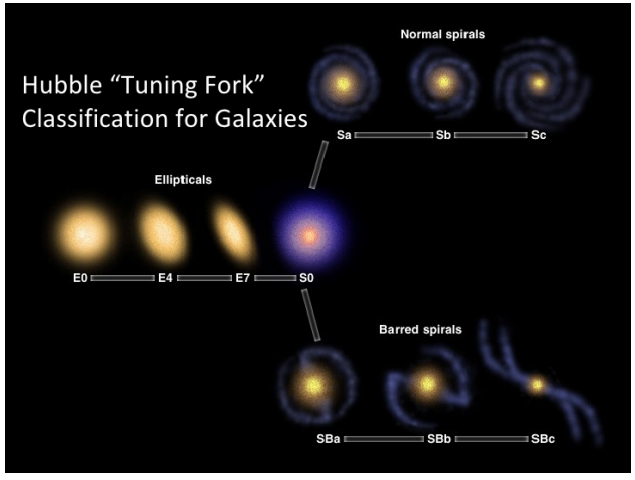
# Introduction à l'apprentissage statistique

Stéphane Canu
asi.insa-rouen.fr/enseignants/~scanu
scanu@insa-rouen.fr




École d'été BasMatI 2018

Porquerolles, june 6, 2018

# Plan

1. Le cas « Charlie »

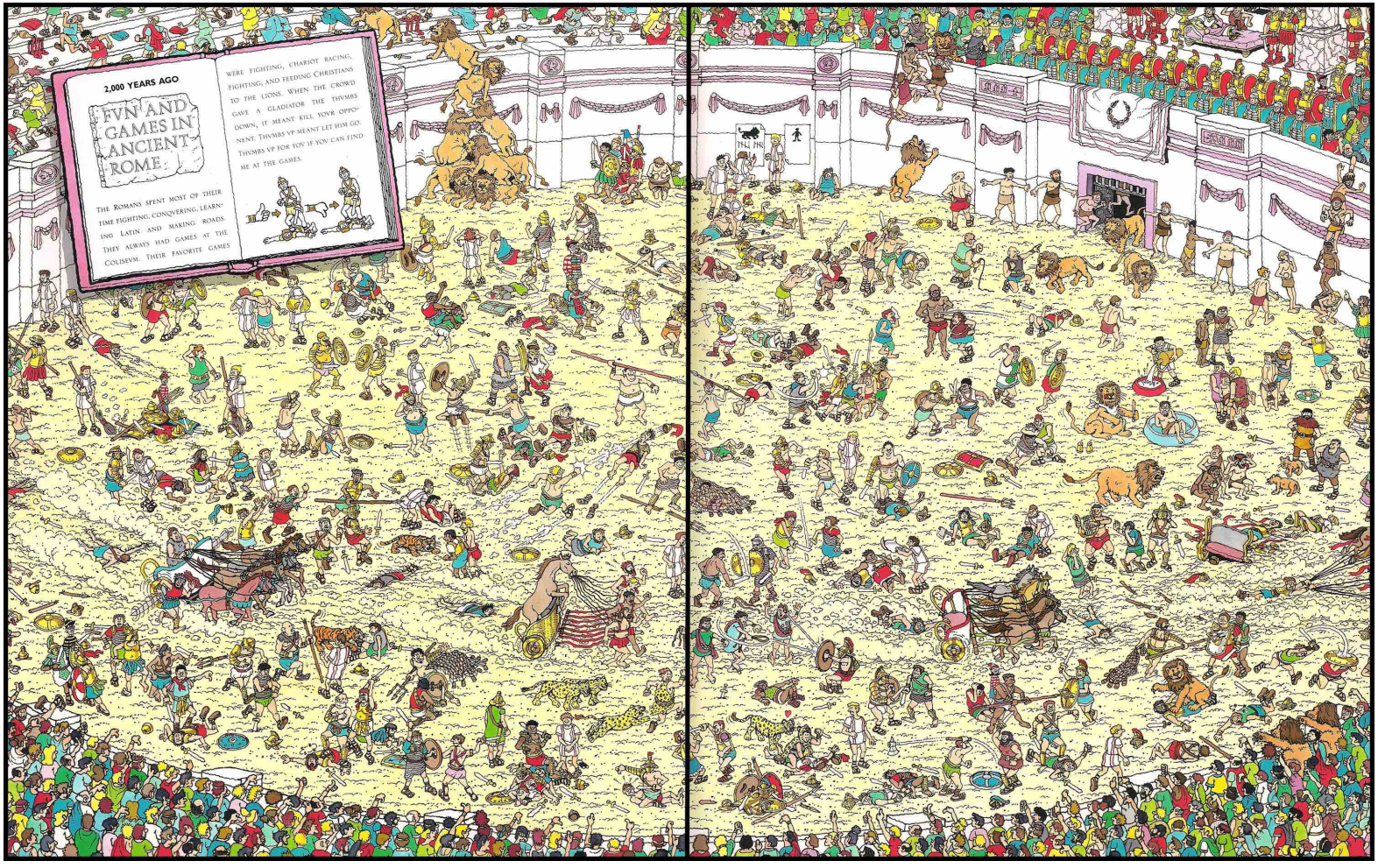
2. L'apprentissage statistique
   - Une brève définition de l'apprentissage statistique
   - Qu'est-ce qu'un bon ensemble d'hypothèses ?

3. Qu'est-ce que la généralisation

# Plan

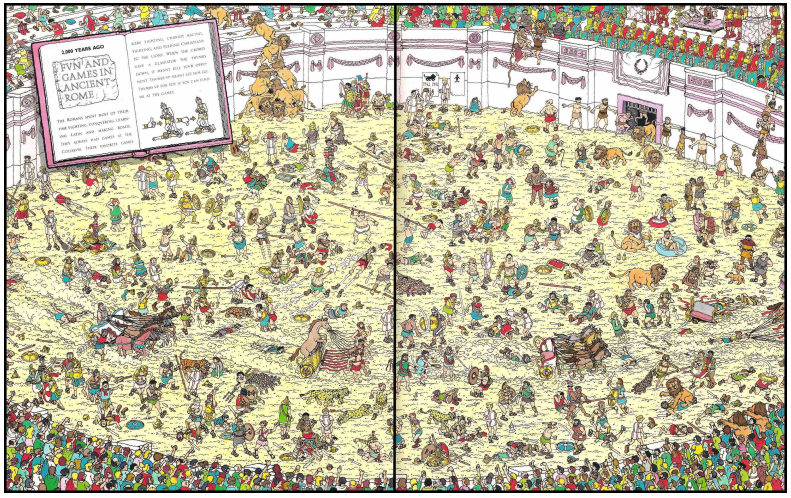


# Où est Charlie ?




**d'après** `randalolson.com/2015/02/03/heres-waldo-computing-the-optimal-search-strategy-for-finding-waldo/`

# La **question** du point de vue de l'apprentissage

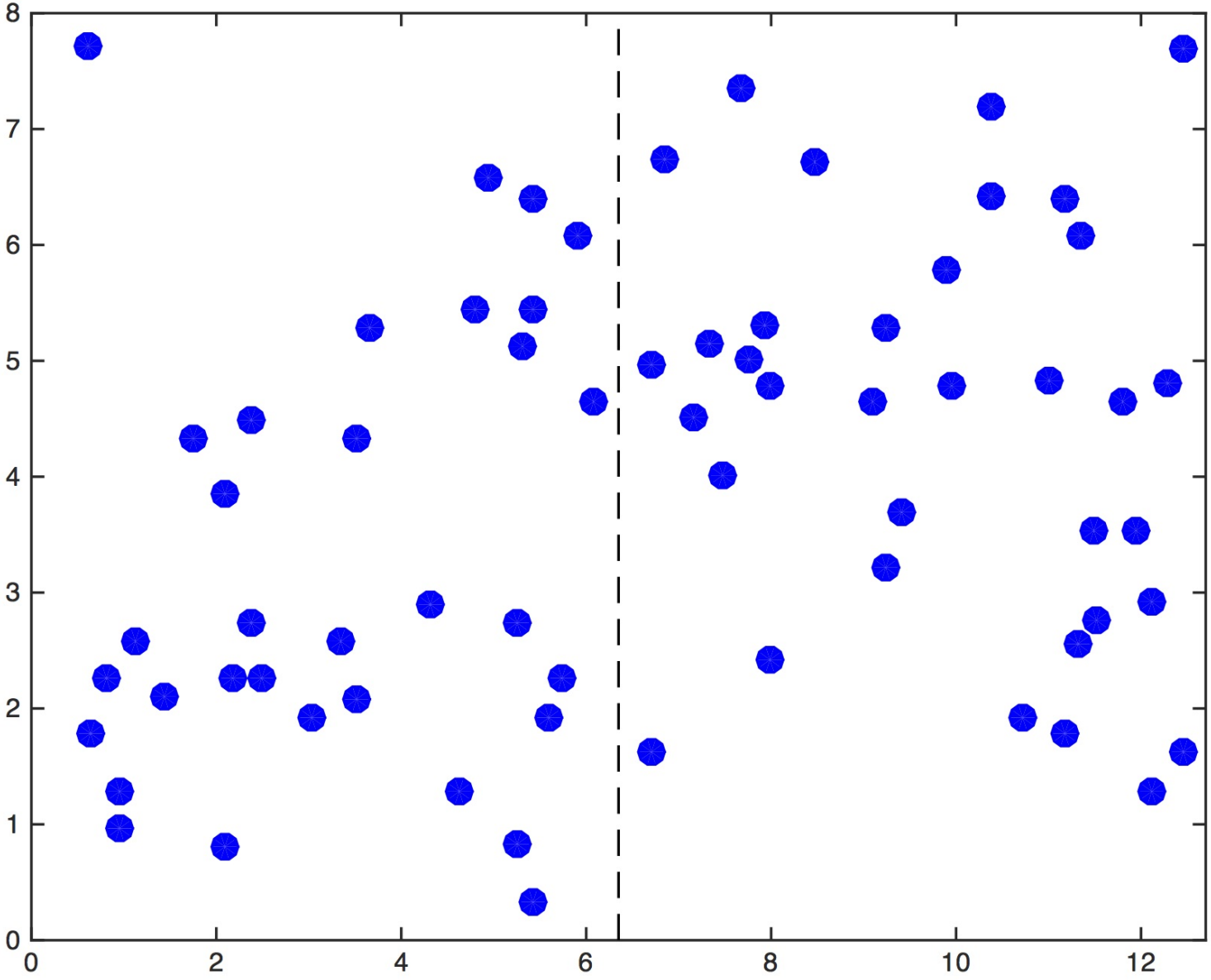



- Charie ou pas Charlie ?


$\rightarrow$ *supposé résolu.*

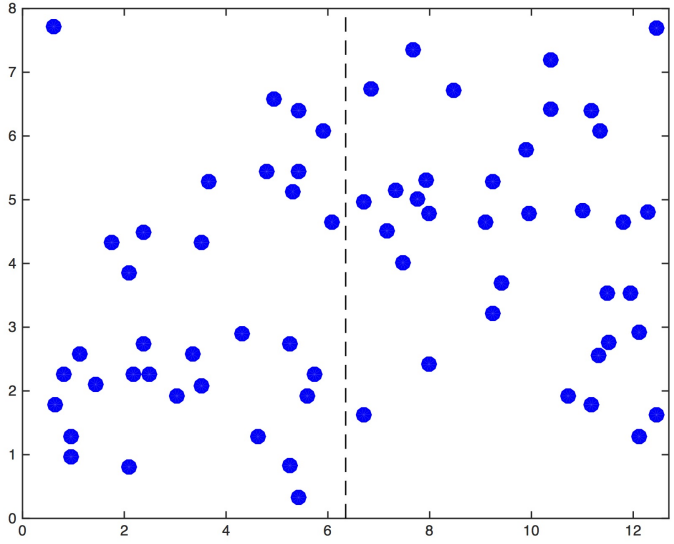
- comment trouver Charlie le plus rapidement possible ?
  $\rightarrow$ **où regarder ?**

# Où **était** Charlie : les DONNÉES




Les 68 localisations de Charlie dans les 7 premiers livres

# Où **était** Charlie : les DONNÉES sous forme matricielle



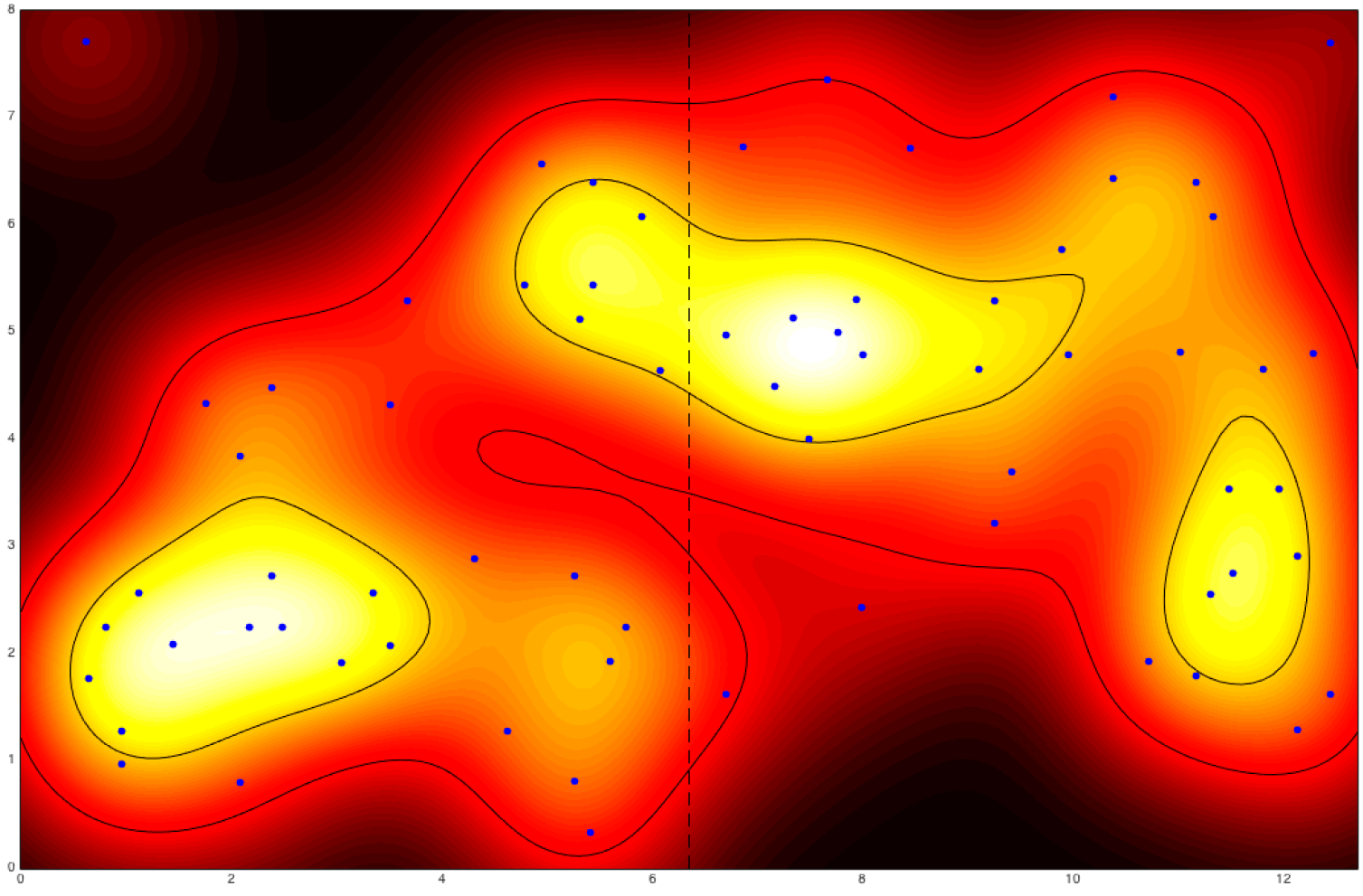

$$M = \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_i & y_i \\ \vdots & \vdots \\ x_{68} & y_{68} \end{pmatrix}$$

$$\begin{cases} n = 68 \text{ observations ou individus} \\ p = 2 \text{ variables} \end{cases}$$

## Données additionnelles

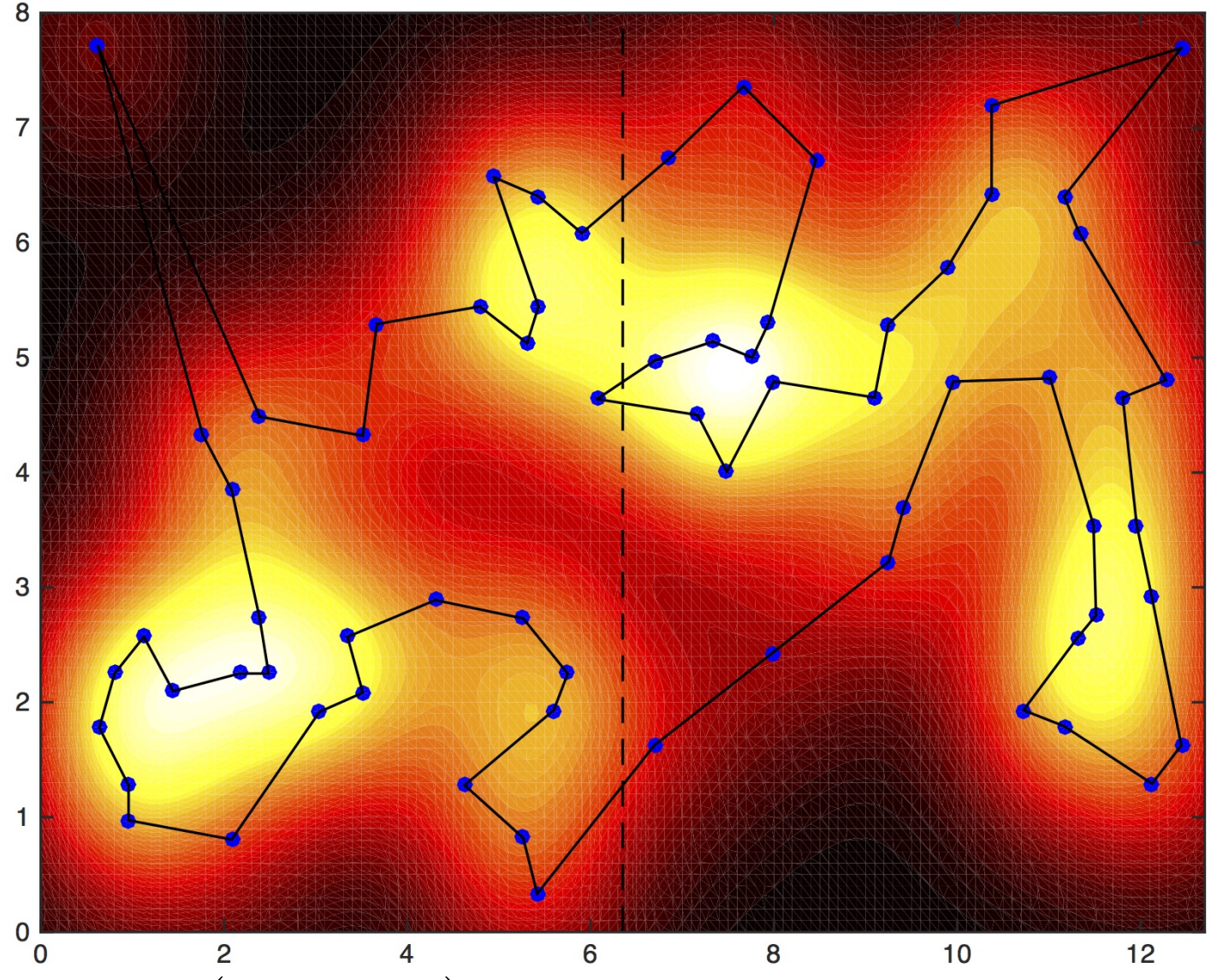
- nouvelles variables explicatives : histogramme des couleurs
- variables à expliquer : étiquette 1 = Charlie / -1 = pas Charlie

# Où **pourrait être** Charlie : le MODÉLE




$f(x, y, \theta)$ : entrèes, paramètres $\longrightarrow$ sortie

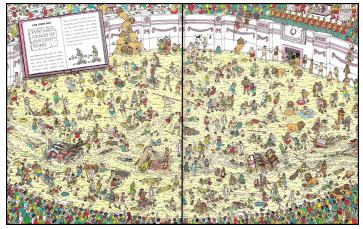
# Comment trouver Charlie efficacement : en OPTIMISANT

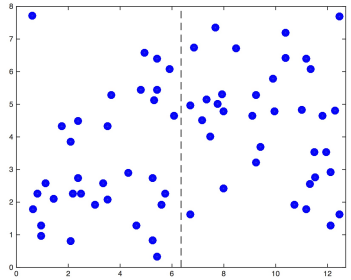


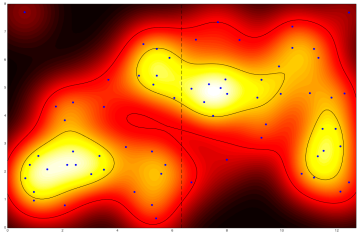

$\min_{\theta}$ *cout*(données, $\theta$)  minimiser le temps de parcours

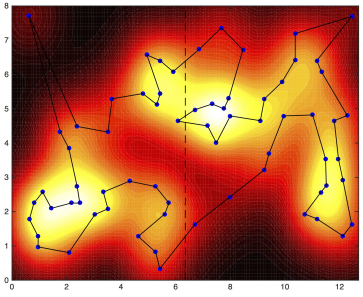
# La démarche de l'apprentissage statistique

- poser une question
  $\rightarrow$ *définir un cout*
- trouver les données
  $\rightarrow$ *des exemples*
- construire un modèle
  $\rightarrow f(x, y, \theta)$
- optimiser : **identifier le modèle**
  $\rightarrow \min_\theta \ cout\big(\text{données}, \theta\big)$







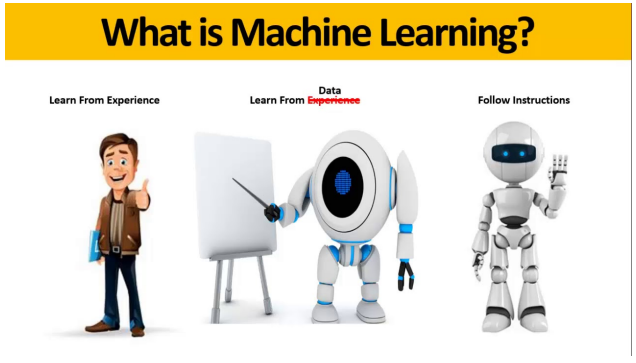


**Généraliser**

$\rightarrow$ Appliquer le modèle à de nouvelles données

# Plan

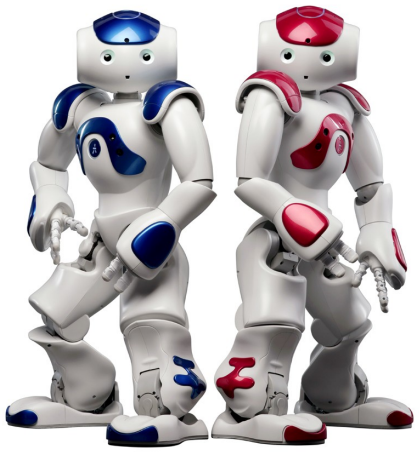


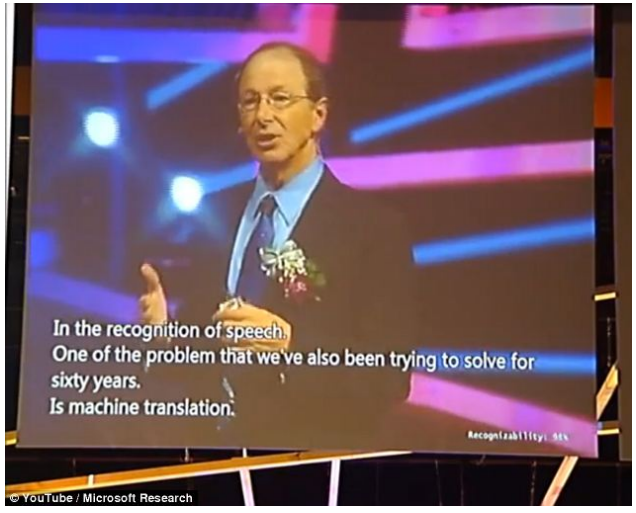


# Apprentissage : humain vs. machine

## Les apprentissages d'un enfant

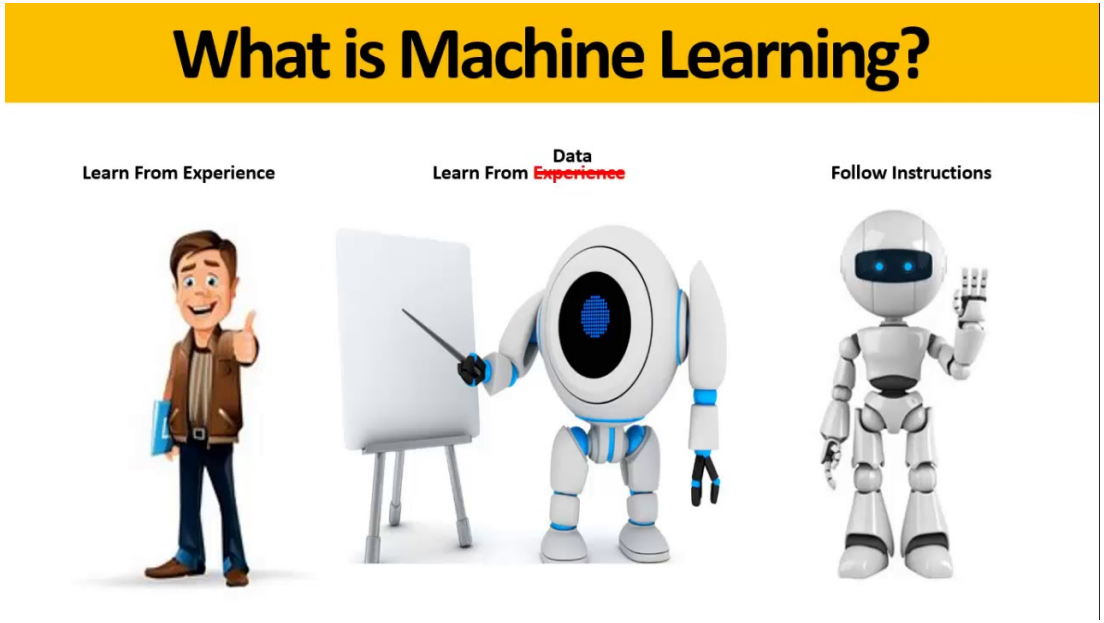
- marcher : un an
- parler : deux ans
- raisonner : le reste






apprendre à partir d'expérience vs. un programme suit des instructions
apprendre à programmer à partir de données.

# Exemples de machines capables d'apprendre




- apprendre à prédire le type d'une galaxie
- apprendre à estimer une durée de vie
- apprendre (découvir) les différent type de galaxies
- apprendre recommander une ligne de code

d'après `https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge`

# Tentative de définition de l'apprentissage statistique

## Machine Learning (T. Mitchell, 2006)

A computer program $\mathcal{A}$ learn from experience $E$ with respect to some class of tasks T and performance measure $L$, if its performance at tasks in T, as measured by $L$, improves with experience $E$

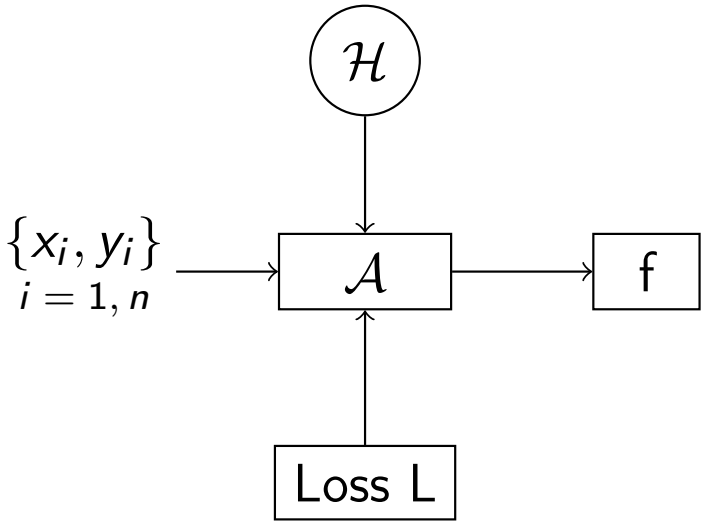

## La bande des 4

- programme $\mathcal{A}$ : algorithme $\mathcal{A}$
- experience E : données $\{x_i, y_i\}$
- performance measure $L$ : cout $L$
- tâche T : modèle $f \in \mathcal{H}$
  - traduire
  - jouer aux échec ou au go
  - conduire
  - ...faire ce que les humains font

# Apprentissage statistique et optimisation

## Composantes :

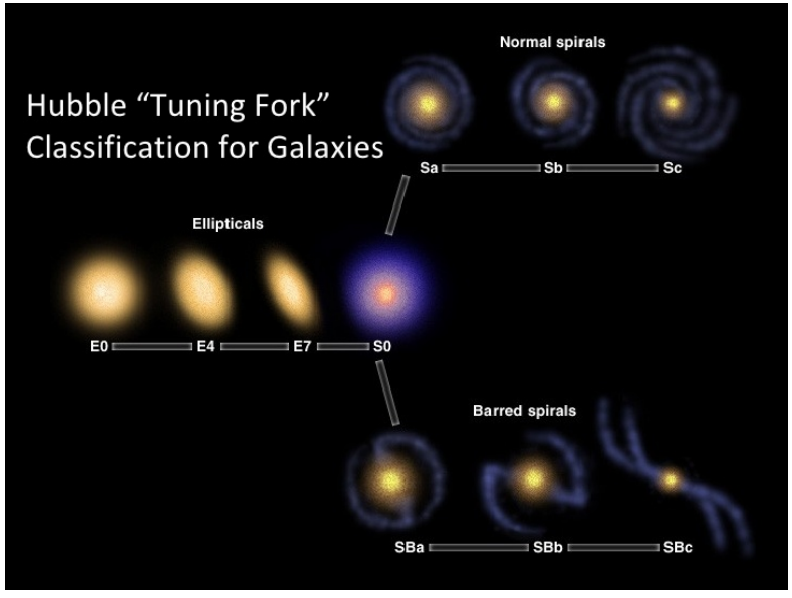
- programme $\mathcal{A}$ :
  algorithme $\mathcal{A}$ .
- experience E :
  données $\{x_i, y_i\}$
- performance $L$ :
  cout $L$ .
- tâche - hypothèses T :
  modèle $f \in \mathcal{H}$




## L'apprentissage comme une optimisation

$$\min_{f \in \mathcal{H}} L(f, \{x_i, y_i\}, i = 1, n) \qquad \hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} L = \mathcal{A}(\mathcal{H}, \{x_i, y_i\}, L)$$

# Exemples de machines capables d'apprendre




| **tâche** | **données** | **performance** |
|---|---|---|
| prédire le type d'une galaxie | images | taux d'erreur |
| estimer une fréquence | images | écart quadratique |
| (découvrir) des types de galaxies | images | entropie de la partition |
| recommander une ligne de code | programmes | renforcement |

# Les tâches T

- apprendre à regarder
- apprendre à entendre
- apprendre à marcher
- apprendre à parler (la notion de pipline)
  - à traduire
  - à dialoguer
- apprendre à aider à soigner (à décider)
- apprendre à conduire (garantir l'apprentissage)
  - 100k vidéo BAIR dataset
- apprendre à jouer (au go)
- apprendre à raisonner (pas encore automatique)

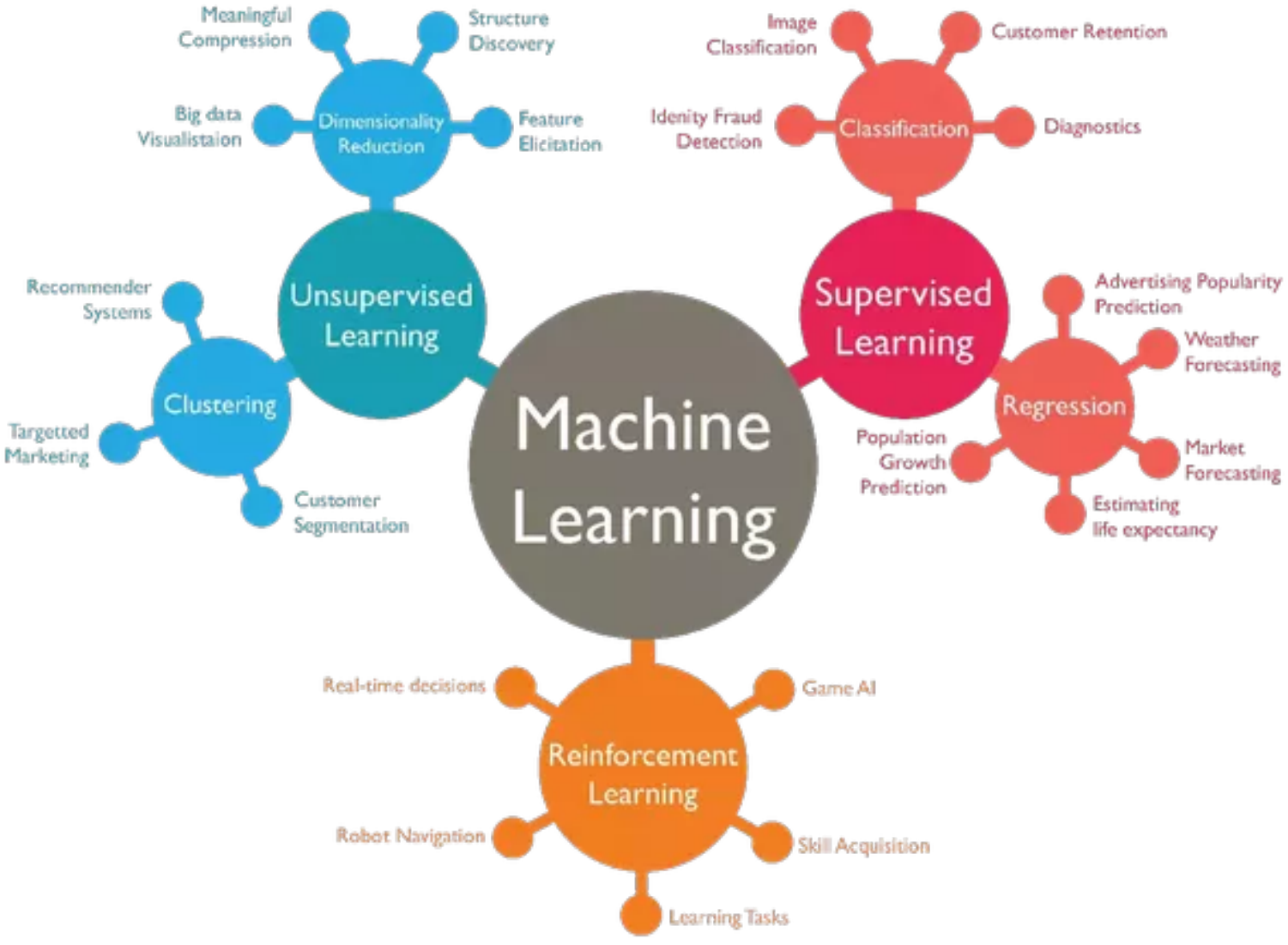
# L'experience E : les données

| | |
|---|---|
| capteurs → | variables qualitatives / ordinales / quantitatives |
| texte → | chaine de caractères |
| parole → | temps - série temporelle |
| images/vidéos → | dépendances 2/3 d |
| réseaux → | graphes |
| jeux → | séquences d'interactions |
| flots → | tickets de caisse, web logs, traffic... |
| étiquettes → | information d'évaluation |

## deux points importants

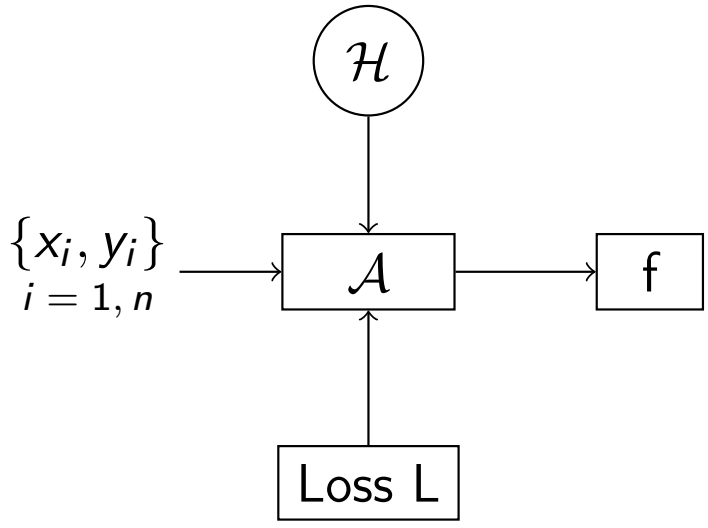
- Les mégadonnées (volume, vitesse, variété, véracité et valeur)
- Les données que l'on connait et celles que l'on ne connait pas encore

# Objectifs et tâches




et les autres : apprentissage par transfert, actif, semi supervisé, en ligne . . .

# Apprentissage et ensemble d'hypothèses




**apprendre c'est :**

sélectionner, parmi des hypothèses, la plus cohérente avec les données

# Les caractéristiques d'un bon ensemble d'hypothèse

- permettre de résoudre des problèmes plus ou moins complexes
  - universel
- facile à mettre en œuvre
  - calculable

# Universalité de l'apprentissage

résultat d'existence

## Théorème d'approximation universel

- soit $\varepsilon > 0$ un réel
- quelle que soit la tâche à apprendre $\forall f^\star$
- il existe $\hat{f} \in \mathcal{H}$, telle que

$$\|f^\star(x) - \hat{f}\| \leq \varepsilon$$

*n* grand plus on a d'exemple, plus le modèle doit pouvoir être complexe

# Les différents types de modèles universels

- modèles basées sur les variables
  - la notion de dictionnaire (base, polynômes, ondelettes. . . )

$$\hat{f}(x) = \sum_{k=1}^{k} \alpha_k \phi_k(x)$$

- modèles basées sur les exemples
  - plus proches voisins
  - noyaux (SVM)

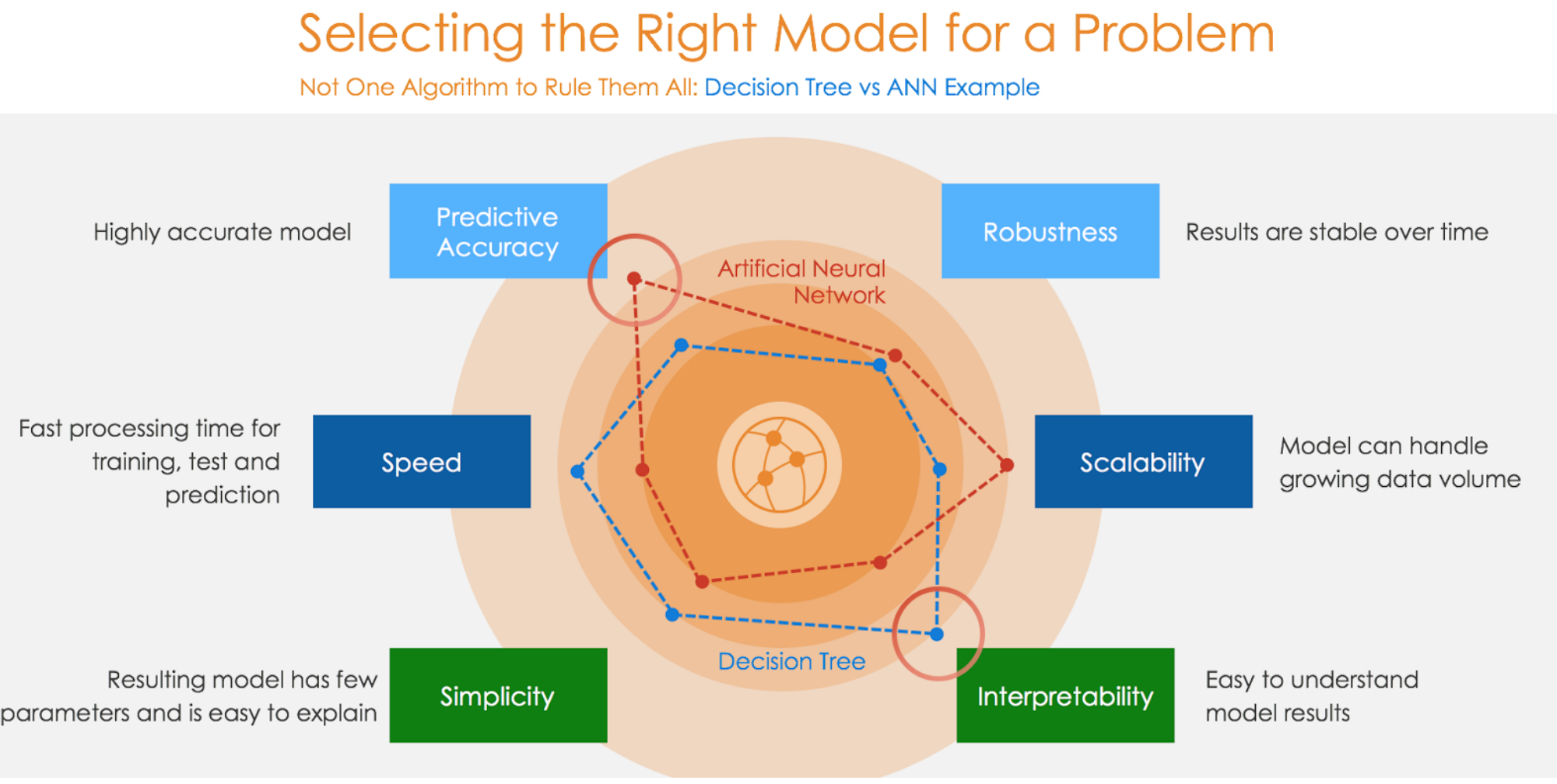
- combinaison d'éléments simples (partitions)
  - les forêts aléatoires
  - le boosting

- modèles profonds (deep learning -> cf. cours de M. Cord)

**linéaires vs non linéaires**

$\rightarrow$ C'est un problème de représentation (extraction de caractéristiques)

# L'universalité n'est pas le seul critère



# le modèle doit être universel et efficace

**universel** capable d'assimiler les données

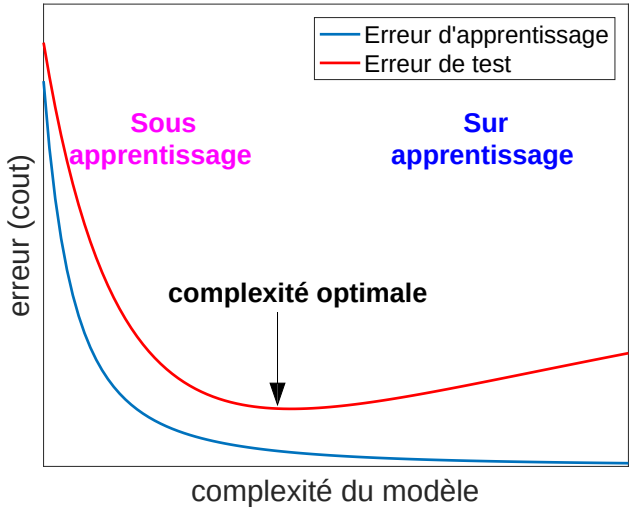
**efficace** en un temps raisonnable

**généraliser** donner de bonnes prédictions sur de nouveaux exemples

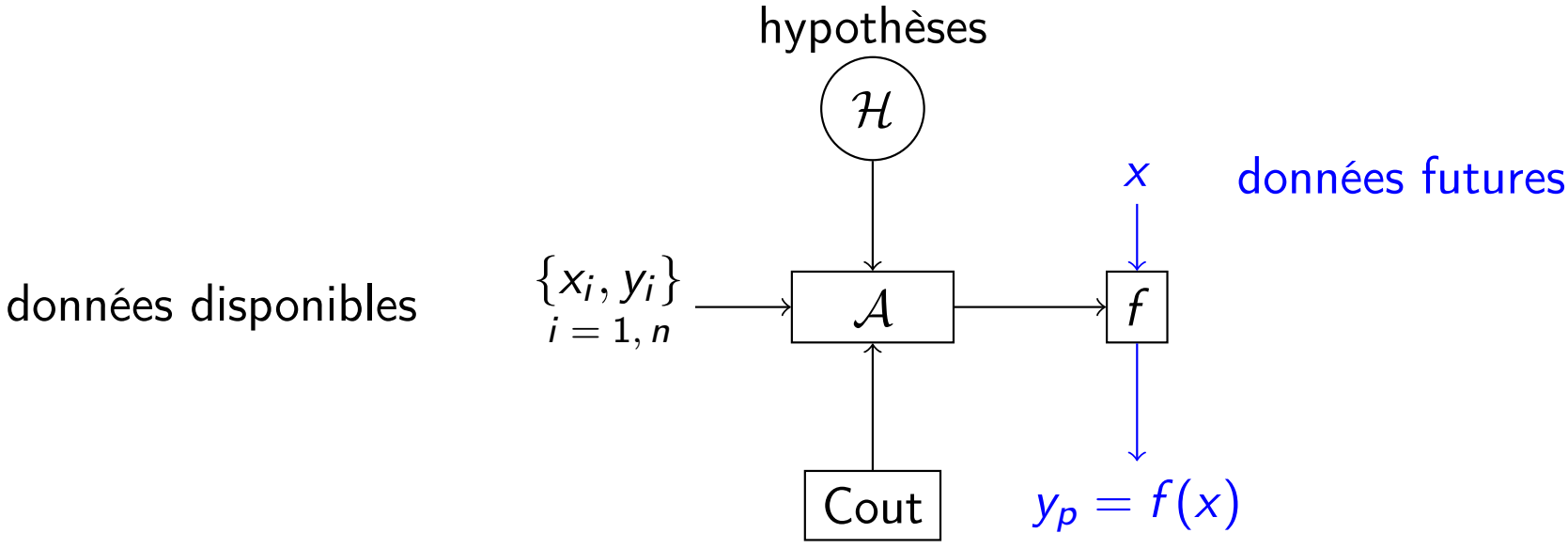
A t'on des garanties ?

# Plan





# Deux types de données en apprentissage




Apprendre = minimiser le cout sur les données disponibles

Généraliser = bien se comporter sur des données future

**Utiliser des exemples pour mesurer la capacité à généraliser**

données disponibles = ensemble d'apprentissage + ensemble de test

# Illustration : apprentissage « par cœur »

## Exemple (stupide) de système à zéro erreur : la mémorisation

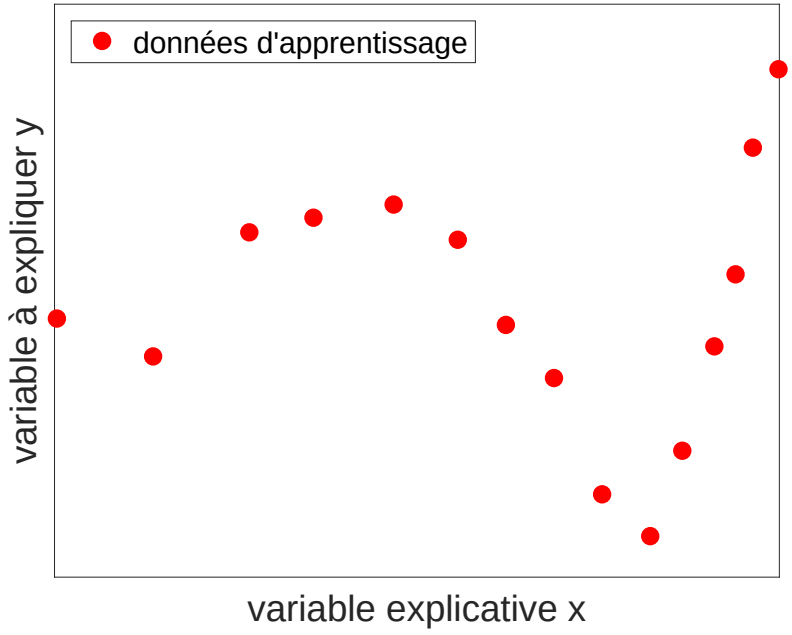
Fonction y $\leftarrow$ `predit`($x$, ensemble d'apprentissage $(x_i, y_i), i = 1, n$)

**Si** ($\exists i \in \{1, n\}$ tel que $x == x_i$) **Alors**
  $y \leftarrow y_i$
**Sinon**
  $y \leftarrow$ *vide*
**Fin Si**

Zéro erreur sur l'ensemble d'apprentissage

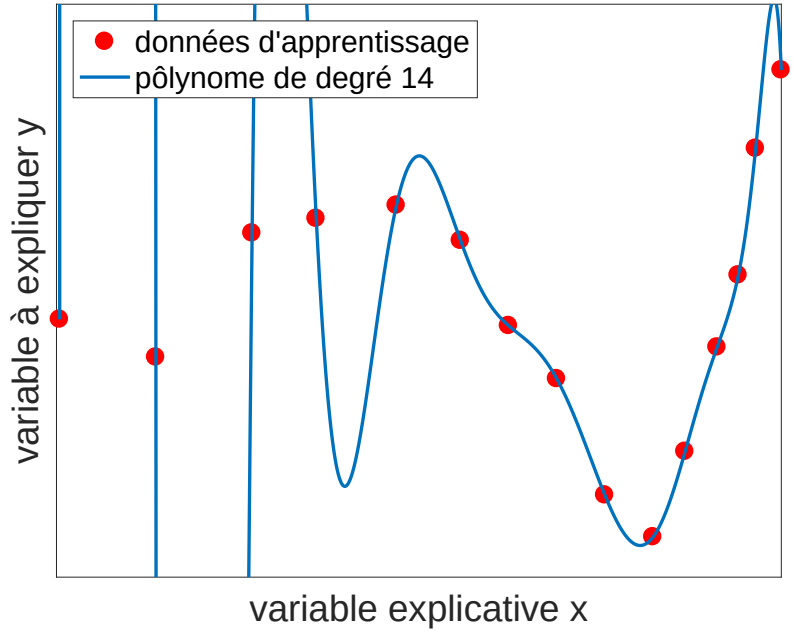
# Exemple d'apprentissage par cœur : l'interpolation

- définir un cout : $(predction - observation)^2$
  $(\quad f(x) \quad - \quad y \quad)^2$
- données 1d : $(x_i \in \mathbb{R}, y_i \in \mathbb{R}), i = 1, n$
- modèle : polynômes $f(x) = \sum_{j=0}^{d} \beta_j x^j$
- optimisation : méthode des moindres carrés



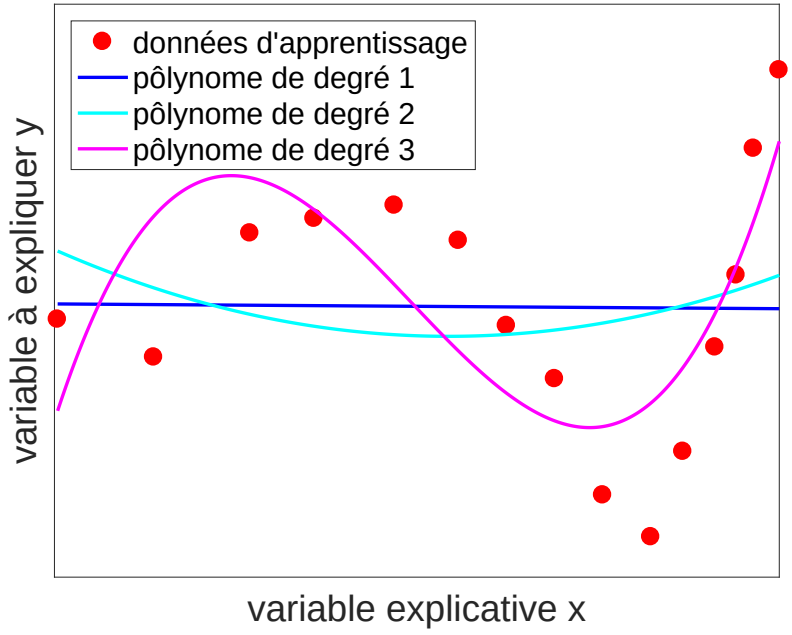
# Exemple d'apprentissage par cœur : l'interpolation

- définir un cout : $(predction - observation)^2$
  $(\quad f(x) \quad - \quad y \quad)^2$
- données 1d : $(x_i \in \mathbb{R}, y_i \in \mathbb{R}), i = 1, n$
- modèle : polynômes $f(x) = \sum_{j=0}^{d} \beta_j x^j$
- optimisation : méthode des moindres carrés



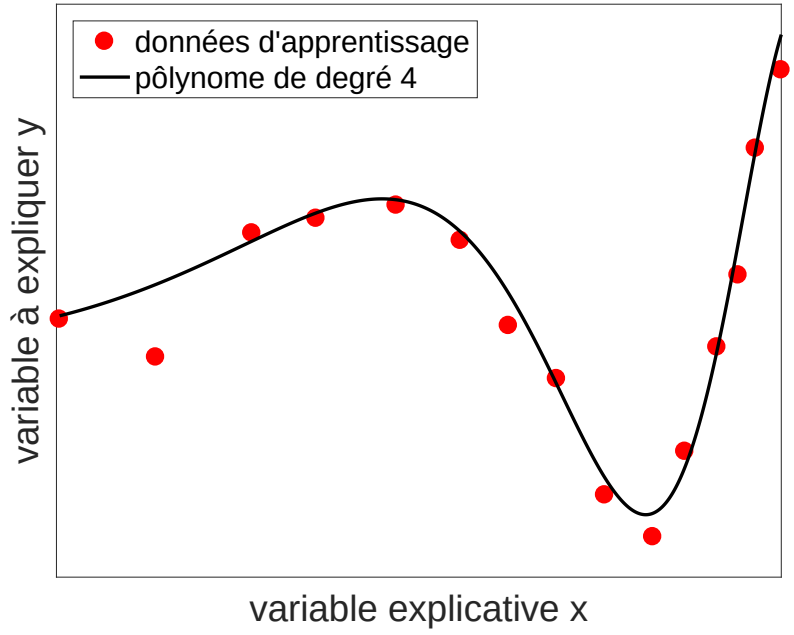
# Exemple d'apprentissage par cœur : l'interpolation

- définir un cout : $(predction - observation)^2$
  $(\quad f(x) \quad - \quad y \quad)^2$
- données 1d : $(x_i \in \mathbb{R}, y_i \in \mathbb{R}), i = 1, n$
- modèle : polynômes $f(x) = \sum_{j=0}^{d} \beta_j x^j$
- optimisation : méthode des moindres carrés

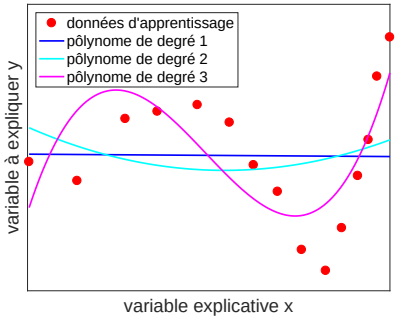


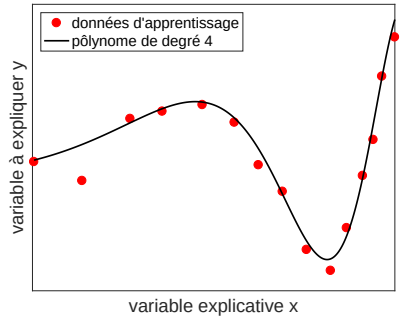
# Exemple d'apprentissage par cœur : l'interpolation

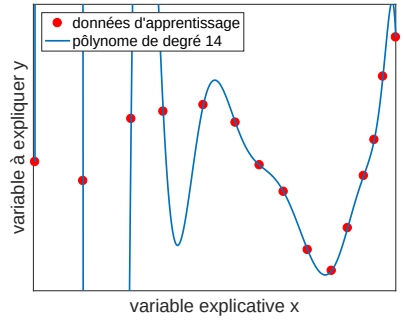
- définir un cout : $(predction - observation)^2$
  $(\quad f(x) \quad - \quad y \quad)^2$
- données 1d : $(x_i \in \mathbb{R}, y_i \in \mathbb{R}), i = 1, n$
- modèle : polynômes $f(x) = \sum_{j=0}^{d} \beta_j x^j$
- optimisation : méthode des moindres carrés

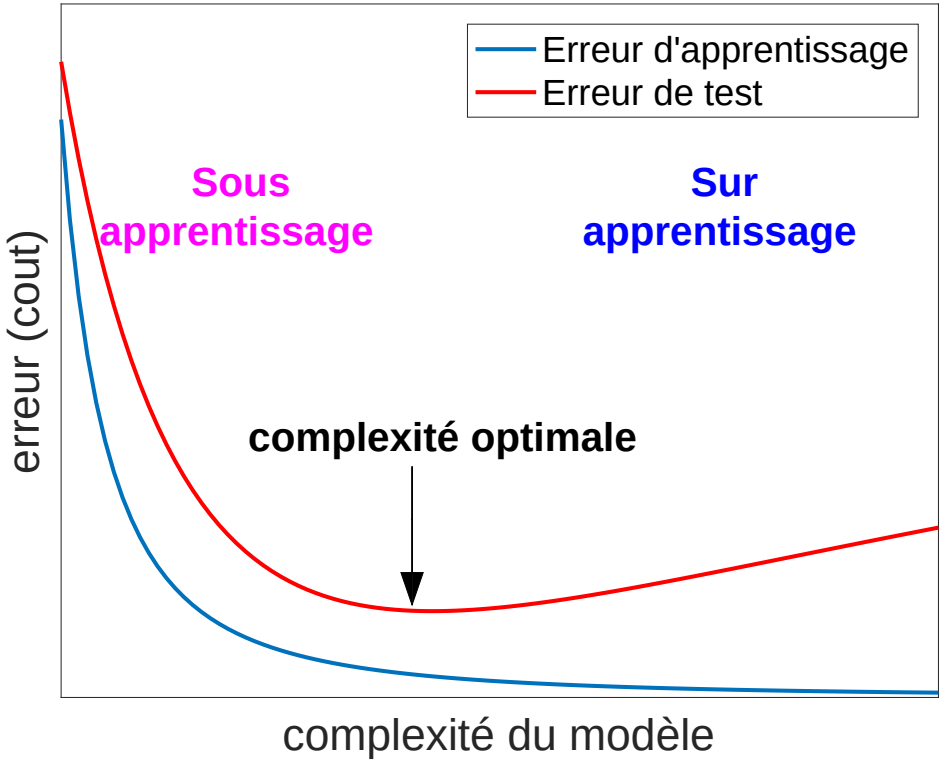


# Courbe d'apprentissage




Apprendre : optimisation bi objectif :

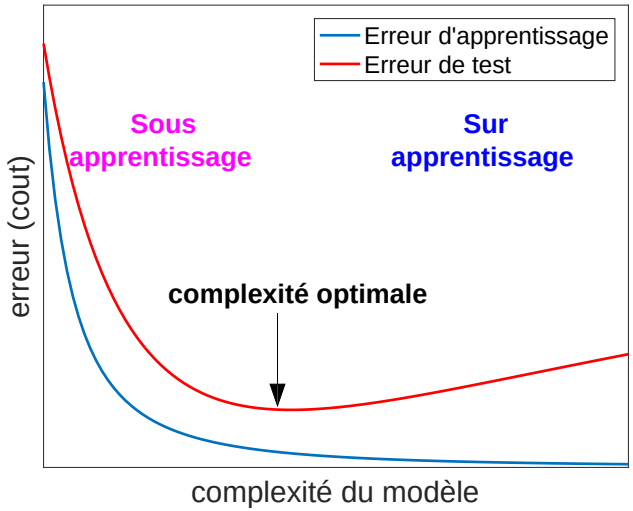
minimiser { l'erreur d'apprentissage
la complexité du modèle

**Compromis biais variance**

# Contrôle de la complexité : le principe du rasoir d'Occam

minimiser $\begin{cases} \text{l'erreur d'apprentissage} \\ \text{la complexité du modèle} \end{cases}$

« Pourquoi faire compliqué quand on peut faire simple »

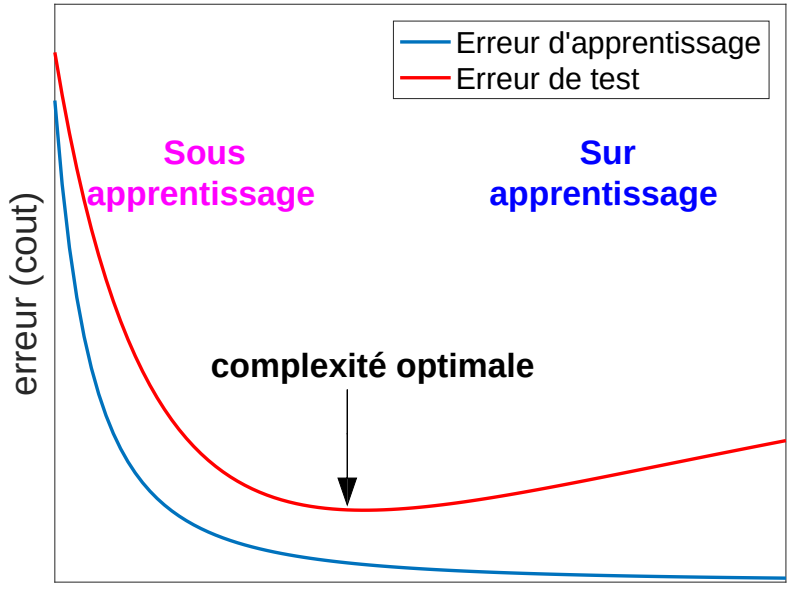

Erreur de généralisation = erreur d'apprentissage + risque de sur apprentissage

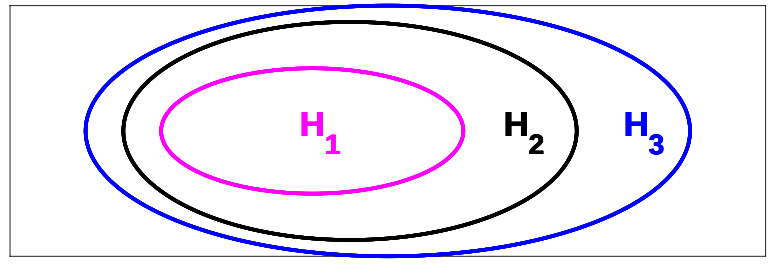
Risque de sur apprentissage = $\phi$(complexité du modèle)

$\rightarrow$ contrôle de la complexité du modèle

# Comment mesurer la complexité d'un modèle ?

- par les modèles :
  - nombre de modèles possibles
  - taille du modèle (# de paramètres)
  - régularité (taille effective) norme $\|f\|$
  - ensemble de modèles $\frac{1}{k}\sum_{k=1}^{m} f_k$
- par les données :
  - nombre de variables explicatives
  - injection de bruit $x_i + \varepsilon$ (data augmentation)
- par l'optimisation :
  - early stopping (semi convergence)
  - optimisation stochastique
- ...




**Ensembles de modèles emboités de complexité croissante**

$\rightarrow$ une famille (universelle) de modèles

# Quelle mesure de complexité choisir ?

la mesure de complexité introduit un biais :

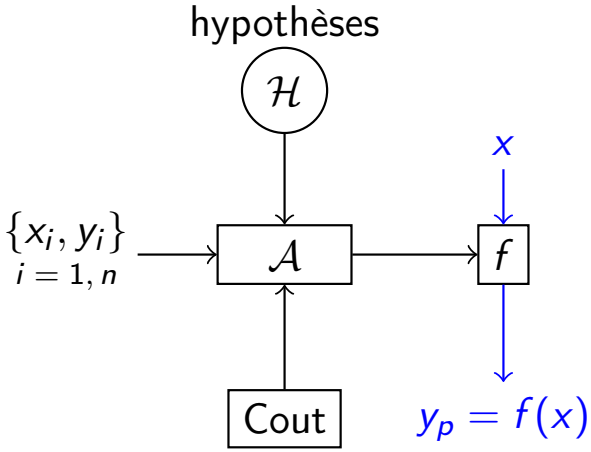
$\longrightarrow$ le biais inductif (ou biais d'apprentissage)


**No free luch theorem pour l'apprentissage (Wolpert 1996)**

Aucun algorithme d'apprentissage n'est intrinsèquement supérieur aux autres sur tous les problèmes.

Utilisons plusieurs mesures de complexité !

# Résumons nous. Apprendre c'est :




- programmation à base d'exemples
- être universel
  - une famille de modèles de complexité croissante
    $\rightarrow$ problème de définition de caractéristiques
  - quelle complexité choisir : *no free lunch*
    $\rightarrow$ utiliser plusieurs moyens de contrôle
- minimiser l'erreur d'apprentissage ET la complexité
  - minimiser un terme d'erreur attaché aux données
  - contrôler la complexité du modèle
    $\rightarrow$ problème de sélection de modèle
    $\rightarrow$ problème d'optimisation multi critère
  - être efficace
- disposer d'une ensemble de test (voir de validation)

Notre cerveau est plutôt bon pour généraliser !