Introduction à l'Apprentissage Automatique

Cécile Capponi, Sébastien Delecraz, Rémi Eyraud
(avec l'aide inestimable de l'équipe QARMA du LIS)

https://pageperso.lis-lab.fr/~remi.eyraud/WP/?page_id=190

L3 Informatique
2018-2019

Plan de cette intervention

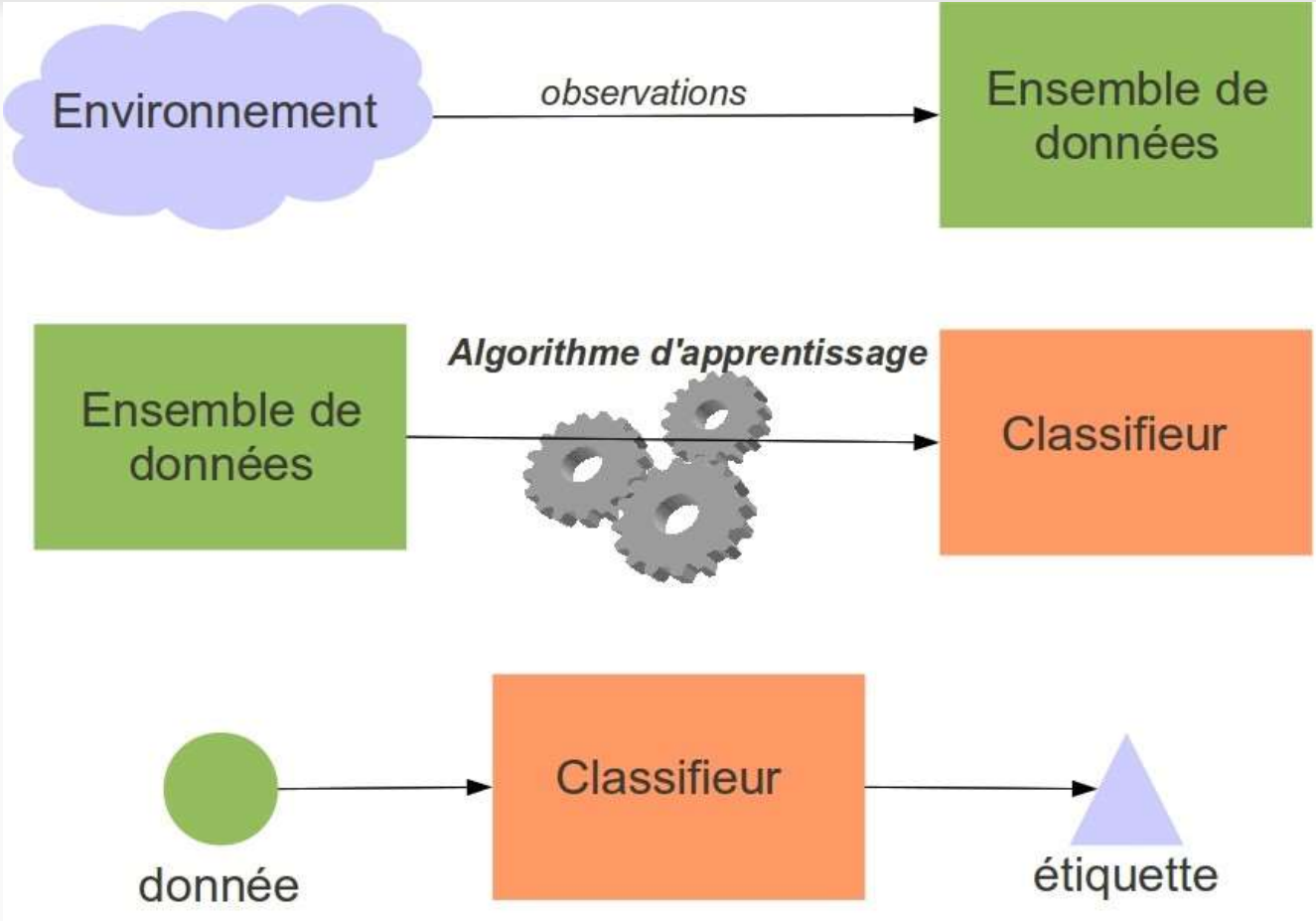
1. L’apprentissage, qu’est-ce que c’est ?
2. Protocole et mesure de qualité
3. Un exemple d’algorithme d’apprentissage
4. Déroulement de l’UE et évaluation

Qu’est-ce que l’apprentissage automatique

Apprentissage Automatique & Intelligence Artificielle & Science des Données

- 2019 : ces mots sont pratiquement synonymes
- Apprentissage machine :
  - Le **moteur** de la Science des Données
  - La partie aux succès récents et retentissants de l’IA

Schéma global



Apprentissage Automatique

- But : **extraire automatiquement** des données la connaissance permettant de prendre des bonnes décisions à l'avenir (sur d'autres / de nouvelles données)
- Moyen : inférer un modèle (mathématique…) qui capture les régularités (statistiques…) observables dans les **données d'apprentissage** : principe de **Généralisation**

Un exemple introductif

Alors que vous venez juste d'atterrir au Groland pour la première fois, vous apercevez un mouton noir. Quelles conclusions en tirer ?

Un exemple introductif

Alors que vous venez juste d'atterrir au Groland pour la première fois, vous apercevez un mouton noir. Quelles conclusions en tirer ?

- Il y a un et un seul mouton noir au Groland (apprentissage par coeur, sous-généralisation)
- Certains moutons sont noirs au Groland
- Tous les moutons du Groland sont noirs (sur-généralisation)

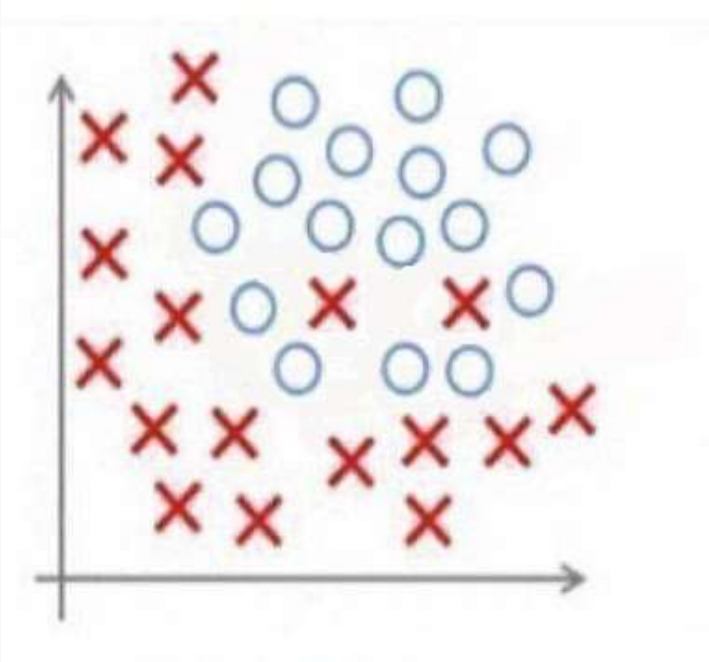
Un apprentissage particulier : La classification supervisée

Classification : Le but est d'apprendre une fonction qui associe une classe à une description de donnée.
Supervisée : pour chaque donnée d'apprentissage on connaît sa classe.

On connaît donc le nombre (fini) de classes et leur sémantique à l'avance.

La classification supervisée : Exemple simple

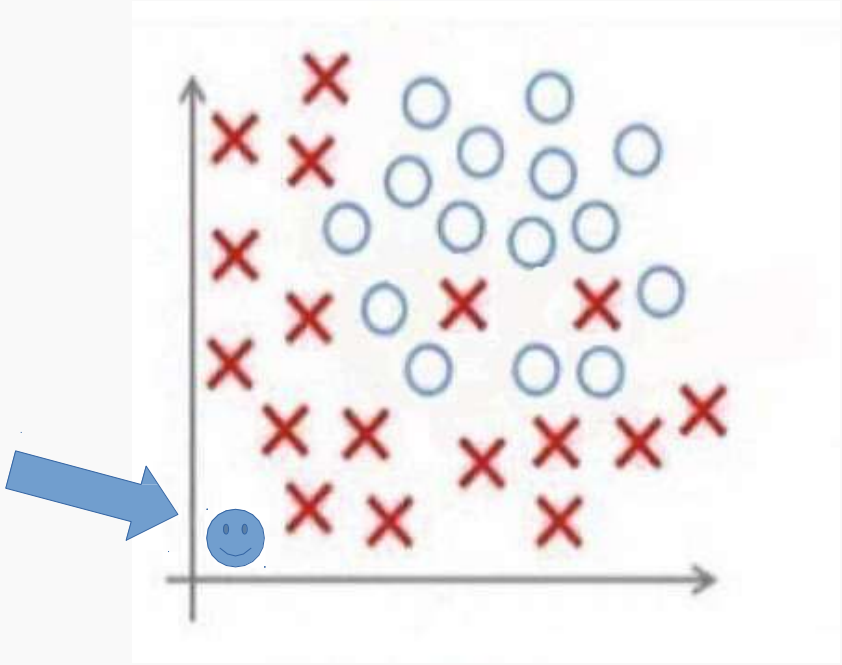
Données d'apprentissage :



La classification supervisée : Exemple simple

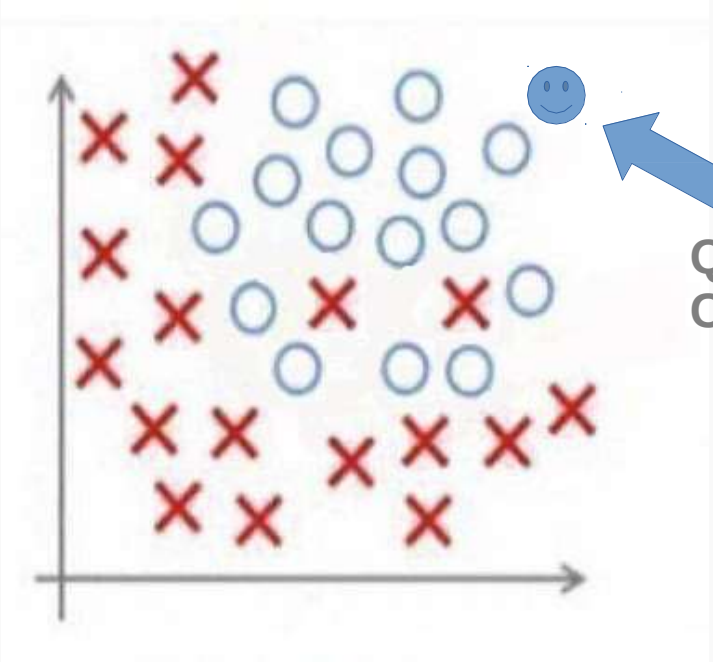
Nouvelle donnée
& Classification :

Quelle classse ?
Cercle ou croix ?



La classification supervisée : Exemple simple

Nouvelle donnée
& Classification :

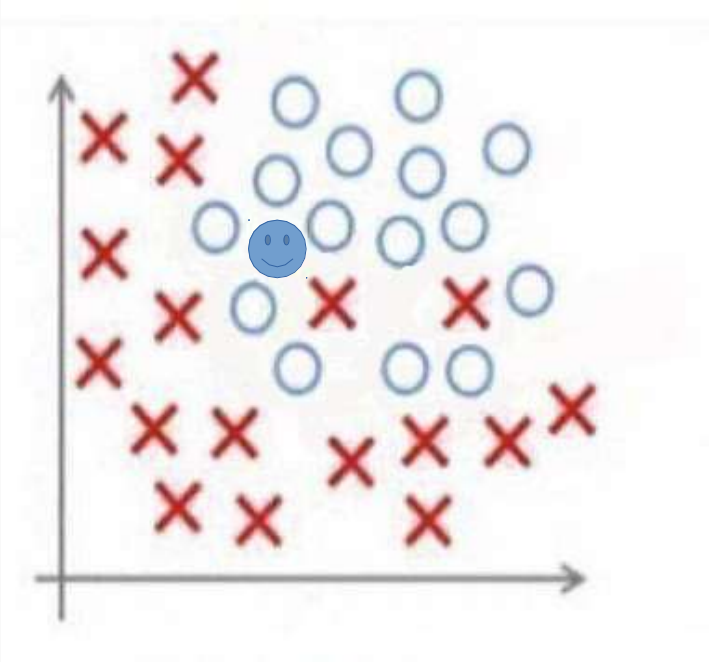


Quelle classse ?
Cercle ou croix ?

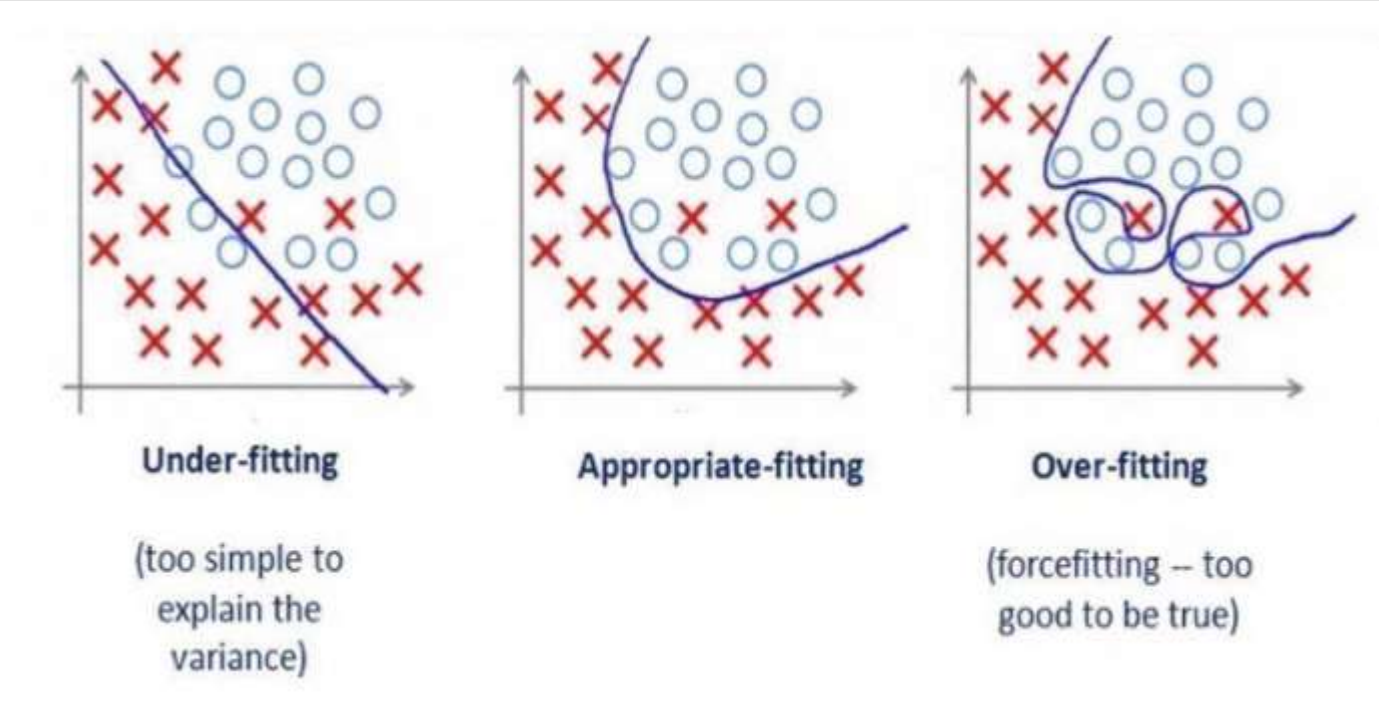
La classification supervisée : Exemple simple

Nouvelle donnée
& Classification :

Et pour cette donnée ?
Quelle classse ?
Cercle ou croix ?



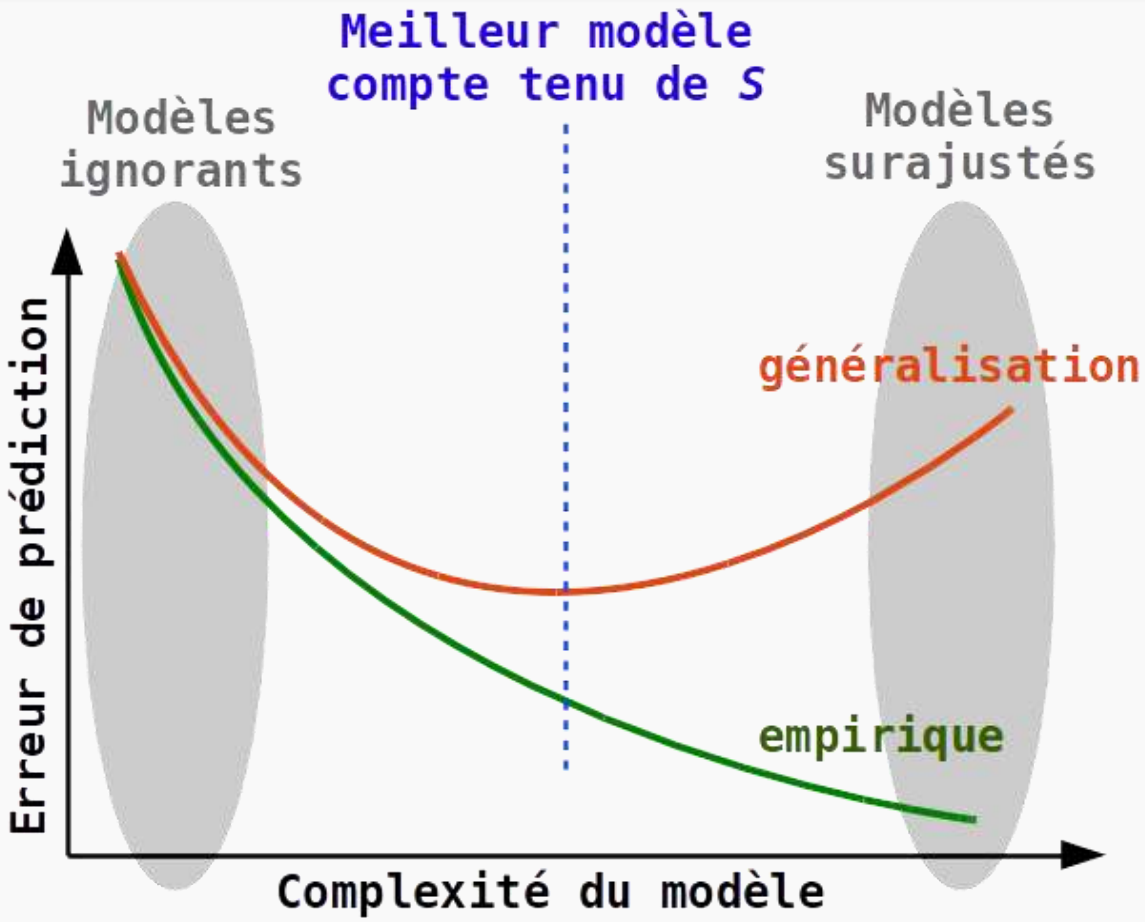
Sous-, Sur-, Correcte Généralisation



Erreur empirique et Généralisation

Erreur empirique : erreur sur l'échantillon d'apprentissage *S*

Erreur en généralisation : Erreur sur toutes les données (inconnue)



Classification supervisée : De vrais exemples

But : écarter automatiquement les SPAMs et autres messages non sollicités.
Données : des messages dont on sait s'ils sont des SPAMs ou non. Objectif : construire un classifieur, capable d'attribuer une de ces deux classes à un nouveau message.

But : reconnaissance de chiffres manuscrits.
Données : des chiffres écrits sur une rétine de 16x16 pixels, associés à une classe parmi {0, 1, ... , 9}
Objectif : attribuer la bonne classe (pattern recognition).

Modélisation de la classification supervisée

- **Attributs**(=variables=colonnes=features) :
  un ensemble **X** = $X_1 \times X_2 \ldots \times X_d$ où chaque X_i est le domaine d'un attribut A_i symbolique ou numérique.
  - Ex.: A_1= age, X_1= [0; 122], A_2= fumeur, X_2= {oui, non}
- **Classes**(=cible=target=label=etiquettes) :
  Un ensemble fini de classes Y.
  - Ex.: Y = {patient_à_risque, patient_sans_risque}
- Une **variable aléatoire** Z=(**X**,Y) à valeurs dans **X** x Y.
  - Ex: le risque cardiaque est lié à l'âge et au fait de fumer

Modélisation de la classification supervisée

- Les exemples/données sont des couples ($\mathbf{x}$,y) de $\mathbf{X} \times Y$ tirés selon la distribution jointe :

$$P(Z=(\mathbf{x},y)) = P(X=\mathbf{x})P(Y=y|X=\mathbf{x}).$$

- Un **échantillon S** = $\{(\mathbf{x}_1,y_1), \ldots , (\mathbf{x}_n,y_n)\}$ est un ensemble fini d'exemples i.i.d. selon P.

Modélisation de la classification supervisée

Exemple d'échantillon d'apprentissage

Ensemble des attributs
X = $X_1 \times X_2$ avec
X_1 = age et
X_2 = Fumeur

Classes Y
Valeurs possibles : {risque, pas risque}

Age	Fumeur	Risque cardio
35	oui	risque
40	non	pas risque
60	oui	risque
20	oui	pas risque

Classifieur

- **Classifieur** : $f : \mathbf{X} \rightarrow Y$
  - Ex.: f1(**x**) = Si fumeur='oui' et age > 59
    alors 'risque' sinon 'pas risque'

Fonction de perte (loss function) :

- $L(y_i, f(\mathbf{x}_i))$ égale à 0 si $y_i = f(\mathbf{x}_i)$ et à 1 sinon.
  - Ex.: L('risque', f1((35, 'oui'))) = 1
    et L('risque', f1((65, 'oui')) = 0

Modélisation de la classification supervisée (fin)

- Classifieur : $f : \mathbf{X} \rightarrow Y$
- Fonction de perte (loss function) :
  $L(y_i, f(\mathbf{x}_i))$ égale à 0 si $y_i = f(\mathbf{x}_i)$ et à 1 sinon

- La **fonction risque** (ou d'erreur) : espérance mathématique de la fonction de perte :

$$\mathbf{R(f)} = \int L(y,f(\mathbf{x}))dP(x,y) = \int_{y \neq f(\mathbf{x})} dP(\mathbf{x},y) = \mathbf{P(y \neq f(x))}$$

- Le **problème général de la classification supervisée** s'écrit :

Etant donné un échantillon $S=\{(x_1,y_1), \ldots, (x_n,y_n)\}$ trouver un classifieur f dont le risque $R(f)$ est le plus petit possible

Ce qu’il ne faut jamais oublier

- Le problème peut être non déterministe

  (= plusieurs classes y peuvent correspondre au même vecteur **x**)
- Les données peuvent être bruitées

  (= pas le bon y attribué à un **x** et/ou les différentes valeurs composant **x** ne sont pas bonnes)
- L’espace de description n’est pas suffisant pour décrire la complexité du problème auquel on s’attaque

  (= chaque **x** ne comporte pas suffisamment d’information pour permettre de trouver le y)

Un exemple d’algorithme d’apprentissage :

Le classifieur naïf de Bayes

Règle de Bayes et optimalité

- Si on cherche à classer **x**, quel est le meilleur y à lui attribuer ?

  Réponse : celui qui maximise P(y | **x**) ! C'est-à-dire le y le plus probable quand on connaît **x**.

  Mathématiquement : $f_{Bayes}(x) = \text{argmax}_y P(y \mid \mathbf{x})$

- C'est ce que l'on appelle la **règle de Bayes** et on peut prouver que c'est le meilleur classifieur possible (= celui dont le taux d'erreur en généralisation est le plus faible possible)

Tristesse immense & espoir infini

- La règle de Bayes n’est pas calculable à partir d’un ensemble de données.
- Le but de tout algorithme d’apprentissage est donc d’inférer un classifieur (presque) aussi bon que celui de la règle de Bayes.
- Bonne nouvelle : il existe des dizaines de (très) bons algorithmes pour faire ça ! Et il en reste très certainement de nouveaux à trouver.

Le classifieur naïf de Bayes

- La règle de Bayes peut se réécrire (à l'aide de la formule de Bayes) :

$$argmax_y P(y|\boldsymbol{x}) = argmax_y \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = argmax_y P(\boldsymbol{x}|y)P(y)$$

  - P(y) peut être estimé en calculant les fréquences de chaque classe dans l'échantillon d'apprentissage
  - Plus difficile pour P(**x**|y) (vraisemblance) : si x =(x_1, x_2, …, x_d) alors il nous faut connaître P({x_1, x_2, …, x_d}|y) ce qui n'est pas possible, en particulier si ces valeurs ne sont pas indépendantes.

Le classifieur naïf de Bayes (2)

- Le problème (pour d =2) :
  $P(\{x_1, x_2\}|y) = P(\{x_1\}|y)\ P(\{x_2\}|y, x_1)$ et $P(\{x_1, x_2\}|y) = P(\{x_2\}|y)\ P(\{x_1\}|y, x_2)$
- En général, ni $P(\{x_2\}|y, x_1)$ ni $P(\{x_1\}|y, x_2)$ ne sont pas (raisonnablement) estimables à partir d'un ensemble de données
- **Classifieur naïf de Bayes** : hypothèse (forte) d'indépendance des attributs :
  $P(\{x_2\}|y, x_1) = P(\{x_2\}|y,)$ et $P(\{x_1\}|y, x_2) = P(\{x_1\}|y)$
  En d'autre termes : les attributs sont indépendants deux à deux

Le classifieur naïf de Bayes (3)

- Si on met les étapes toutes ensembles, le classifieur naïf de Bayes est :

$$argmax_y P(y|\boldsymbol{x}) = argmax_y P(\boldsymbol{x}|y)P(y) \approx argmax_y P(y)\prod_{i=1}^{d} P(x_i|y)$$

  - P(y) est estimable à partir des données
  - $P(x_i \mid y)$ l'est aussi facilement :
    - Si x_i prend des valeurs discrètes, il suffit de calculer les fréquences des valeurs pour chaque classes
    - Si x_i prend des valeurs continues, on suppose que ces valeurs correspondent à une gaussienne et on évalue la moyenne et la variance à l'aide des données d'apprentissage.

Classifieur naïf de Bayes : Exemple

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	y
🔵	A	3	▶
🔵	B	3	♂
🔵	A	3	▶
🔵	A	2	♂
🔴	B	2	♂
🔴	A	3	▶
🔴	B	3	♂
🔴	B	2	▶
🔴	B	2	♂
🔴	A	2	♂
🟢	B	3	▶
🟢	B	2	♂

$P(▶)=5/12=0.42$ $P(A|▶)=3/5=0.6$
$P(♂)=7/12=0.58$ $P(B|▶)=2/5=0.4$
$P(🔵|▶)=2/5=0.4$ $P(A|♂)=2/7=0.29$
$P(🔴|▶)=2/5=0.4$ $P(B|♂)=5/7=0.71$
$P(🟢|▶)=1/5=0.2$ $P(3|▶)=4/5=0.8$
$P(🔵|♂)=2/7=0.29$ $P(2|▶)=1/5=0.2$
$P(🔴|♂)=4/7=0.57$ $P(3|♂)=2/7=0.29$
$P(🟢|♂)=1/7=0.14$ $P(2|♂)=5/7=0.71$

$P(▶|🟢\,A\,3) => P(🟢\,A\,3|▶)P(▶)=P(🟢|▶)P(A|▶)P(3|▶)P(▶)=0.04$
$P(♂|🟢\,A\,3) => P(🟢\,A\,3|♂)P(♂)=P(🟢|♂)P(A|♂)P(3|♂)P(♂)=0.006$

Classifieur naïf de Bayes en python

```
# Ceci est un commentaire – importation de la fonction load_iris de la librairie scikit-learn :
from sklearn.datasets import load_iris  # récupération des données Iris  donnees = load_iris()
# Stockage de la matrice de description des données :
X = donnees.data
# Stockage des classes de chaque donnée :
y = donnees.target
# importation de la fonction découpant les données en test et train :
from sklearn.model_selection import train_test_split
# Génération aléatoire des échantillons (X_train, y_train) et (X_test, y_test) :
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

Classifieur naïf de Bayes en python

```
# importation de la librairie pour le classifieur naive Bayes :
from sklearn.naive_bayes import GaussianNB  # Création d'une instance de ce classifieur : classifieur = GaussianNB()
# Apprentissage sur les données d'entrainement :
classifieur.fit(X_train, y_train)
# Utilisation du classifieur appris sur les données de test :
y_predits = classifieur.predict(X_test)
# Calcul du taux de réussite :
from sklearn.metrics import accuracy_score
print("Taux de réussite : ", accuracy_score(y_test,y_predits))
```

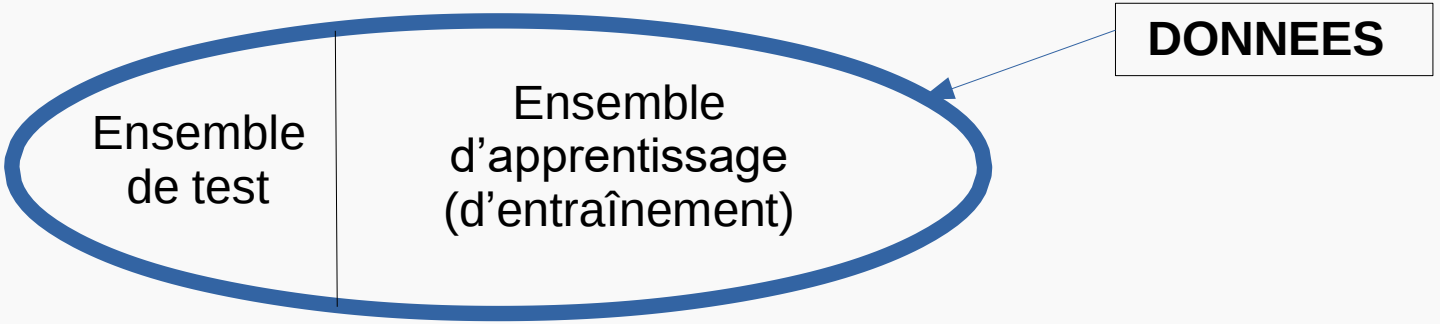
Validation d'un apprentissage
Mesures de qualité

L’apprentissage en pratique

- On dispose d'un échantillon d'apprentissage S qu'on suppose i.i.d.
- On recherche une fonction *h* de classification dont le risque est le plus faible possible.
- Il existe toujours une fonction f_{min} de risque minimal... inaccessible !

Validation empirique d’un apprentissage

- Plusieurs méthodes permettent de valider (ou d'infirmer) la qualité d'un processus d'apprentissage.
- Une des approches consiste à n'utiliser qu'une partie des données pour apprendre et à se servir des autres données pour tester le résultat.
- Différentes mesures permettent alors de comparer des processus (taux d’erreur, F-score, etc.)



Validation croisée (cross-validation)

- La **cross-validation** est une généralisation de la méthode précédente.
- Elle consiste à diviser les données en *c* folders, à en enlever un pour l'apprentissage puis à l'utiliser pour la
- phase de test.
- Le processus est ensuite réitéré sur chaque folder

L'erreur moyenne tend alors vers l'erreur en généralisation (estimateur non-biaisé).

Mesure de qualité : classification binaire supervisée

- **Matrice de confusion**
  :

	Classé +	Classé -
Exemple +	V_p	F_n
Exemple -	F_p	V_n

- Taux d'erreur = 1 - taux de réussite (*accuracy*) = $\frac{F_p + F_n}{F_p + F_n + V_p + V_n}$

- L'erreur (ou taux d'erreur) ne fait pas de distinction entre les erreurs : pas toujours une bonne mesure de qualité d'un apprentissage