# ENSEA
# 19/03/2015
# Introduction à l'apprentissage machine

*Geoffroy.Peeters@ircam.fr*
*UMR SMTS IRCAM CNRS UPMC*

## Table des matières

# 1 Introduction

## 1.1 Qu'est que l'apprentissage machine ?

- un des champs d'étude de l'intelligence artificielle
- la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage
- permet de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques

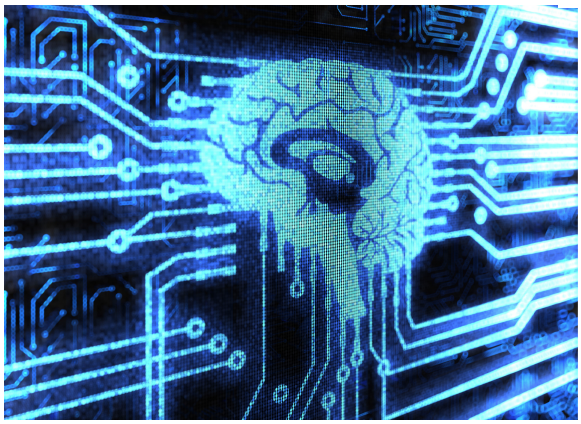


*Figure : source : http ://sira-corp.com/new/MachineLearning*

**Exemple : Reconnaissance de caractère.** Comment reconnaître des caractères manuscrits ?

- par énumération de règles
  - si intensité pixel à la position ... alors c'est un "3"
  - long et fastidieux, diffiçile de couvrir tous les cas
- en demandant à la machine d'apprendre
  - lui laisser faire des essais et apprendre de ses erreurs
  - → apprentissage machine (machine-learning)

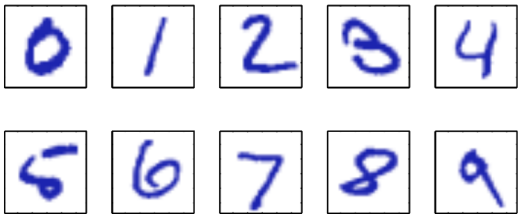


*Figure : source : Hugo Larochelle*

### Comment ça marche ?

- On donne à l'algorithme des données d'**entrainement**
- l'algorithme d'apprentissage machine **apprend** un **modèle** capable de généraliser à de nouvelles données

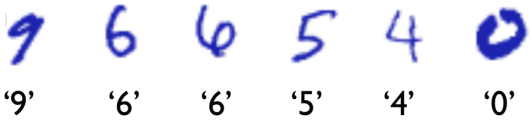



*Figure :*

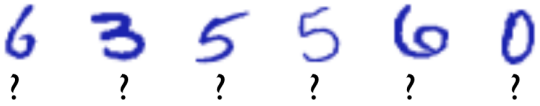
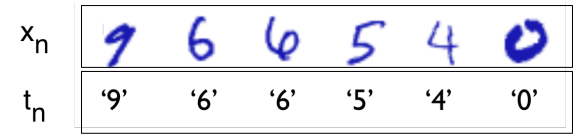



*Figure : source : Hugo Larochelle*

### Notations

- On appel **ensemble d'entraînement** :
  - $\mathbb{D}_{train} = \{(x_1, t_1), \cdots, (x_N, t_N)\}$
    - $x_n$ une **observation** (entrée du système) et
    - $t_n$ la **cible** correspondante (sortie du système)
- L'apprentissage machine fournit un **modèle** $y(x)$ qui prédit $t$ en fonction de $x$ : $y(x) = \hat{t}$
- L'objectif est de trouver un modèle tel que $y(x_n) = \hat{t}_n \simeq t_n$
- On mesure la qualité de l'apprentissage (la qualité du modèle) sur un **ensemble de test** :
  - $\mathbb{D}_{test} = \{(x_{N+1}, t_{N+1}), \cdots, (x_{N+M}, t_{N+M})\}$



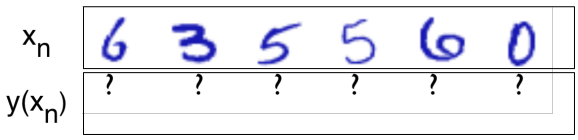




*Figure : source : Hugo Larochelle*

## 1.2 Deux grands types d'apprentissage machine

**Apprentissage supervisé** Nous considérons un ensemble d'**observations** (entrées du système) $\{x_1, \cdots, x_N\}$
- Nous donnons également à la machine les **cibles** (sorties du système) souhaitées $\{t_1, \cdots, t_N\}$
- $\mathbb{D}_{train} = \{(x_1, t_1), \cdots, (x_N, t_N)\}$
- L'objectif de la machine est d'apprendre les cibles (sorties) correctes pour de nouvelles observations (entrées)

**Apprentissage non-supervisé** Nous considérons un ensemble d'**observations** (entrées du système) $\{x_1, \cdots, x_N\}$
- Nous ne donnons pas à la machine les cibles
- $\mathbb{D}_{train} = \{x_1, \cdots, x_N\}$
- L'objectif de la machine est de créer un modèle de $x$, un partitionnement (clustering) des données
  - Utilisation ? analyse de données, prise de décisions

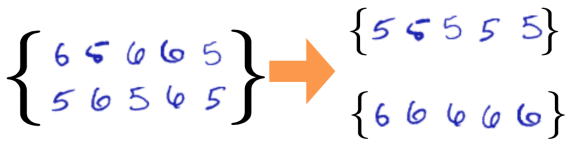


*Figure : source : Hugo Larochelle*

## 1.3 Deux grandes cibles pour l'apprentissage supervisé

**La régression :**
- La cible est un **nombre réel** : $t_n \in \mathbb{R}$
- Exemples :
  - Economie (prédiction de valeur en bourse) :
    - $x$= information sur l'activité économique de la journée, $t$= la valeur d'une action demain
  - Audio (reconnaissance de musique) :
    - $x$= le contenu spectral du signal, $t$= la hauteur en Hz d'une note

**La classification**
- La cible est un **indice de classe** : $t_n \in \{1, \cdots, C\}$
- Exemples :
  - Image (reconnaissance de caractères) :

  - $x$= vecteur d'intensité des pixels, $t$= l'identité du caractère
- Audio (reconnaissance de parole) :
  - $x$ : le contenu spectral du signal audio, $t$= le phonème prononcé

**Exemple d'apprentissage supervisé en musique : reconnaissance du genre**

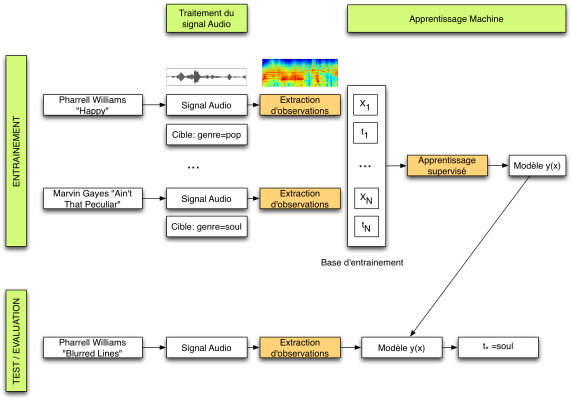



*Figure :*

**Exemple d'apprentissage non-supervisé en musique : regroupement de morceaux par similarité de contenu**

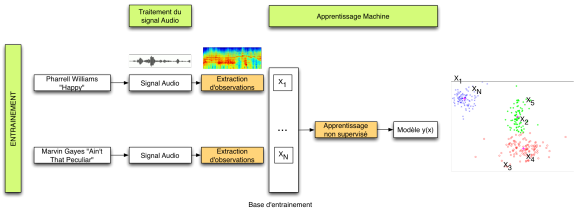



*Figure :*

# 2 Régression

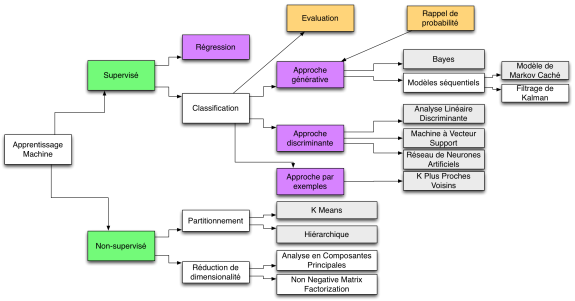



*Figure :*

## 2.1 Régression polynomiale en $D = 1$ dimension

On cherche à prédire une cible $t_n$ qui est un **nombre réel** : $t_n \in \mathbb{R}$

- Données :
  - entrée (observation) : $x$
  - sortie (cible) : $t \in \mathbb{R}$
- Objectif :
  - prédire $t$ en fonction de $x$ : $\hat{t} = y(x)$
  - $y$ est notre modèle que nous devons apprendre à partir de l'ensemble d'entraînement $\mathbb{D}_{train} = \{(x_1, t_1), \cdots, (x_N, t_N)\}$

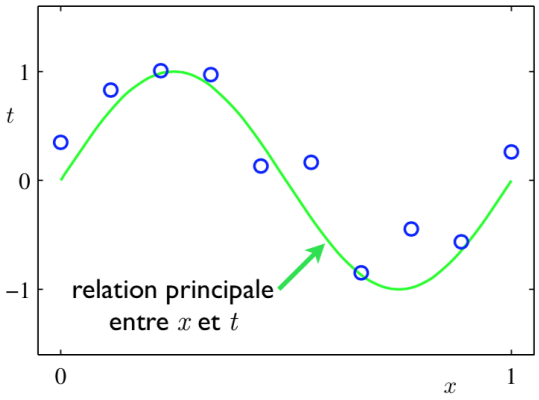



*Figure : source : Hugo Larochelle*

**Forme du modèle $y$ ?**

- nous lui imposons une forme de polynôme d'ordre $M$ de coefficients $w_m$

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M$$
$$= \sum_{m=0}^{M} w_m x^m$$

**Estimation du modèle = estimation des coefficients $w_m$ :**

- on note $\boldsymbol{w}$ le vecteur des coefficients $[w_1, \cdots, w_M]$
- on cherche le vecteur $\boldsymbol{w}$ tel qu'il minimise l'erreur de prédiction sur l'ensemble d'entraînement $\mathbb{D}_{train}$
  - Erreur de prédiction sur une donnée $t_n$ :
    - $\epsilon(\boldsymbol{w}, n) = (y(x_n, \boldsymbol{w}) - t_n)^2$
  - Erreur de prédiction totale (sur l'ensemble des données d'entraînement) :
    - $E(\boldsymbol{w}) = \frac{1}{N} \sum_{n=1}^{N} \epsilon(\boldsymbol{w}, n)$

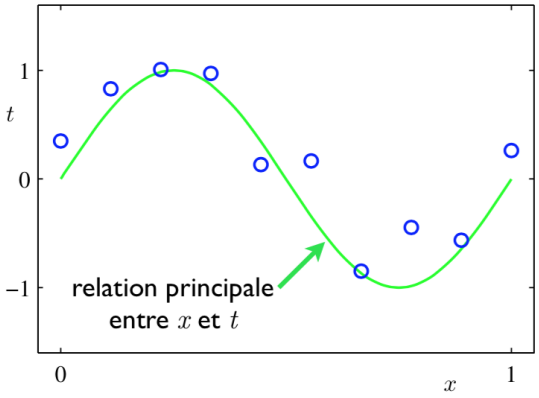



*Figure : source : Hugo Larochelle*

**Comment choisir l'ordre du polynôme $M$ ?**

- si $M$ est trop petit :
  - on modélise mal les données, grande perte sur l'ensemble d'entraînement
  - → **sous-apprentissage**
- si $M$ est trop grand :
  - on apprend "par coeur" de l'ensemble d'entraînement
  - → **sur-apprentissage**

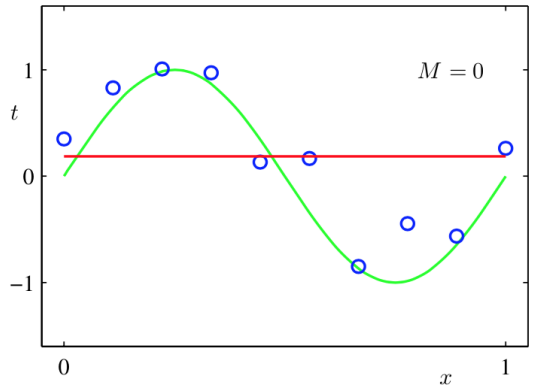



*Figure : Exemple de sous-apprentissage pour la régression (source : Hugo Larochelle)*

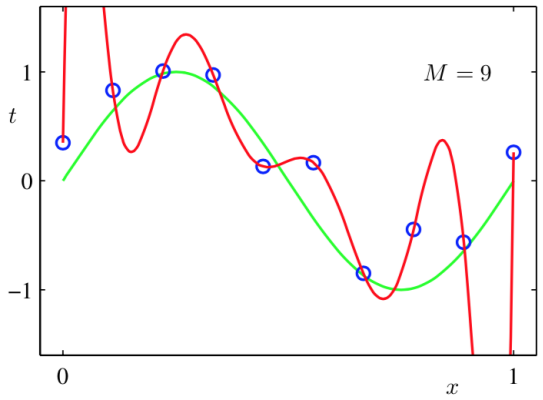



*Figure : Exemple de sur-apprentissage pour la régression (source : Hugo Larochelle)*

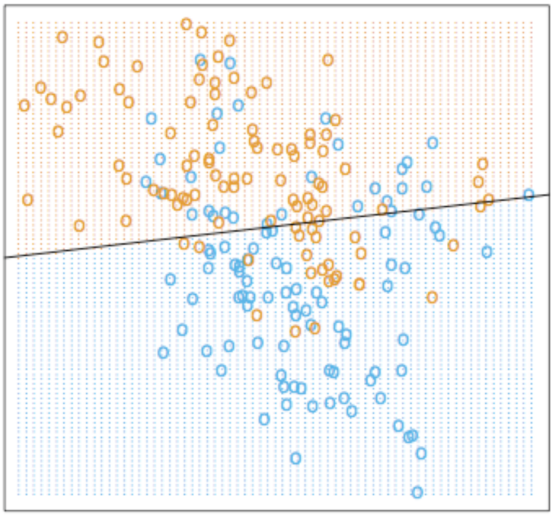


*Figure : Exemple de sous-apprentissage pour la classification (source : Arshia Cont)*

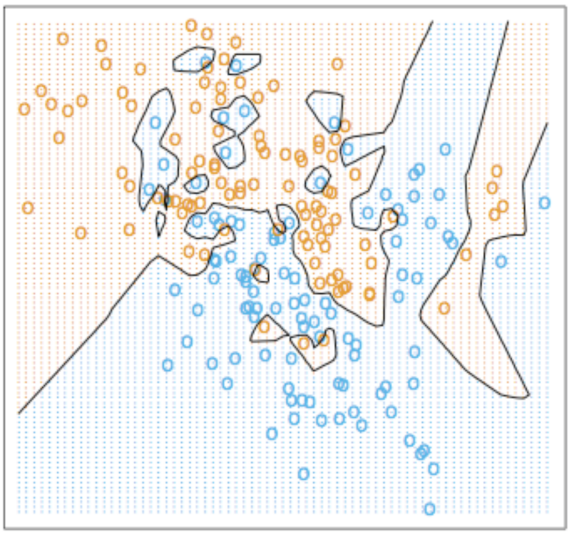


*Figure : Exemple de sur-apprentissage pour la classification (source : Arshia Cont)*

### Généralisation

- On cherche une valeur de $M$ qui permet de retrouver la **tendance générale** de la relation entre $x$ et $t$
  - sans apprendre le bruit

- va permettre de généraliser à de nouvelles données

## 2.2 Régression polynomiale en $D > 1$ dimension

- $D$ dimensions ?
  - le nombre de dimension de l'observation $x$
  - exemple :
    - $D = 1$ : on considère l'intensité globale d'une image
    - $D = 64$ : on considère l'intensité de chaque pixel d'une image de (8,8)
- Pour un polynôme d'ordre $M = 3$ et $D = 1$ **dimensions** : $x = x_i$ :

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Pour un polynôme d'ordre $M = 3$ et $D = 3$ **dimensions** : $\boldsymbol{x} = [x_i, x_j, x_k]$ :

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

  - on a $1 + D + D^2 + D^3$ paramètres à estimer
  - pour $D$=100, $M$=3, on aurait un million de paramètres à estimer !!!

### Malédiction de la dimension

- Pour $D$=100, $M$=3, on aurait un million de paramètres à estimer !!!
- Pour pouvoir garantir qu'on va bien généraliser à une nouvelle entrée $x$, il faut avoir des entrées similaires à $x$ dans l'ensemble d'entraînement
  - Au plus le nombre de dimension $D$ augmente, au plus il devient difficile d'avoir des entrées similaires à $x$
- Preuve (interprétation géométrique) :
  - on divise également l'espace des observations en régions (hyper-cubes)
  - quand $D$ augment, le nombre de régions augmente en $O(3^D)$
    - Il devient impossible de garantir qu'on aura bien un exemple dans chaque région !!!

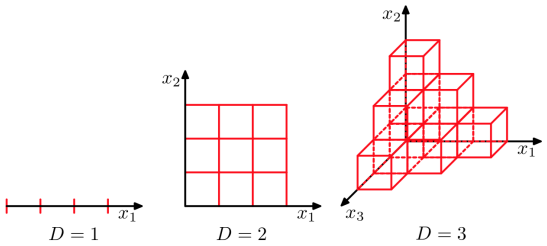



*Figure : source : Hugo Larochelle*

# 3 Apprentissage supervisé

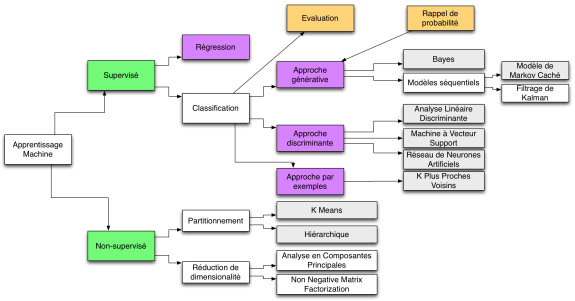



*Figure :*

## 3.1 Exemple : Système de communication

Imaginons un système de communication dont l'entrée est $Y$ et la sortie $X$.

- on observe uniquement la sortie $X$
- on souhaite retrouver $Y$ (non-observable) à partir de $X$
  - $\rightarrow$ on **infère** $Y$ à partir de $X$

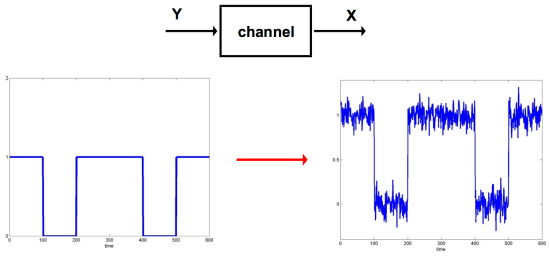



*Figure : source : Arshia Cont*

**Solution 1 : Approche générative :**

- On **apprend** la fonction qui **génère**
  - les valeurs de $X$ quand $Y = 0$ : $P(X|Y = 0)$
  - les valeurs de $X$ quand $Y = 0$ : $P(X|Y = 1)$
- On en déduit la probabilité que $P(Y = 0|X)$ et $P(Y = 1|X)$
- On décide que $Y = 0$ si $P(Y = 0|X) > P(Y = 1|X)$
- Ceci conduit à une **fonction de décision** $g(x)$
  - Cette fonction de décision est une conséquence des modèles génératifs

En résumé :

- nous partons de l'hypothèse qu'il existe une famille de modèles paramétriques permettant de générer $X$ connaissant $Y$
  - Exemple : apprentissage Bayesien, modèle de Markov caché, réseaux de neurones artificiels

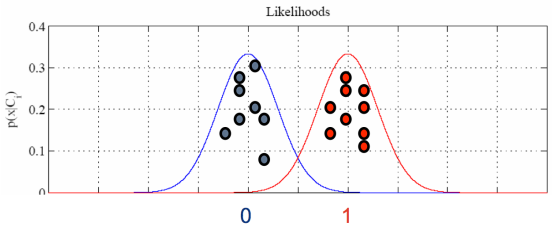



*Figure : source : Arshia Cont*

**Solution 2 : Approche discriminante :**

- On apprend directement **la fonction de décision** $g(x)$ qui sépare le mieux
  - les valeurs de $X$ correspondant à $Y = 0$ et
  - les valeurs de $X$ correspondant à $Y = 1$
- On ne considère pas la manière dont $X$ est généré à partir de $Y$ !!!

En résumé

- nous n'avons pas d'hypothèse sur le modèle sous-jacent à $X$ mais nous étudions comment séparer ses valeurs
  - Exemple : analyse linéaire discriminante, machine à vecteur support (SVM)

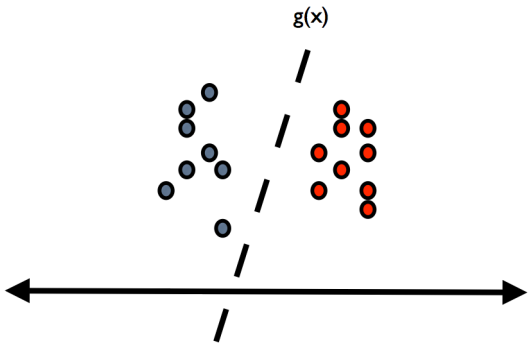



*Figure : source : Arshia Cont*

**Solution 3 : Approche par exemplification :**

- On possède une série d'exemples de couples assignant une observation $X$ à une cible $Y$ : $\mathbb{D}_{train} = \{(x_1, y_1), \cdots, (x_N, t_N)\}$
  - pour une nouvelle observation $x^\star$, on cherche les observations $X$ de la base d'entrainement les plus proches de $x^\star$,
  - on assigne à $x^\star$ le $y$ correspondant aux $X$ les plus proches
  - Exemple : K-plus-proche-voisin

## 3.2 Généralisation

### Apprentissage

- Apprendre un modèle (génératif ou discriminant) à partir des observations $X$ et des valeurs à prédire $Y$
- Le modèle doit permettre une bonne prédiction de $Y$ en fonction des observations $X$

### Généralisation

- Capacité du modèle à prédire correctement des valeurs $Y^\star$ en fonction de $X^\star$ en dehors de l'ensemble d'apprentissage
- Sur-apprentissage (over-fitting)
- En pratique on évalue les performances d'un modèle appris en séparant :
  - Ensemble d'entrainement (training-set) : $\{X, Y\}$
  - Ensemble de test (test-set) : $\{X^\star, Y^\star\}$

# 4 Approche générative

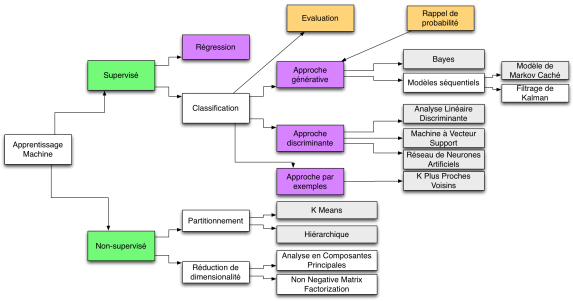



*Figure :*

## 4.1 Décision Bayesienne

- On souhaite trouver la classe $Y = \{0, 1\}$ en fonction de $x$
- En l'absence d'information, les deux probabilités sont équiprobable :
  - $p(Y = 0) = p(Y = 1) = 0.5$
  - $p(Y)$ est appelé **probabilité a priori** (prior)
- On considère que $X$ a été généré (modèle génératif) par $Y$ : $p(X|Y)$
  - plus précisément on considère que $X$ est une version bruitée de $Y$
    - $X = Y + \epsilon$
    - ou $\epsilon$ est un bruit de moyenne nul et de variance $\sigma^2$ : $\epsilon \sim \mathcal{N}(0, \sigma)$
  - on peut donc modéliser $X$ comme
    - $P(X|Y = 0) \sim \mathcal{N}(0, \sigma)$
    - $P(X|Y = 1) \sim \mathcal{N}(1, \sigma)$
  - $P(X|Y)$ est appelé **vraisemblance** (likelihood)

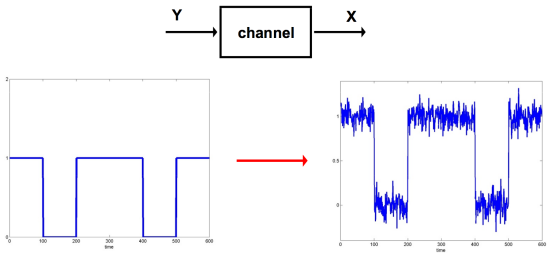



*Figure :*

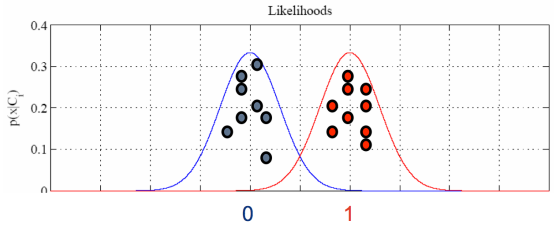



*Figure : source : A. Cont*

**Comment choisir la meilleure classe $Y = 0$ ou $1$ ?**

- Méthode du **Maximum A Posteriori** (MAP)
  - on choisi la classe dont la probabilité a posteriori est maximale
  - $i^\star(x) = \arg\max_i\{p(Y = y_i|X)\}$
- Problème :
  - on ne connait pas $p(Y = y_i|x)$, mais on connait $P(X|Y = y_i)$
  - comment passer de l'un à l'autre $\rightarrow$ inférence Bayesienne

**Inférence Bayesienne**

- permet de mettre à jour les informations à **priori** pour créer les informations à **posteriori** en fonction des informations que nous avons sur $X$ (**vraisemblance**)
- Rappel :
  - $P(X, Y) = P(Y|X)P(X) = P(X|Y)P(Y)$
- Inférence Bayesienne :

$$p(Y = y_i|x) = p(Y = y_i) \cdot \frac{p(x|Y = y_i)}{p(x)}$$

$$\textbf{posterior} = \textbf{prior} \cdot \frac{\textbf{vraisemblance}}{\textbf{evidence}}$$

**Comment choisir la meilleure classe $Y = 0$ ou $1$ ?**

- Méthode du **Maximum A Posteriori** (MAP) :
  - si on omet le dénominateur $p(x)$ (puisqu'il qui est le même pour toutes les classes)
  - $i^\star(x) = \arg\max_i\{p(Y = y_i)p(x|Y = y_i)\}$
- Pour le calcul on considère les log-probabilités
  - $i^\star(x) = \arg\max_i\{\log(p(Y = y_i)) + \log(p(x|Y = y_i))\}$

### 4.1.1 Inférence Bayesienne dans le cas Gaussien : * Cas Simple*

- Cas Simple ?
  - on considère que toutes les classes sont **a priori** equi-probables : $P(Y = 0) = P(Y = 1) = 0.5$
    - alors $i^\star(x) = \arg\max_i\{\log(p(x|Y = y_i))\}$
  - on considère que $X$ a une dimension ($D = 1$) et que les variances sont les mêmes ($\sigma = \sigma_0 = \sigma_1$)
    - alors $P(X|Y = 0) \sim \mathcal{N}(\mu_0, \sigma)$ et
    - alors $P(X|Y = 1) \sim \mathcal{N}(\mu_1, \sigma)$
- On peut montrer que
  - $i^\star(x) = \arg\min_i(w_i x + w_{i0})$
    - avec $w_i = -2\mu_i$
    - avec $w_{i0} = \mu_i^2$
- On peut montrer que les classes $i$ et $j$ sont **équi-probables** pour $x$ tel que
  - $-2\mu_0 x + \mu_0^2 = -2\mu_1 x + \mu_1^2$
- Ces valeurs de $x$ définissent une **frontière de décision** $g(x)$
  - $g(x) = \frac{\mu_1+\mu 2}{2}$

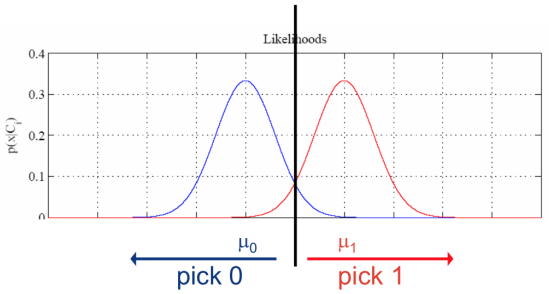



*Figure : source : Arshia Cont*

**Illustration de l'influence de la probabilité a priori quand $D = 1$**

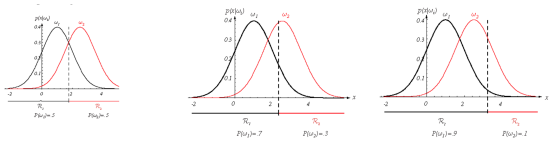


*Figure : source : Arshia Cont*

### 4.1.2 Inférence Bayesienne dans le cas Gaussien : * Cas général *

- Cas Général ?

  - on considère $P(Y = 0) \neq P(Y = 1)$
  - on considère que $X$ a plusieurs dimensions ($D > 1$) et que les matrices de co-variances sont égales $\mathbf{\Sigma} = \mathbf{\Sigma}_0 = \mathbf{\Sigma}_1$)
    - alors $P(X|Y = 0) \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma})$ et
    - alors $P(X|Y = 1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{\Sigma})$
- On peut montrer que
  - $i^\star(x) = \arg\min_i\{\boldsymbol{w}_i^T\boldsymbol{x} + \omega_{i0}\}$
    - avec $\boldsymbol{w}_i = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i$
    - avec $\omega_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \log(P(Y = i))$
- On peut montrer que les classes $i$ et $j$ sont **equi-probables** pour $\boldsymbol{x}$
  - $\boldsymbol{w}_i^T\boldsymbol{x} + \omega_{i0} = \boldsymbol{w}_j^T\boldsymbol{x} + \omega_{j0}$
- Ces valeurs de $\boldsymbol{x}$ définissent la **frontière de décision**
  - $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$
    - avec $\boldsymbol{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$
    - avec $w_0 = -\frac{(\boldsymbol{\mu}_i+\boldsymbol{\mu}_j)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}{2} + \log\frac{P(Y=i)}{P(Y=j)}$

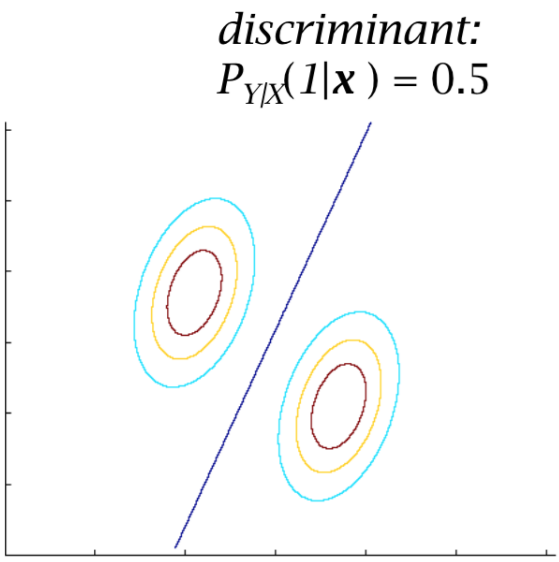



*Figure : source : Arshia Cont*

**Illustration de l'influence de la probabilité a priori quand** $D = 2$

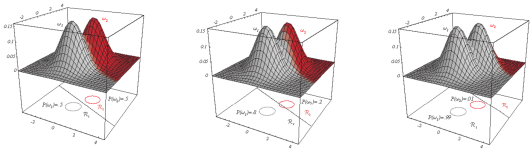


*Figure : source : Arshia Cont*

**Illustration de l'influence de la probabilité a priori quand $D = 3$**

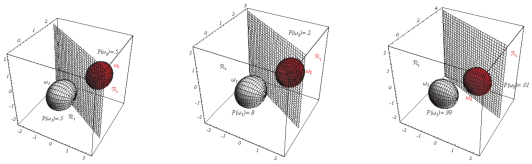


*Figure : source : Arshia Cont*

### 4.1.3 Conclusion

- Dans l'approche générative, nous apprenons les modèles $P(X|Y)$ ayant engendré $X$ en fonction de $Y$
- Nous utilisons l'information a priori sur les classes $Y$ $P(Y)$
- Nous combinons les deux pour obtenir une probabilité a posteriori $P(Y|X)$ en utilisant la loi de Bayes
- La conséquence de cette modélisation est une frontière de décision entre classes $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$

**Approche discriminante**

- Nous apprenons directement la frontière $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$ qui permet le mieux de séparer les classes $Y$
- Nous ne faisons pas d'hypothèse sur le modèle ayant engendré $X$

## 4.2 Modèles séquentiels

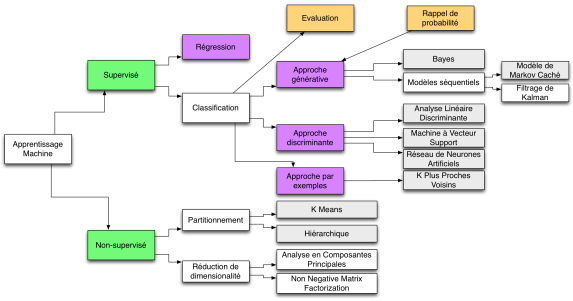


*Figure :*

### 4.2.1 Modèles de Markov cachés

- Andreï A. Markov (1856-1922) : un mathématicien Russe
- Chaîne de Markov :
  - un processus stochastique à **temps discret** $t$ pouvant être dans des **états discrets** $S \in [1, \cdots, I]$
  - $S_t$ la valeur de l'état à l'instant $t$
- Chaîne de Markov d'ordre 1 :
  - la prédiction de l'état actuel $t$ ne dépend que de l'instant précédent $t-1$ :
    - $p(S_t|S_{t-1}, S_{t-2} \cdots S_0) = p(S_t|S_{t-1})$

*Figure :*

**Exemple : Doudou le hamster**

- Doudou le hamster à 3 états dans sa journée :
  - soit il dort : il est dans l'état $S_1$ (copeaux)

  - soit il mange : il est dans l'état $S_2$ (mangeoire)
  - soit il fait du sport : il est dans l'état $S_3$ (roue)
- On peut représenter la succession de ces états par une matrice de transition entre états

$$T_{ij} = P(S_{t+1} = j, S_t = i) = \begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{pmatrix}$$

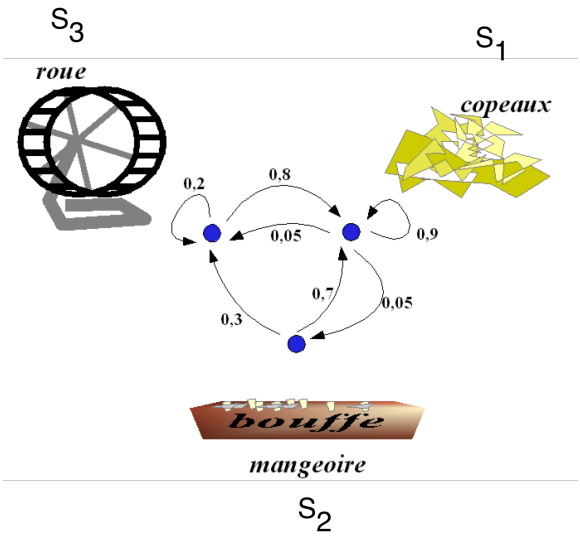



*Figure : Modèle de Markov d'une journée de Doudou le hamster*

## Modèle de Markov caché ?

- Dans un modèle de Marlov caché on observe pas directement les états $S_i$, ils sont "cachés"
- on observe une une émission de ces états $S_i$
  - exemple : on observe le bruit $X$ que fait Doudou le hamster
- Pour chaque état, nous pouvons cependant définir la probabilité que $X$ ait été émis par $S_i$ : $p(X|S_i)$

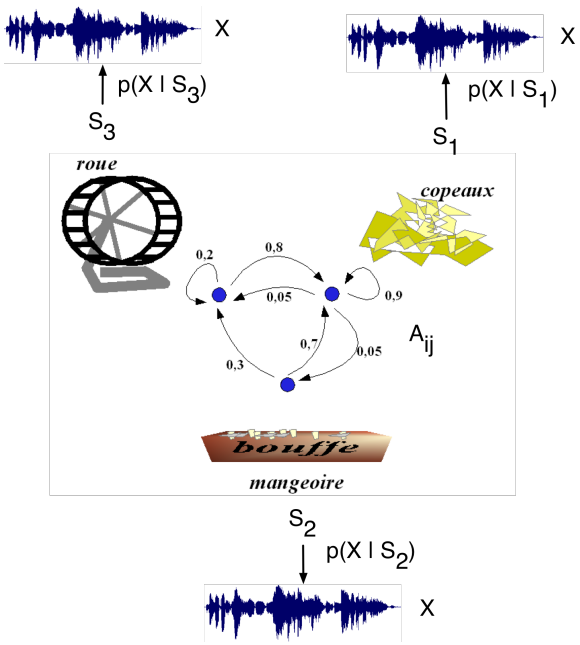



*Figure : Modèle de Markov caché d'une journée de Doudou le hamster*

**Eléments de définition d'un modèle de Markov caché**

- Ils sont
  - La définition des états $S_i$
  - La définition des observations $X$
  - La probabilités d'émission $A_i(X) = P(X_t = X|S_t = i)$ pour chaque état $S_i$
    - i.e. la probabilité que l'état $S_i$ émette $X$
  - La probabilités de transition $T_{ij} = P(S_{t+1} = j|S_t = i)$
    - i.e. la probabilité de transiter de $S_i$ au temps $t$ vers $S_j$ au temps $t + 1$
  - La probabilités initiales $\pi_j = P(s_1 = j)$
    - i.e. la probabilité qu'initialement $(t = 0)$ le modèle se trouve dans l'état $S_j$
- On note $\{\lambda\}$ l'ensemble de ces éléments

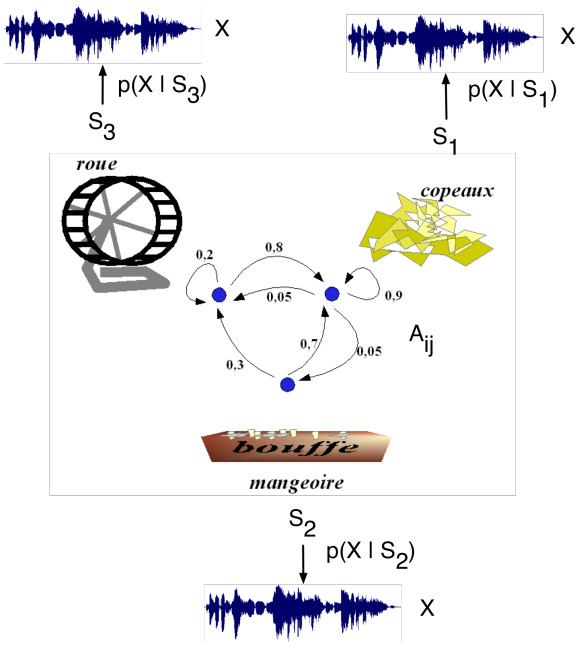



*Figure :*

**Le modèle de Markov caché permet de résoudre les trois problèmes suivants :**

- **Décodage de la séquence d'états** :
  - Etant donné une suite d'observation $X_t$ et un modèle $\{\lambda\}$, quelle est la suite d'état $S_i$ correspondant
    - $S^\star = \arg\max_S P(S|X, \lambda)$
    - Exemple : si on observe la séquence de son $X$ de Doudou et étant donné son modèle dormir/manger/exercice $\{\lambda\}$, quel est la séquence d'activités $S$ de Doudou ?
- **Evaluation** :
  - Etant donné une suite d'observation $X$ et un modèle $\{\lambda\}$, quelle est la probabilité que ce modèle ait généré $X$
    - $P(X|\{\lambda\})$
    - Exemple : comment déterminer si une séquence d'observation du son $X$ correspond au modèle $\{\lambda_1\}$ dormir/manger/exercice de Doudou le hamster , ou à un modèle $\{\lambda_2\}$ dormir/manger/travailler de Bill le salarié
- **Entrainement** :
  - Etant donné une/des suites d'observations $\{X_t\}$, trouver le modèle $\lambda^\star$ qui maximize la vraisemblance des observations :

- $\lambda^\star = \arg\max_\lambda P(X|\lambda)$
- Exemple : comment déterminer les paramètres $\{\lambda\}$ du modèle de Doudou le hamster ?

### 4.2.2 Exemple d'utilisation des modèles de Markov caché : Reconnaissance d'accords

- **Objectif**
  - On veut estimer la suite d'accords $\{C_M, C\#_M, \cdots, C_m, \cdots\}$ d'un morceau de musique à partir de l'observation de son signal audio
- **Méthode**
  - Etat $S_i$ :
    - On définit les différents accords $\{C_M, C\#_M, \cdots, C_m, \cdots\}$ à estimer comme les états $S_i$
  - Observation :
    - on extrait à chaque instant $t$ du signal audio une observation appelée chroma/ Pitch Class Profile
  - Pour chaque accord/état, on définit la probabilité qu'il émette l'observation chroma/ Pitch Class Profile
    - $p(X = chroma|S = C_M)$, $p(X = chroma|S = C\#_M)$, ...

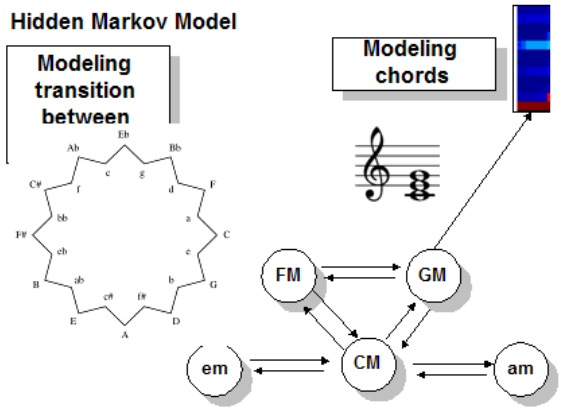



*Figure :*

  - Matrice de transitions entre accords/états :
    - on définit la matrice de transition de manière à respecter la théorie musicale
    - en théorie musicale (cercle des quintes, relatifs majeur-mineur), certains accords s'enchainent mieux ($G_M$ vers $C_M$ = consonance), que d'autres ($G_M$ vers $C\#_M$ = dissonance)

- **Solution**
  - On estime la suite d'accords par **décodage** d'un modèle de Markov Caché

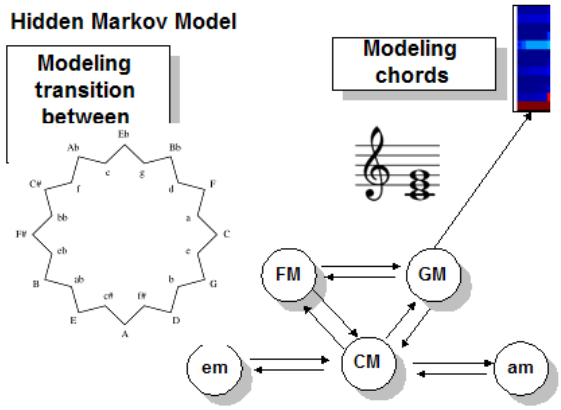



*Figure :*

### 4.2.3 Exemple d'utilisation des modèles de Markov caché : Reconnaissance de parole

Un système de reconnaissance de parole est composé de quatre grandes parties :

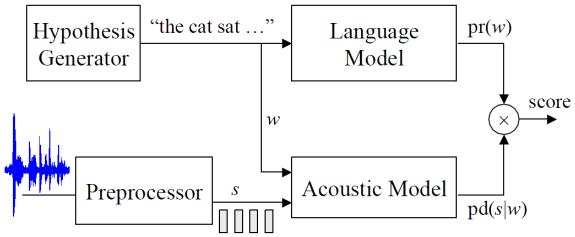



*Figure : Schéma général d'un système de reconnaissance de parole (source : Mike Brookes)*

- 1) **Modèle de langage** :
  - Représente la probabilité d'une séquence de mots (dépend du vocabulaire et de la grammaire d'une langue, indépendant du signal audio)

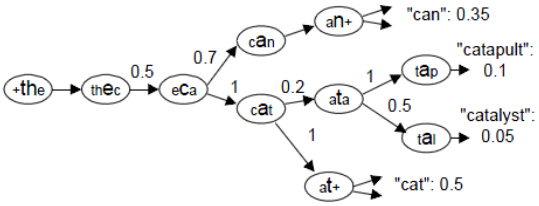



*Figure : Modèle de langage (source : Mike Brookes)*

- 2) **Phonétiseur** :
  - Transforme les mots en séquence de **phonèmes**
    - Phonème : plus petite unité distinctive que l'on puisse isoler : cote (/k ?t/) et côte (/kot/)
    - Pour une même langue, selon l'accent ,il existe plusieurs prononciations d'un même mot (petit, p'tit) donc plusieurs suite de phonèmes possibles
    - 37 phonèmes en français

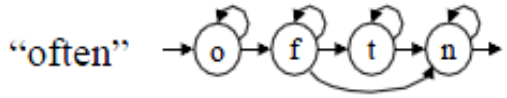



*Figure :*

  - Transforme les mots en séquence de **tri-phones**
    - $37^3$ tri-phone en français

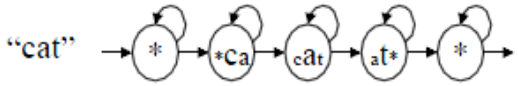



*Figure :*

- 3) **Pré traitement audio** :
  - extrait des observations $X$ pertinentes du signal audio
    - Généralement : MFCC + $\Delta$ MFCC + $\Delta\Delta$ MFCC (L=25ms, 40 bandes, $3 \star 39$ coefficients)

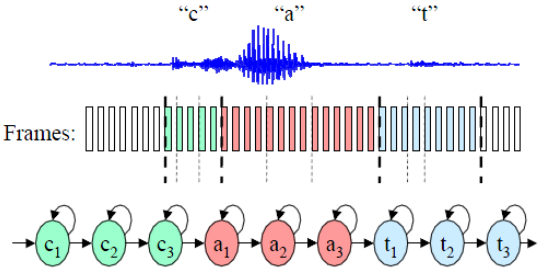



*Figure : Représentation d'un mot en séquence de phonème et représentation acoustique des phonèmes (source : Mike Brookes)*

- 4) **Modèle acoustique :**
  - définit un modèle de Markov caché permettant la jonction entre un phonème / tri-phone et les différentes occurrences acoustiques (différents locuteurs)

### 4.2.4 Exemple d'utilisation de modèles de Markov caché : alignement de parole à un texte

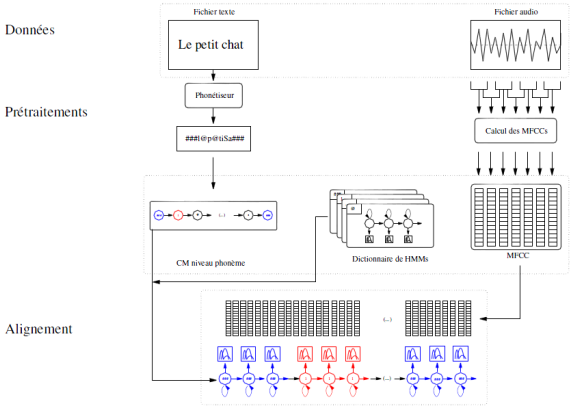



*Figure : source : Pierre Lanchantin*

# 5 Approche discriminante

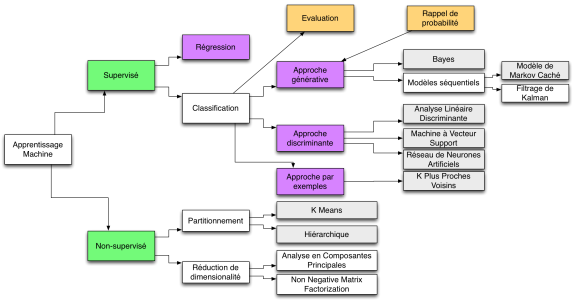



*Figure :*

## 5.1 Frontière de décision

- La conséquence de l'approche générative est une frontière de décision séparant les deux classes
- Nous l'appelons $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$
- $y(\boldsymbol{x})$ divise l'espace en deux sous-espaces
  - $y(\boldsymbol{x}) = 0$ : points sur la frontière
  - $y(\boldsymbol{x}) > 0$ : points dans la direction de $\boldsymbol{w}$
  - $y(\boldsymbol{x}) < 0$ : points dans la direction opposée de $\boldsymbol{w}$
- Distance du point $\boldsymbol{x}$ à la frontière : $\frac{y(x)}{||\boldsymbol{w}||}$
- Pour la classification, choix de
  - la classe 0 si $y(x) < 0$
  - la classe 1 si $y(x) > 0$

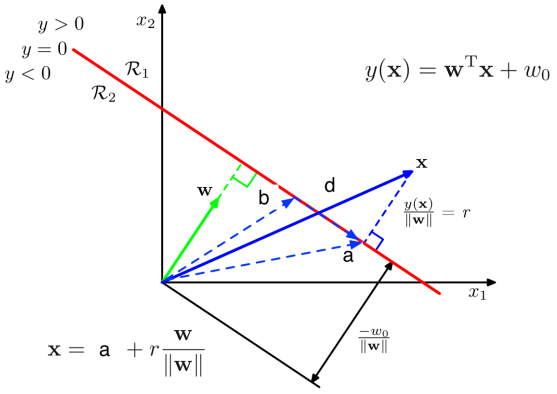



*Figure : source : Hugo Larochelle*

**Comment choisir cette frontière pour séparer le mieux les classes ?**

- Analyse Linéaire Discriminante
- Machine à Vecteur Support

## 5.2 Analyse Linéaire discriminante

- $y(\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x}$ est le résultat de la projection de $\boldsymbol{x}$ sur l'hyper-plan $\boldsymbol{w}$
- l'ALD cherche la projection qui
  - **maximise la séparation des moyennes** $m_i$ des données projetées
    - $m_i = \frac{1}{N_i}\sum_{n\in C_i} \boldsymbol{w}^T\boldsymbol{x}_n$
  - **minimise les variances intra-classes** $s_i^2$ des entrées projetées
    - $s_i^2 = \sum_{n\in C_i}(\boldsymbol{w}^T\boldsymbol{x}_n - m_i)^2$

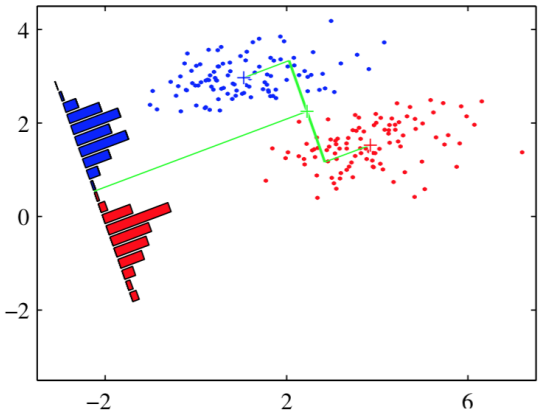



*Figure : source : Hugo Larochelle*

- Au total on maximise

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\textbf{variance inter-classes}}{\textbf{variance intra-classe}}$$

- La solution est $\boldsymbol{w} \propto S_w^{-1}(m_1 - m_2)$
- avec $S_w$ la matrice de co-variance intra-classe

$$S_w = \sum_{n\in C_i}(x_n - m_1)(x_n - m_1)^T + \sum_{n\in C_i}(x_n - m_2)(x_n - m_2)^T$$

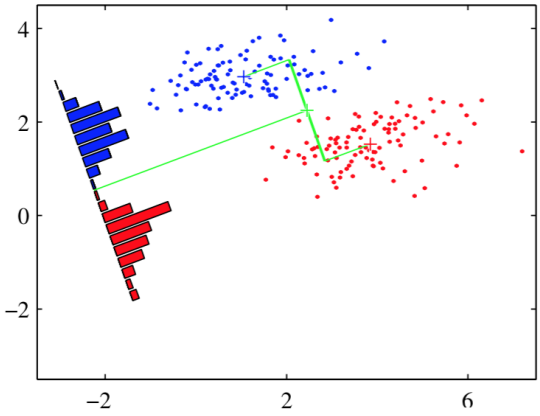



*Figure : source : Hugo Larochelle*

**Exemple : application de l'ALD pour la reconnaissance des instruments**

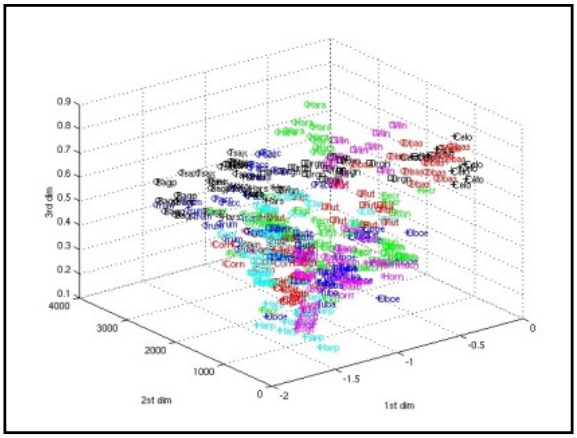



*Figure : Chaque point représente un son d'instrument dans l'espace à 3 dimensions des observations ($D = 3$, chaque couleur représente une classe (classe d'instrument de musique)*

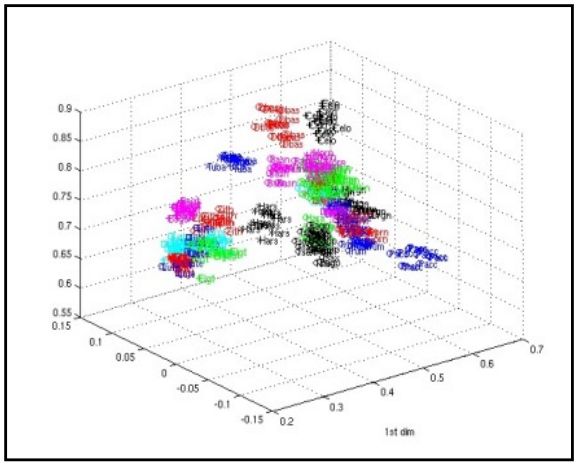



*Figure : Après projection des points sur l'axe de l'ALD, les classes (couleurs) sont mieux séparées.*

## 5.3 Réseaux de neurones artificiels

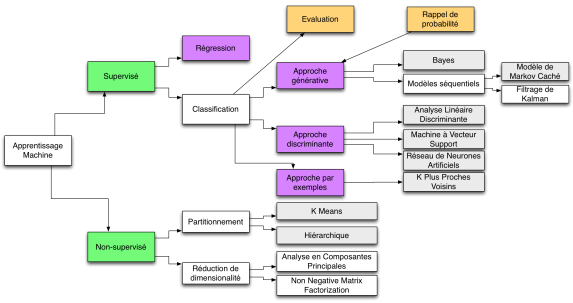



*Figure :*

### 5.3.1 Cerveau et neurones

- Les réseaux de neurones artificiels tentent de reproduire la manière dont le cerveau traite l'information
- Dans le cerveau l'information est traitée par un réseau complexe de neurones inter-connectés
- Les neurones des différentes régions du cerveau sont spécialisée dans des traitements spécifiques

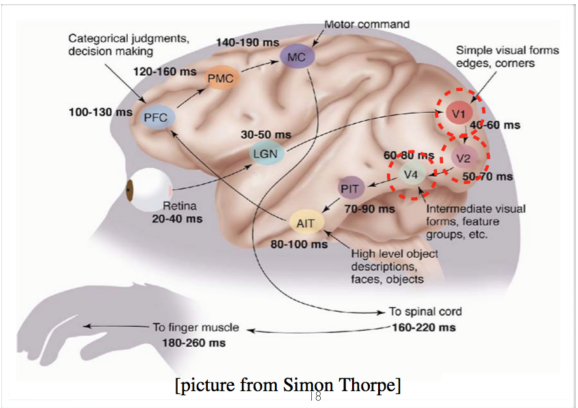



*Figure : source : Hugo Larochelle*

- Il y a environ $10^{10}$ à $10^{11}$ neurones dans notre cerveau
- Entre ces neurones circulent un signal électrique
- Tous les neurones sont connectés entre eux à travers des **dendrites**
  - chaque neurone transforme l'information qu'il reçoit dans le corpus de sa cellule (**soma**)
  - chaque neurone retourne le signal à travers un cable appelé **axon**
  - le point de connection entre ce cable (axon) et les dendrites des autres neurones sont appelés **synapses**

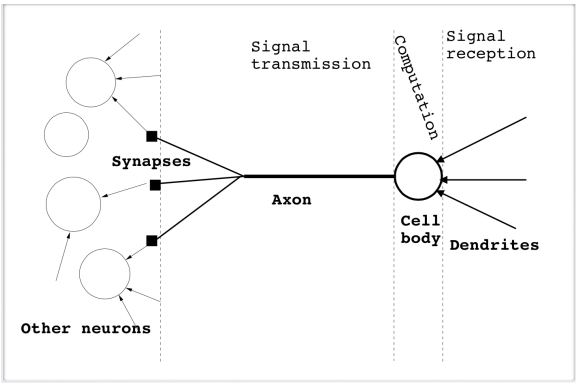



*Figure : source : Hugo Larochelle*

### 5.3.2 Réseaux de Neurones Artificiels

- Artificial Neural Network (ANN)
- Un Réseau de Neurone Artificiel reproduit l'interconnection entre les différents neurones

- Les neurones artificiels sont organisés en couches (layer)
  - Multi-Layer-Perceptron (MLP)
- Tous les neurones d'une couche sont connectés aux neurones de la couche suivante

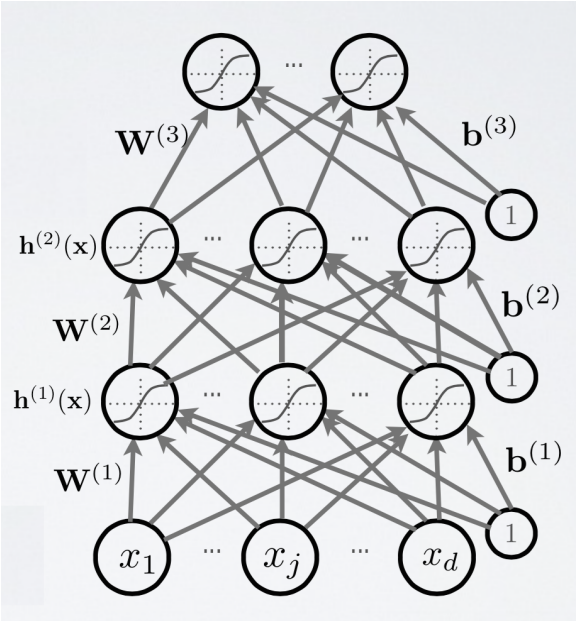



*Figure : source : Hugo Larochelle*

**Un neurone artificiel**

- chaque neurone est représenté par une fonction
  - prenant en entrée le signal des autres neurones (avec une pondération spécifique à chacun d'entre eux)
  - effectue une transformation de la somme des signaux résultants
  - retourne le signal vers l'étage de neurones suivant

**Un neurone artificiel mathématiquement**

- 1 Pre-activation d'un neurone (activation des entrées)
  - $a(\boldsymbol{x}) = \sum_i w_i x_i + b = \boldsymbol{w}^T \boldsymbol{x} + b$
    - $\boldsymbol{w}$ sont les poids des connections (détermine quels neurones précédents apportent une information)
    - $b$ est le biais du neurone
- 2 Activation du neurone
  - $h(\boldsymbol{x}) = g(a(\boldsymbol{x})) = g(\boldsymbol{w}^T \boldsymbol{x} + b)$

– $g$ est la fonction d'activation (généralement une fonction non-linéaire)

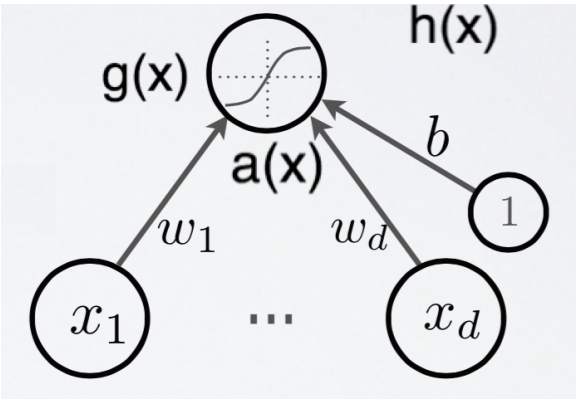



*Figure : source : Hugo Larochelle*

**Plusieurs couches de neurones artificiels mathématiquement**

- 1 Pre-activation d'un neurone à l'étage $k$
  - $\boldsymbol{a}^{(k)}(x) = \boldsymbol{w}^{(k)}\boldsymbol{h}^{(k-1)}(\boldsymbol{x}) + \boldsymbol{b}^{(k)}$
- 2 Activation d'un neurone à l'étage $k$
  - $\boldsymbol{h}^{(k)} = \boldsymbol{g}(\boldsymbol{a}^{(k)}(\boldsymbol{x}))$
- 3 Activation de l'**étage de sortie**
  - $\boldsymbol{h}^{(L+1)} = \boldsymbol{o}(\boldsymbol{a}^{(L+1)}(\boldsymbol{x})) = \boldsymbol{f}(\boldsymbol{x})$

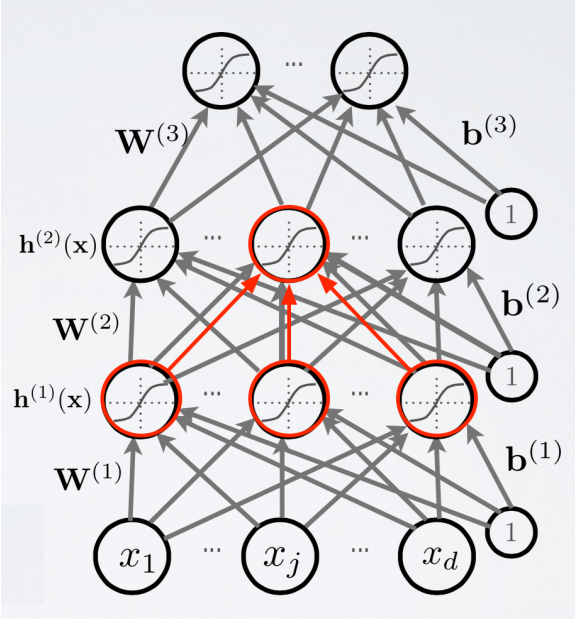



*Figure : source : Hugo Larochelle*

### 5.3.3 Entraînement d'un Réseau de Neurones Artificiels

- beaucoup de paramètres à apprendre : $\boldsymbol{w}^{(k)}, \boldsymbol{b}^{(k)}, \cdots$
- on impose les valeurs à l'étage de sortie
- algorithme de **back propagation**
- descente de gradient

# 6 Approche par exemplification

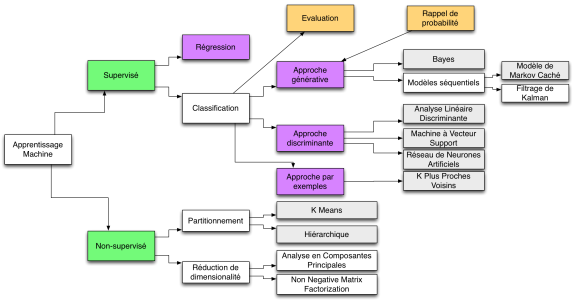



*Figure :*

## 6.1 Algorithme des K Plus Proches Voisins

- Note : une observation $\boldsymbol{x}_n$ est un point dans un espace à $D$ dimensions, appelé espace des descripteurs
- **Entraînement** :
  - exemple : $\boldsymbol{x}_n$ = la valeur des descripteurs audio à $D$ dimensions
  - on remplit l'espace des descripteurs par l'ensemble des $N$ "points" d'apprentissage : $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$
  - à chaque point $x_n$ est associé sa classe $t_n \in \{1, \cdots, C\}$
- **Evaluation** :
  - Soit $x^\star$ une observation nouvelle de classe $t^\star$ inconnue
  - On recherche dans l'espace des descripteurs les $K$ points les plus proches de $x^\star$ selon une distance
    - généralement on utilise une distance euclidienne
  - On associe à $x^\star$ la classe majoritaire parmi celles assignées au $K$ plus proche voisins $\{t_k\}$
    - $t^\star = \arg\max_{c \in \{1,\cdots,C\}} \sum_{k=1}^{K} \delta(c, t_k)$
- **Paramètres** :
  - on doit choisir le nombre $K$ de plus proches voisins considérés
  - on doit choisir le type de distance utilisé (euclidienne ou autres)
    - si euclidien, cela suppose que les dimensions $d$ sont d'échelles comparables ; sinon normalisation
- **Avantage** :
  - Il n'y a pas de modèle à apprendre !
- **Désavantage** :
  - demande le stockage et l'accès à toutes les données (le nombre de données peut être très très grand)
  - il faut calculer la distance entre $\boldsymbol{x}_m$ et tous les point $\boldsymbol{x}_k$

– ce coût de calcul peut être très important

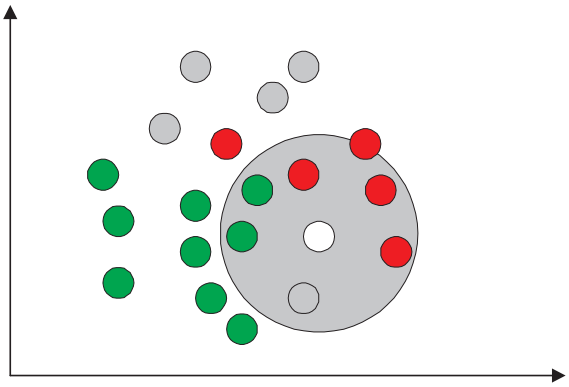


*Figure : Algorithm des K plus proches voisins. Le point blanc est de classe inconnue. Si $K = 6$ on lui assignera la classe "rouge" car c'est la classe majoritaire parmi c'est 6 plus proches voisins.*

# 7 Evaluation

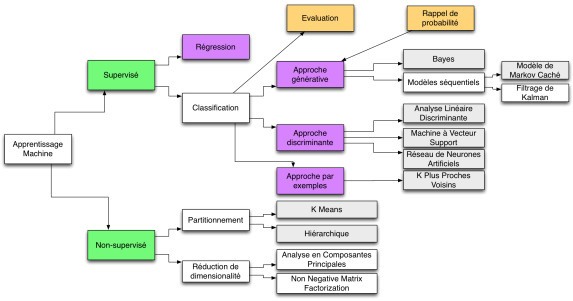



*Figure :*

## 7.1 Base d'évaluation

Un base d'évaluation sert à évaluer les performances d'un système de prédiction (régression ou classification).

- Elle est composée d'une partie
  - entrainement $\mathbb{D}_{train}$ : pour entraîner le système
  - test $\mathbb{D}_{test}$ : pour évaluer les performances du système entraîné
- Les deux parties sont formés de couples
  - observations/ cibles $(x_n, t_n)$

### 7.1.1 Exemple de base type d'évaluation : base Iris

La base IRIS permet de prédire 3 variétés de fleurs IRIS (Setosa, Versicolour, Virginica) à partir de mesures de la longueur et épaisseur de leurs pétales et sépales

- Observations $X$ : $X$ a 4 dimensions ($D = 4$)
  - 1. sepal length in cm
  - 2. sepal width in cm
  - 3. petal length in cm
  - 4. petal width in cm

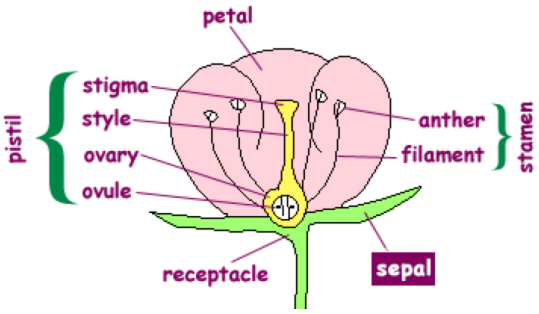



*Figure : Illustration des sépals et pétals*

- Cibles $Y$ : $Y$ a 3 valeurs de classe possibles ($K = 3$)
  - 1. Iris Setosa
  - 2. Iris Versicolour
  - 3. Iris Virginica

*Figure : Variété Setosa d'iris*

*Figure : Variété Versicolour d'iris*

*Figure : Variété Virginica d'iris*

## 7.2 Comment séparer $\mathbb{D}$ en ensemble d'entraînement $\mathbb{D}_{train}$ et de test $\mathbb{D}_{test}$ ?

- Dans le cas idéal, nous avons accès à deux bases indépendantes d'entraînement $\mathbb{D}_{train}$ et de test $\mathbb{D}_{test}$.
- Dans la plupart des cas, nous n'avons qu'une base $\mathbb{D}$ qu'il faut diviser en une partie pour l'entrainement $\mathbb{D}_{train}$ et une autre pour le test $\mathbb{D}_{test}$.
- Comment faire pour les séparer ?
  - 1. N-Fold Cross-Validation
  - 2. Leave-one-out Cross-Validation

### 7.2.1 N-Fold Cross-Validation

- $\mathbb{D}$ est divisé en $N$ sous-ensemble $\mathbb{D}_n$
  - a) on en désigne un parmi les N qui sera utilisé pour le test
  - b) on utilise les $N-1$ autres pour l'entraînement
  - on réitère a) et b) en choisissant à chaque fois un nouveau sous-ensemble de test parmi les $N$ possibles
- on calcul la moyenne des indices à travers les folds

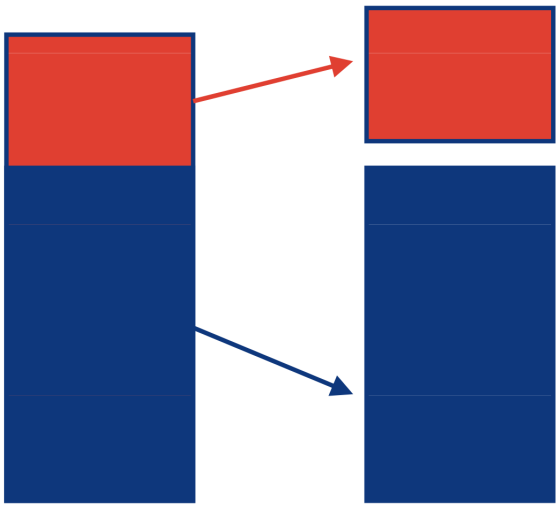


*Figure : source : Arshia Cont*

### 7.2.2 Leave-one-out Cross-Validation

- le cas limite du N-Fold Cross Validation quand $N$ est égal au nombre de données
  - a) on choisi une donnée qui sera utilisé pour le test
  - b) on utilise toutes les autres données pour l'entraînement
  - on réitère ...
- on calcul la moyenne des indices ...

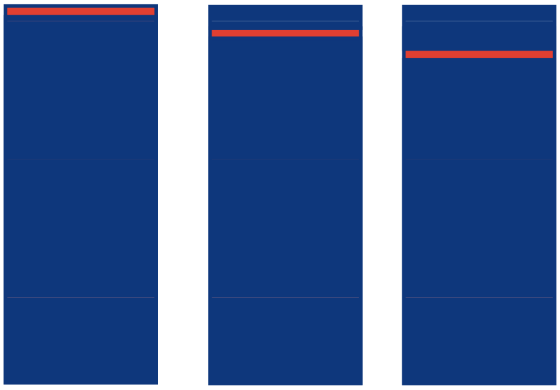


*Figure : source : Arshia Cont*

## 7.3 Comment évaluer les performances d'un algorithme de classification ?

- On évalue les performances sur l'ensemble de test $\mathbb{D}_{test}$
- Pour chaque $x_m$ de $\mathbb{D}_{test}$, on compare la prédiction $\hat{t}_m = y(x_m)$ à la "vérité terrain" $t_m$

**Dans le cas de deux classes ($c_1$=Positif, $c_2$=Negatif), on mesure les quantités suivantes**

- **True Positif** (TP) :
  - Nombre de données Positives détectées correctement (True) comme Positives
- **False Positif** (FP) :
  - Nombre de données Négatives détectées faussement (False) comme Positives → Fausse Alarm (False Alarm)
- **True Negatif** (TN) :
  - Nombre de données Négatives détectées correctement (True) comme Négatives
- **False Negatif** (FN) :
  - Nombre de données Positives détectées faussement (False) comme Négatives → Détection Manquée (Miss Detection)

**Matrice de confusion.**

- On peut représenter cela dans une **matrice de confusion**

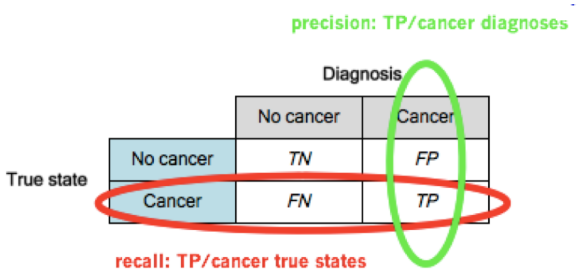



*Figure : Exemple de matrice de confusion pour la détection de risque de cancer*

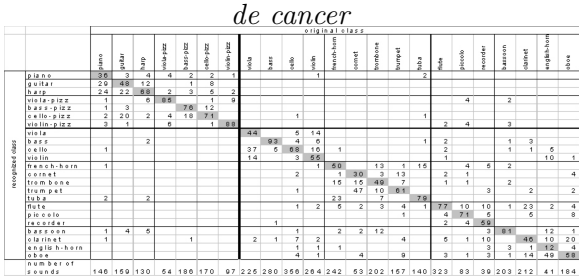


*Figure : Exemple de matrice de confusion pour la reconnaissance des instruments de musique*

## Calcul des indices

- **Rappel** (Recall) :
  - Mesure la capacité à retrouver tous les $c_i$

$$\begin{aligned} Recall &= \frac{\#\text{ données détectées comme } c_i \text{ et étant réellement } c_i}{\#\text{ données étant réellement à } c_i} \\ &= \frac{TP}{TP + FN} \end{aligned} \tag{1}$$

- **Precision** :
  - Mesure la capacité à retrouver uniquement des $c_i$ (moteur de recherche comme Google)

$$\begin{aligned} Precision &= \frac{\#\text{ données détectées comme } c_i \text{ et étant réellement } c_i}{\#\text{ données détectées comme } c_i \text{ (correctes ou fausses)}} \\ &= \frac{TP}{TP + FP} \end{aligned} \tag{2}$$

- **F-measure** :

$$F - measure = \frac{2Rappel \cdot Precision}{Rappel + Precision} \tag{3}$$

- **Accuracy** :
  - Mesure les performances globales indépendamment de la **distribution** des classes

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

# 8 Element de probabilité

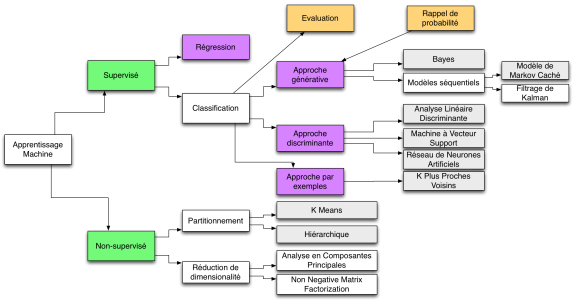



*Figure :*

## 8.1 Introduction

Nous considérons nos données ($X$ et $Y$) comme des variables aléatoires :
- la valeur d'une variable aléatoire est incertaine (avant de l'observer)
- la loi de probabilité de la variable aléatoire caractérise notre incertitude par rapport à sa valeur

## 8.2 Variable aléatoire

### 8.2.1 Variable aléatoire discrète

- $X$ ne prend que des valeurs **discontinues** dans un intervalle donné (borné ou non)
  - Exemple : le nombre d'élèves qui assiste à ce cours
- **Loi de probabilité** (distribution de probabilité) :
  - caractérise $X$ par l'ensemble des valeurs qu'il peut prendre et par l'expression mathématique de la probabilité de ces valeurs
  - la probabilité que $X = x_i$ est notée $P(X = x_i)$
- **Propriété** :
  - $P(X = x_i) \geqslant 0$
  - $\sum_i P(X = x_i) = 1$
- Fonction de **répartition** (distribution des probabilités cumulées) :
  - $t \to F_X(t) = P(X < t)$

### 8.2.2 Exemples de loi de probabilité discrètes

- Loi **uniforme discrète**
  - $P(X = x_1) = P(X = x_2) = \cdots = P(X = x_n) = \frac{1}{n}$
- Loi **de Bernoulli**

  - correspond à une expérience à deux issues (succès–échec), généralement codées respectivement par les valeurs 1 et 0
  - dépend d'un paramètre $p \in [0, 1]$
  - $P(X = 1) = 1 - P(X = 0) = p$
- Loi **binomiale**
  - nombre $k$ de succès obtenus à l'issue de $n$ épreuves de Bernoulli indépendantes de paramètre $p \in [0, 1]$
  - $P(X = k) = \binom{n}{k} p^k (1 - p(^\circ n - k)$
- Loi **géométrique**
  - loi qui modélise le temps d'attente du premier succès dans une série d'épreuves de Bernoulli indépendantes à probabilité de succès $p \in [0, 1]$
  - $P(X = k) = (1 - p)^{k-1} p$

### 8.2.3 Variable aléatoire continue

- $X$ peut prendre toutes les valeurs **continues** dans un intervalle donné (borné ou non borné)
  - Exemple : le nombre de Joules dépensé par votre cerveau pour suivre ce sours
- **Fonction densité de probabilité** :
  - associe une probabilité à chaque ensemble de valeurs définies dans un **intervalle** donné
  - probabilité associée à un évènement est nulle
    - $P(X = a) = 0$
    - impossible d'observer exactement cette valeur
  - probabilité associé à une intervalle $[a, b]$
    - $P(a \leqslant X \leqslant b) = \int_a^b f(x)dx$
- **Propriétés** :
  - $f(x) \geqslant 0 \quad \forall x \in \mathbb{R}$
  - $\int_{-\infty}^{\infty} f(x)dx = 1$
- Fonction de **répartition** :
  - $t \to F_X(t) = P(X < t) = \int_{-\infty}^{t} f(x)dx$
  - $P(a \leqslant X \leqslant b) = F_X(b) - F_X(a) = \int_a^b p(x)dx$

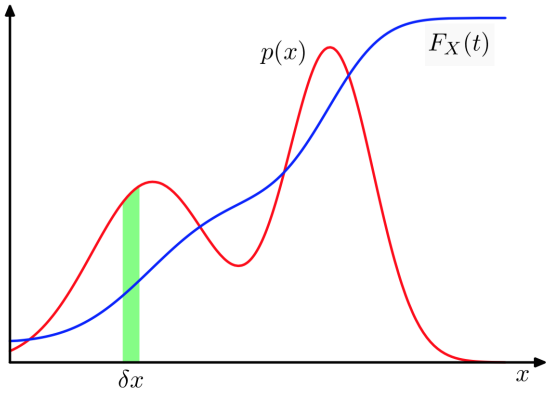



*Figure : source : Hugo Larochelle*

### 8.2.4 Exemples de loi de probabilité continues

- Loi **exponentielle**
  - modélise le temps de vie d'un phénomène puisque c'est l'unique loi absolument continue possédant la propriété de perte de mémoire. En ce sens elle est l'analogue continu de la loi géométrique
  - $f(x) = \lambda e^{-\lambda x}$
- Loi **normale**, ou loi **gaussienne**
  - décrit le comportement des séries d'expériences aléatoires lorsque le nombre d'essais est très grand. C'est la loi limite dans le théorème central limite
  - $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

## 8.3 Espérance $E(X)$

- L'espérance $E(X)$ donne une "idée générale» (la tendance générale, la **moyenne**) de la valeur de $X$
- **Calcul**
  - moyenne des valeurs possibles de X pondérées par les probabilités associées à ces valeurs
- Cas discret : $E(X) = \sum_{n=1}^{N} x_n P(X = x_n)$
- Cas continu : $E(X) = \int_{x=-\infty}^{\infty} x f(x) \partial x$
- **Propriétés**
  - $E(X + Y) = E(X) + E(Y)$
  - $E(aX) = aE(X) \quad \forall a \in \mathbb{R}$

## 8.4 Variance $V(X)$

- La variance $V(X)$ mesure la **dispersion** de $X$ autour de son espérance
  - on appelle écart-type $\sigma(A) = \sqrt{V(X)}$
- **Calcul**
  - espérance mathématique du carré de l'écart à l'espérance mathématique.
- Cas discret : $V(X) = \sum_{n=1}^{N} \{x_n - E(X)\}^2 P(X = x_n)$
- Cas continu : $V(X) = \int_{x=-\infty}^{\infty} \{x - E(X)\}^2 f(x) dx$
- **Propriétés**
  - $V(X) = E(X - E(X))^2$
  - $V(X) = E(X^2) - E^2(X)$
  - $V(aX) = a^2 V(X) \quad \forall a \in \mathbb{R}$
  - $V(aX + b) = a^2 V(X) \quad \forall (a, b) \in \mathbb{R}$

## 8.5 Cas de deux variables aléatoires

- Loi **jointe**, probabilité **jointe** :
  - probabilité d'observer simultanément $X = x_i$ et $Y = y_j$
    - Cas discret : $p_{xy} = P(X = x_i, Y = y_j)$
    - Cas continu : $p_{xy} = P(\{x_a < X < x_b\}, \{y_c < Y < y_d\})$
- Variables aléatoires **indépendantes** :
  - $P(X, Y) = P(X)P(Y)$
  - dans ce cas on a
    - Espérance $E(X, Y) = E(X)E(Y)$
    - Variance $V(X + Y) = V(X) + V(Y)$
- **Covariance** :
  - Rappel : la variance est définie comme $V(X) = E(\{X - E(X)\}\{X - E(X)\}$
  - La covariance est définie comme $cov(X, Y) = E(\{X - E(X)\}\{Y - E(Y)\})$
- **Corrélation** : $R(X, Y) = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)}$
- Propriétés :
  - si X et Y sont deux variables aléatoires indépendantes alors $cov(X, Y) = 0$
- Probabilité **marginale** :
  - lorsqu'on ne s'intéresse pas à toutes les variables aléatoire qu'on a défini
    - $P(X = x_i) = \frac{c_i}{N} = \sum_j P(X = x_i, Y = y_j)$
- Probabilité **conditionnelle** :

– la valeur d'une variable aléatoire "étant donnée" une valeur assignée à d'autres variables

$$\begin{aligned} P(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} \\ &= \frac{n_{ij}}{c_i}\frac{c_i}{N} \\ &= P(Y = y_j | X = x_i)P(X = x_i) \end{aligned}$$

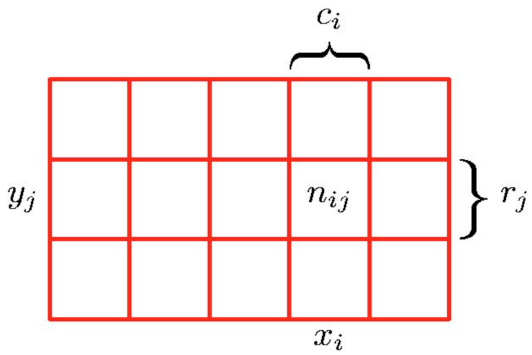



*Figure : source : Arshia Cont*

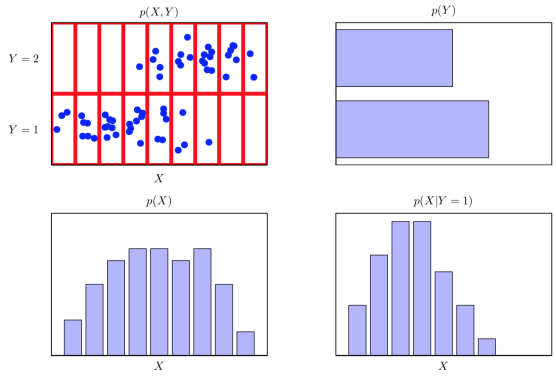



*Figure : source : Hugo Larochelle*

## 8.6 Loi de Bayes

- Une probabilité jointe peut toujours être décomposé
  – comme le produit d'une probabilité conditionnelle et marginale

$$P(X, Y) = P(Y|X)P(X) = P(X|Y)P(Y)$$

- La **loi de Bayes** permet d'inverser l'ordre d'une probabilité conditionnelle :
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$
  - dans lequel $p(X) = \sum_Y p(X|Y)p(Y)$
  - $p(Y)$ est appelée probabilité **a priori**
  - $p(Y|X)$ est appelé probabilité **a posteriori**

## 8.7 La loi normale ou loi gaussienne

- loi simple et pratique pour exprimer notre incertitude sur $X$
- densité de probabilité la plus élevé pour $X = \mu$
- incertitude sur $X$ exprimé par la variance $\sigma^2$

### 8.7.1 Formulation à $D = 1$ dimension

- $p(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$
- Espérance $E(x) = \mu$
- Variance $V(x) = \sigma^2$

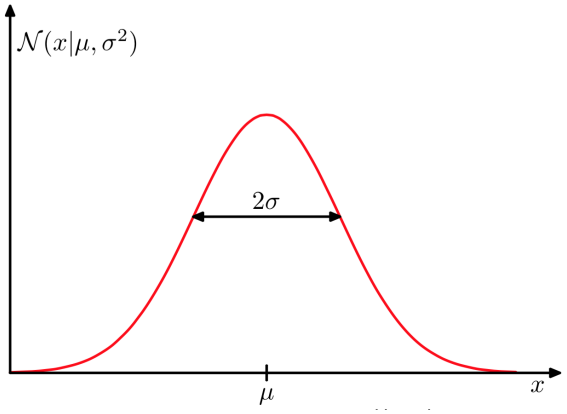



*Figure : source : Hugo Larochelle*

### 8.7.2 Formulation à $D > 1$ dimension

- $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\Sigma^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$
- Espérance $E(\boldsymbol{x}) = \boldsymbol{\mu}$
- Matrice de co-variance $cov(\boldsymbol{x}) = \boldsymbol{\Sigma}$
- Remarque :
  - $\boldsymbol{\Sigma}$ est une matrice!!!

– permet de représenter les dépendances (corrélation) entre les dimensions $d$ de $X$

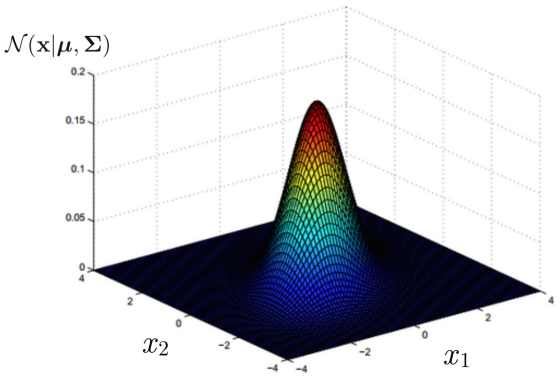



*Figure : source : Hugo Larochelle*

### 8.7.3 Exemples de dépendances représentées par $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

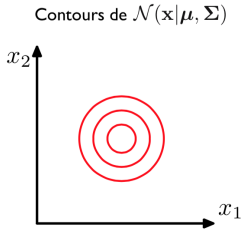



*Figure : Variables indépendantes et $\sigma_1 = \sigma_2$ (source : Hugo Larochelle)*

$$\boldsymbol{\Sigma} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

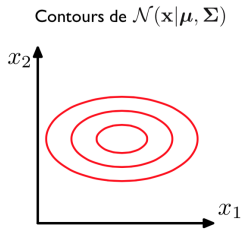



*Figure : Variables indépendantes et $\sigma_1 = 2\sigma_2$ (source : Hugo Larochelle)*

$$\mathbf{\Sigma} = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$$

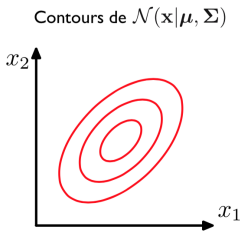



*Figure : Variables corrélées (source : Hugo Larochelle)*