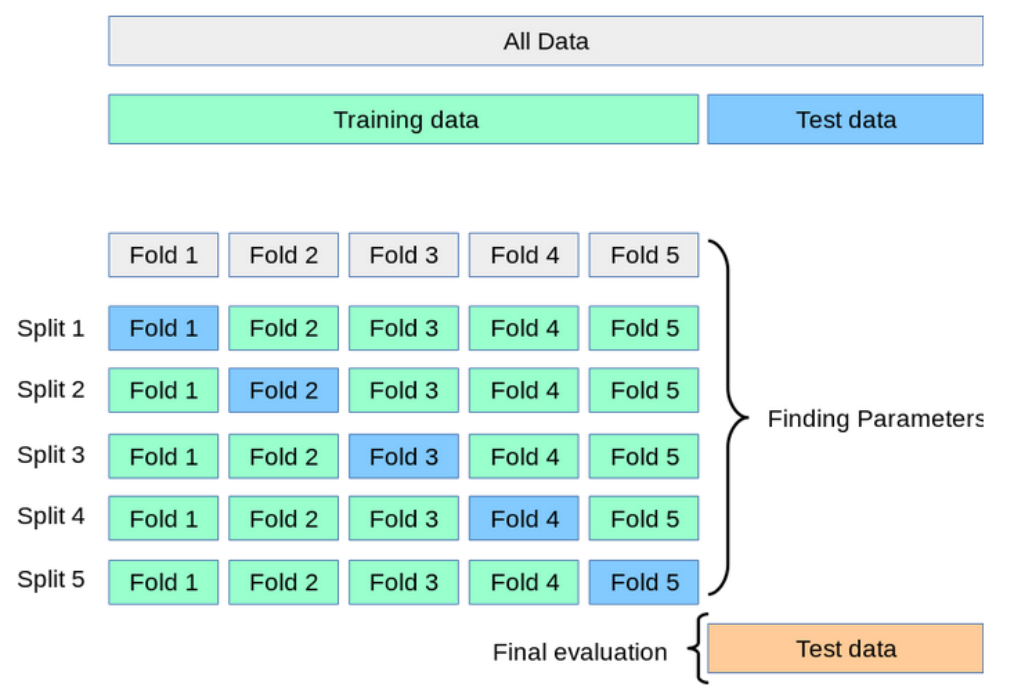
- learning set, validation set, test set
- cross validation
- *leave-one-out*
- bootstrap and others

Train-validation-test split

- Divide randomly the overall dataset in 3 sets:
  - Learning set $\mathcal{S}$
  - Validation set $\mathcal{V}$
  - Test set $\mathcal{T}$
- The error made on $\mathcal{T}$ is a proxy for $\hat{\mathcal{R}}_{\text{actual}}$ of the actual risk
- The optimal parameters are chosen thanks to the validation set $\mathcal{V}$
- We can do several $\mathcal{S}$ and $\mathcal{T}$ splits to have a confidence interval around the value

Cross validation

- Divide the dataset in k folds of equal size ($\neq k$ in the k nearest neighbor method!)



source: scikitlearn

Evaluation metrics

Confusion matrix – Classification task

		Predicted label			
		$\hat{y}_1$	$\hat{y}_2$	$\cdots$	$\hat{y}_L$
True label	y_1	n_{11}	n_{12}	$\cdots$	n_{1L}
	y_2	n_{21}	n_{22}	$\cdots$	n_{2L}
	$\vdots$				
	y_L	n_{L1}	n_{L2}	$\cdots$	n_{LL}

- "confusion" because it is easy to see if 2 classes are mixed up
- metrics
  - error rate $= \dfrac{\text{nb of misclassified samples}}{\text{total number of samples}} = \dfrac{\sum_{i,j,i\neq j} n_{ij}}{N}$
  - accuracy = 1 - error rate $= \dfrac{\sum_i n_{ii}}{N}$

When we have a class of interest $y = 1$

		Predicted label	
		$\hat{y}_1 = 1$	$\hat{y}_2 = 0$
True label	$y_1 = 1$	n_{11}	n_{12}
	$y_2 = 0$	n_{21}	n_{22}

		Predicted label	
		$\hat{y}_1 = 1$	$\hat{y}_2 = 0$
True label	$y_1 = 1$	True positives TP	False negatives FN
	$y_2 = 0$	False positives FP	True negatives TN

- Metrics
  - $\frac{n_{11}}{n_{11}+n_{12}} = \frac{VP}{VP+FN}$ = true positive rate TPR = sensitivity
  - $\frac{n_{22}}{n_{22}+n_{21}} = \frac{VN}{VN+FP}$ = true negative rate TNR = specificity
  - $\frac{n_{12}}{n_{12}+n_{11}} = \frac{FN}{FN+VP}$ = false negative rate FNR
  - $\frac{n_{21}}{n_{21}+n_{22}} = \frac{FP}{FP+VN}$ = false positive rate FPR

Other evaluation metrics

- Depends on the type of task
  - classification: often a combination of TPR/TNR/FNR/FPR
  - reduction of dimension: % of inertia kept
  - clustering: homogeneity of the clusters w.r.t. the number of clusters
  - etc.

What's next

What's next

- Regression task (the label to be predicted is a continuous variable)
- Regression with multitemporal data + a classification algorithm: SVM
- Tree-based algorithms and ensemble learning theory
- Dimension reduction, attribute selection, other learning paradigms
- How to deal with physical models, dynamical systems?