

# ENSEA 2ème AuPAM (2017-2018)

## Introduction à l'apprentissage machine

Geoffroy.Peeters@ircam.fr  
UMR SMTS 9912 (IRCAM CNRS UPMC)

1. Introduction
  - 1.1 Qu'est que l'apprentissage machine ?
  - 1.2 Deux grands types d'apprentissage machine
  - 1.3 Deux grandes cibles pour l'apprentissage supervisé
  - 1.4 Trois grandes approches pour l'apprentissage supervisé
  - 1.5 En résumé
  - 1.6 Généralisation
2. Régression
  - 2.1 Régression polynomiale en  $D = 1$  dimension
  - 2.2 Régression polynomiale en  $D > 1$  dimensions
3. Approche générative
  - 3.1 Décision Bayésienne
  - 3.2 Modèles séquentiels
4. Approche discriminante
  - 4.1 Arbres de décision
  - 4.2 Frontière/surface de décision
  - 4.3 Analyse Linéaire Discriminante (ALD)
  - 4.4 Machine à Vecteurs Supports (SVM)
  - 4.5 Réseaux de neurones artificiels
5. Approche par exemplification

- 5.1 Algorithme des K Plus Proches Voisins
6. Bases de référence pour l'entraînement et le test
  - 6.1 Base IRIS
  - 6.2 Base MNIST
  - 6.3 Base CIFAR-10/ CIFAR-100
7. Evaluation d'un système de classification
  - 7.1 Indices de performance
  - 7.2 Séparer ensemble d'entraînement et de test
8. Apprentissage non-supervisé
  - 8.1 Introduction
  - 8.2 Algorithmes des K-Means (nuées dynamiques)
  - 8.3 Algorithme hiérarchique par agglomération
9. Elements de probabilité
  - 9.1 Introduction
  - 9.2 Variable aléatoire
  - 9.3 Espérance  $E(X)$
  - 9.4 Variance  $V(X)$
  - 9.5 Cas de deux variables aléatoires
  - 9.6 Loi de Bayes
  - 9.7 La loi normale ou loi gaussienne

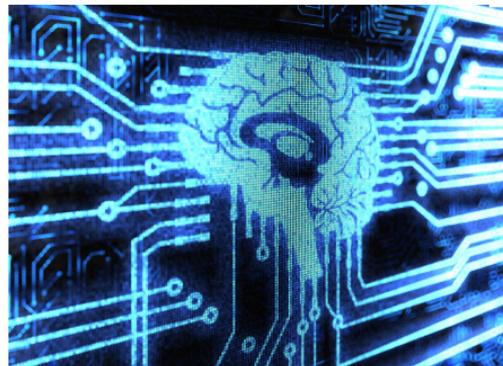
- Avant vendredi 17h
  - Envoyer impérativement par mail à reynal add ensea.fr
    - choix de theme de nano projet
    - nom des élèves constituant leur groupe de projet (de 1 à 3 élèves max)
- Jeudi 29 mars de 13h30 à 15h30
  - Examen écrit

# 1- Introduction

## Qu'est que l'apprentissage machine?

### 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



source : <http://sira-corp.com/new/MachineLearning>

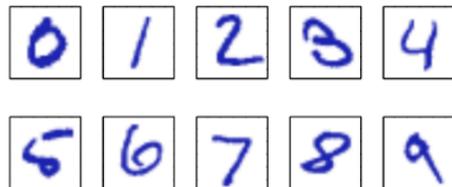
# 1- Introduction

## Qu'est que l'apprentissage machine ?

### 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, difficile 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)



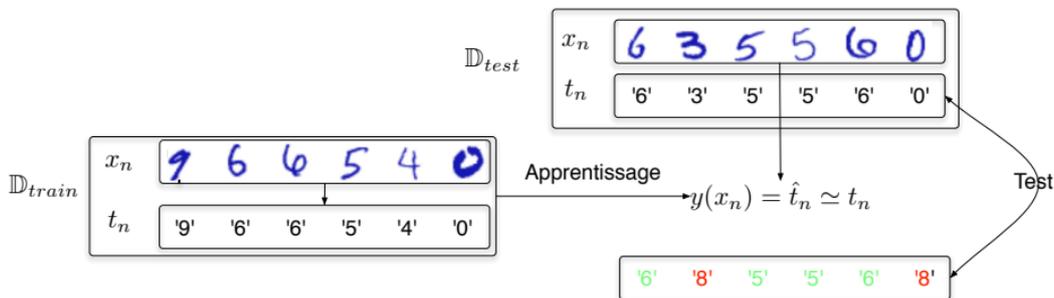
source : Hugo Larochelle

# 1- Introduction

## Qu'est que l'apprentissage machine ?

### Comment ça marche ?

- 1) On donne à l'algorithme des données d'**entraînement**
  - On appel **ensemble d'entraînement** :  $\mathbb{D}_{train} = \{(x_1, t_1), \dots, (x_N, t_N)\}$ 
    - $x_n$  une **observation** (entrée du système)
    - $t_n$  la **cible** correspondante (sortie) du système
- 2) L'algorithme d'apprentissage machine **apprend** un **modèle**  $y(x)$ 
  - Le modèle cherche à prédire  $t$  en fonction de  $x$  :  $y(x_n) = \hat{t}_n \simeq t_n$
- 3) On mesure la qualité de l'apprentissage (du modèle) sur un **ensemble de test** :  
 $\mathbb{D}_{test} = \{(x_{N+1}, t_{N+1}), \dots, (x_{N+M}, t_{N+M})\}$ 
  - Le modèle doit être capable de généraliser à de nouvelles données



# 1- Introduction

## Deux grands types d'apprentissage machine

### Apprentissage supervisé

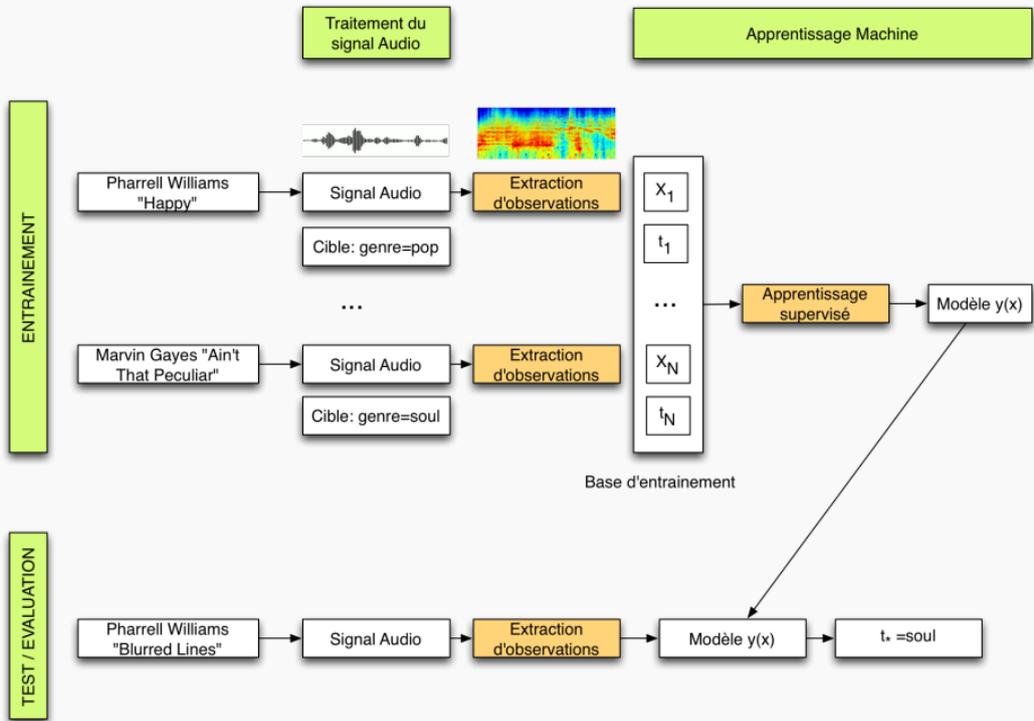
Nous considérons un ensemble d'**observations** (entrées du système)  $\{x_1, \dots, x_N\}$

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

# 1- Introduction

## Deux grands types d'apprentissage machine

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



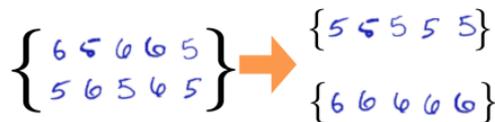
# 1- Introduction

## Deux grands types d'apprentissage machine

### Apprentissage **non-supervisé**

Nous considérons un ensemble d'**observations** (entrées du système)  $\{x_1, \dots, x_N\}$

- Nous ne donnons pas à la machine les cibles
- $\mathbb{D}_{train} = \{x_1, \dots, x_N\}$
- L'objectif de la machine est de créer un modèle de  $x$ , un partitionnement (clustering) des données
  - Utilisation? analyse de données, prise de décisions

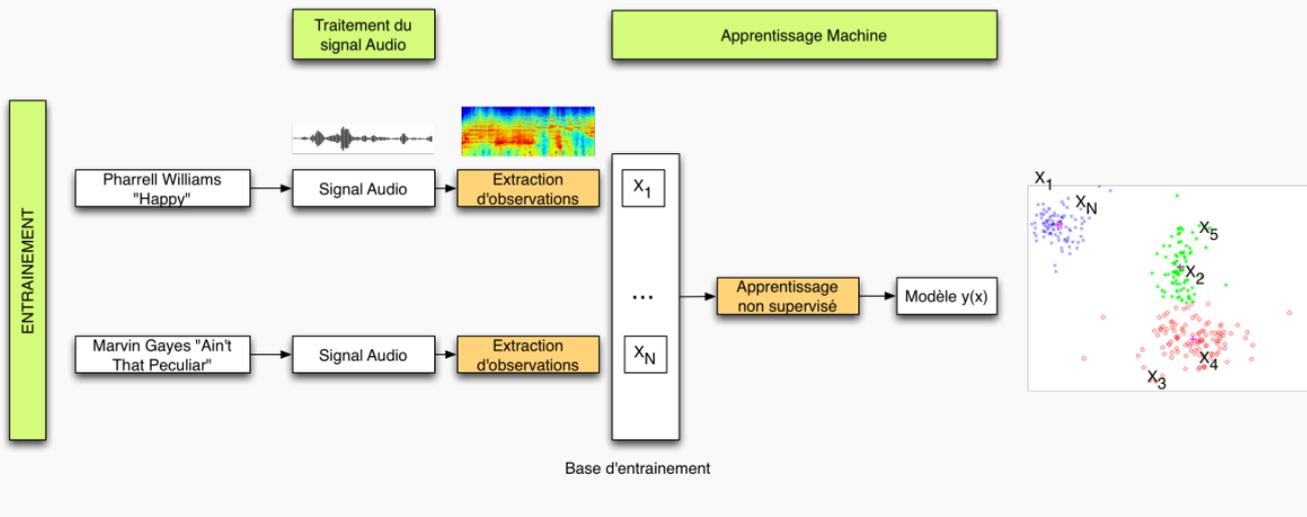


source : Hugo Larochelle

# 1- Introduction

## Deux grands types d'apprentissage machine

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



# 1- Introduction

## 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) :
    - $x$  = activité économique de la journée,  $t$  = la valeur d'une action demain
  - Audio (reconnaissance de tempo) :
    - $x$  = le contenu spectral du signal,  $t$  = le tempo du morceau

### La classification

- La cible est un **indice de classe** :  $t_n \in \{1, \dots, C\}$
- 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

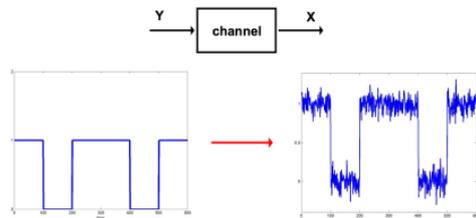
# 1- Introduction

## Trois grandes approches pour l'apprentissage supervisé

### 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  $X$  (observable)
  - → on **infère**  $Y$  à partir de  $X$



source : Arshia Cont

# 1- Introduction

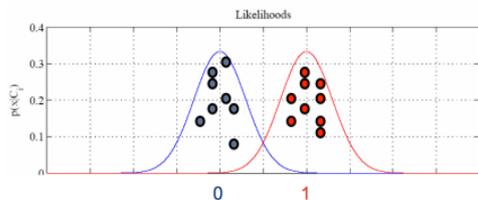
## Trois grandes approches pour l'apprentissage supervisé

### 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  $Y = 1$  :  $P(X|Y = 1)$
- On en déduit les probabilités  $P(Y = 0|X)$  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**  $g(x)$ 
  - $g(x)$  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 Bayésien, modèle de Markov caché, réseaux de neurones artificiels



source : Arshia Cont

# 1- Introduction

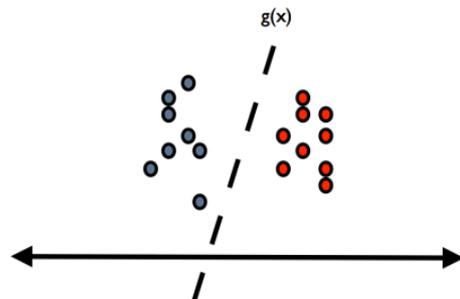
## Trois grandes approches pour l'apprentissage supervisé

### 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)



source : Arshia Cont

# 1- Introduction

## Trois grandes approches pour l'apprentissage supervisé

### Solution 3 : Approche par exemplification

- On possède une série d'exemples de couples assignant une observation  $X$  à une cible  $Y$  :

$$\mathbb{D}_{train} = \{(x_1, y_1), \dots, (x_N, t_N)\}$$

- pour une nouvelle observation  $x^*$ , on cherche les observations  $X$  de la base d'entraînement les plus proches de  $x^*$ ,
- on assigne à  $x^*$  le  $y$  correspondant aux  $X$  les plus proches
- Exemple : K-plus-proche-voisin

# 1- Introduction

## En résumé

Deux grands types d'apprentissage :

- **Supervisé**

- Deux grandes cibles :

- **Régression**

- **Classification**

- Trois grandes approches :

- Approche **générative** :

- nous partons de l'hypothèse qu'il existe une famille de modèles paramétriques permettant de générer  $X$  connaissant  $Y$

- Exemple : apprentissage Bayésien, modèle de Markov caché

- Approche **discriminante** :

- nous n'avons pas d'hypothèse sur le modèle sous-jacent à  $X$  mais nous étudions comment les séparer

- Exemple : analyse linéaire discriminante, machine à vecteur support (SVM), réseaux de neurones artificiels (ANN, Deep Learning)

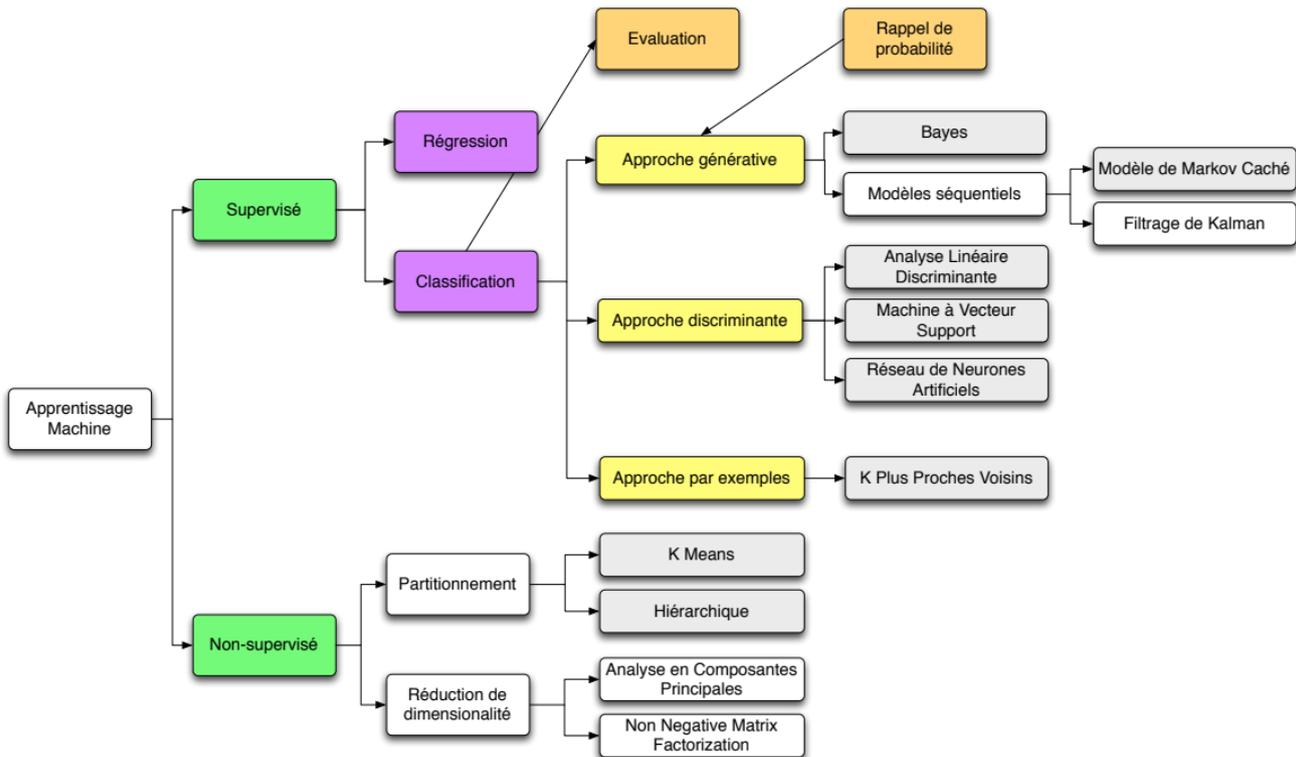
- Approche **par exemplification** :

- K-plus-proche-voisin

- **Non-supervisé**

# 1- Introduction

## En résumé



# 1- Introduction

## Généralisation

### Apprentissage

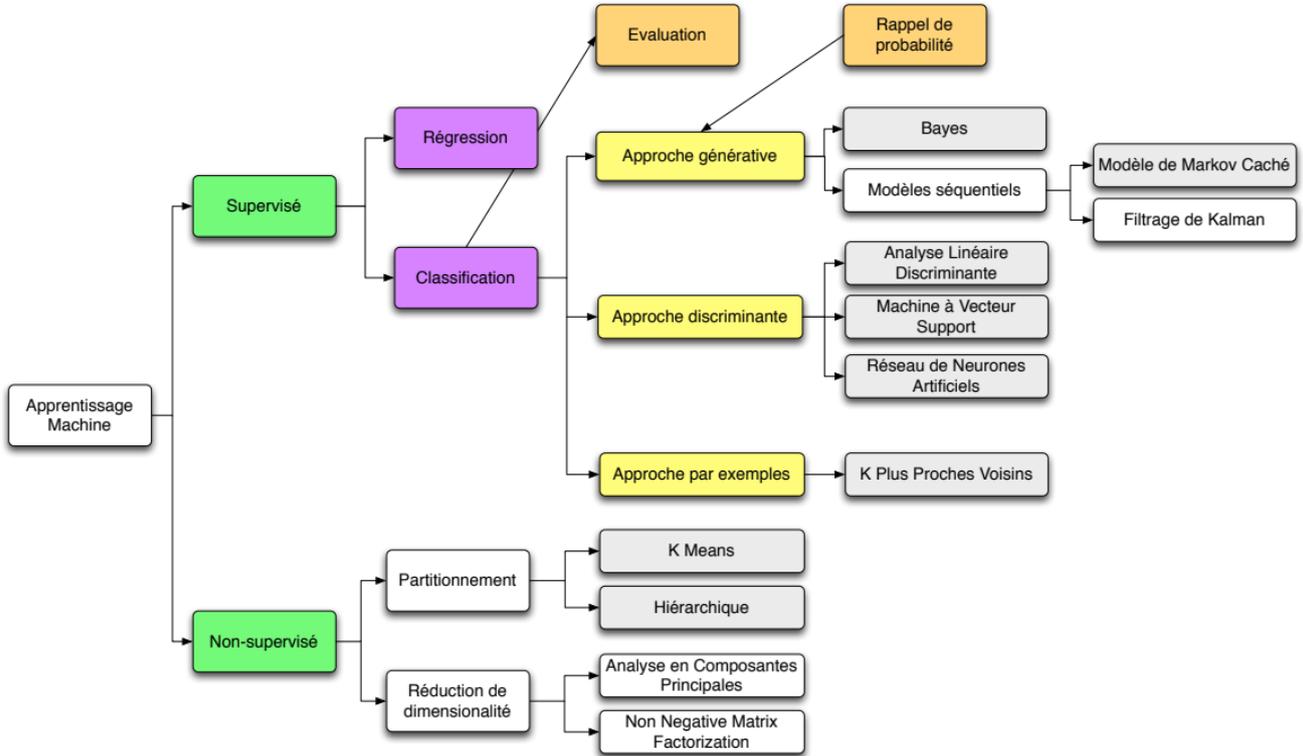
- Apprendre un modèle (génératif ou discriminant) à partir des observations  $X$  et des valeurs à prédire  $Y$
- Le modèle doit permettre une bonne prédiction de  $Y$  en fonction des observations  $X$

### Généralisation

- Capacité du modèle à prédire correctement des valeurs  $Y^*$  en fonction de  $X^*$  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'entraînement (training-set) :  $\{X, Y\}$
  - Ensemble de test (test-set) :  $\{X^*, Y^*\}$

# 1- Introduction

## Généralisation



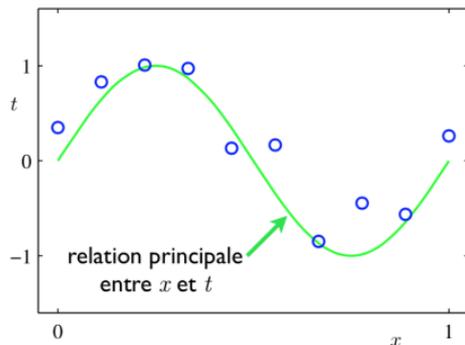
# Régression

## 2- Régression

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

On cherche à prédire une cible  $t_n$

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



source : Hugo Larochelle

## 2- Régression

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

#### Forme du modèle $y$ ?

- le modèle est un polynôme d'ordre  $M$  de coefficients  $w_m$

$$y(x, \mathbf{w}) = w_0 + w_1x + \dots + w_Mx^M = \sum_{m=0}^M w_mx^m$$

#### Estimation du modèle ?

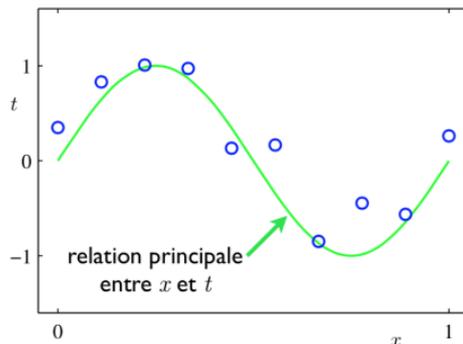
- = estimation des coefficients  $w_m$
- on note  $\mathbf{w}$  le vecteur  $[w_0, \dots, w_M]$
- on cherche le  $\mathbf{w}$  qui minimise l'erreur de prédiction sur l'ensemble d'entraînement  $\mathbb{D}_{train}$

- Erreur de prédiction sur une donnée  $t_n$  :

$$\bullet \epsilon(\mathbf{w}, n) = (y(x_n, \mathbf{w}) - t_n)^2$$

- Erreur de prédiction totale (sur l'ensemble des données d'entraînement) :

$$\bullet E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \epsilon(\mathbf{w}, n)$$



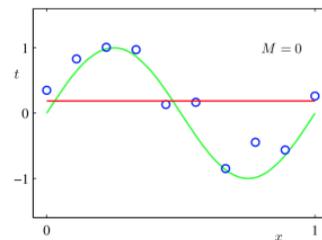
source : Hugo Larochelle

## 2- Régression

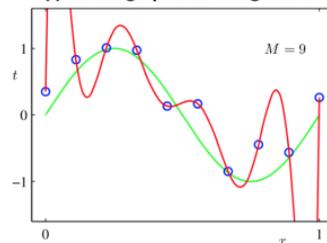
### Régression polynomiale en $D = 1$ dimension

#### Choix de l'ordre du polynôme $M$ ?

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



Sous-apprentissage (source : Hugo Larochelle)



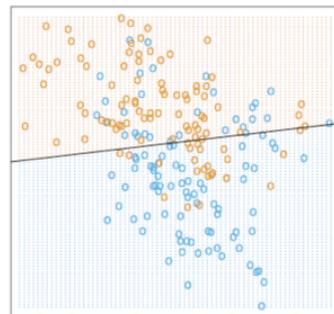
Sur-apprentissage (source : Hugo Larochelle)

## 2- Régression

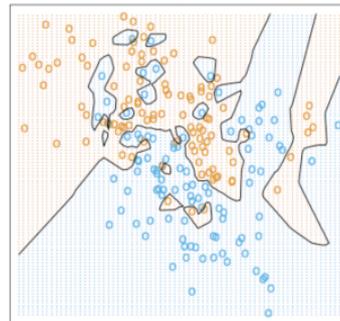
### Régression polynomiale en $D = 1$ dimension

#### Dans le cas de la classification

- **Sous-apprentissage** :
  - grande perte sur l'ensemble d'entraînement
- **Sur-apprentissage** :
  - apprentissage "par coeur" de l'ensemble d'entraînement



Sous-apprentissage (source : Arshia Cont)



Sur-apprentissage (source : Arshia Cont)

## 2- Régression

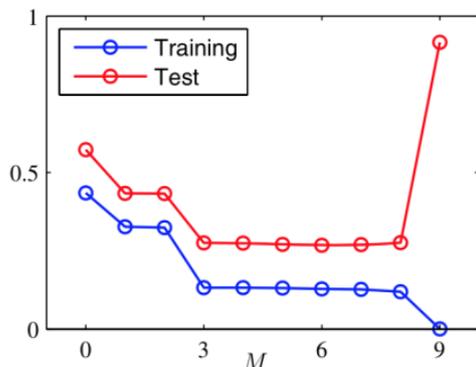
### Régression polynomiale en $D = 1$ dimension

#### 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

#### Capacité

- = Aptitude d'un modèle à apprendre "par coeur"
- plus  $M$  est grand,
  - plus le modèle a de capacité
- plus la capacité est grande,
  - plus la différence entre l'erreur d'entraînement et l'erreur de test augmente



source : Hugo Larochelle

## 2- Régression

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

Au plus on a de données d'entraînement, au plus le modèle entraîné va bien généraliser

#### Régularisation

- Objectif : utiliser un grand  $M$  avec peu de données
- Régularisation : on pénalise l'utilisation des paramètres  $w_m$ 
  - **Ridge** regression : on pénalise la norme  $L_2$  de  $\mathbf{w}$ 
    - $E(\mathbf{w}) = \sum_{n=1}^N (y(x_n, \mathbf{w}) - t_n)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$  avec  $\|\mathbf{w}\|_2^2 = w_0^2 + w_1^2 + \dots + w_M^2$
  - **Lasso** regression : on pénalise la norme  $L_1$  de  $\mathbf{w}$ 
    - $E(\mathbf{w}) = \sum_{n=1}^N (y(x_n, \mathbf{w}) - t_n)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_1$  avec  $\|\mathbf{w}\|_1 = |w_0| + |w_1| + \dots + |w_M|$

#### Sélection de modèle

- Choix de  $M$  et choix de  $\lambda$  ?
- $M$  et  $\lambda$  sont appelés des **hyper-paramètres**
- Comment déterminer les hyper-paramètres ? **sélection de modèles**
  - on rajoute une nouvelle base
    - base d'entraînement :  $\mathbb{D}_{train}$
    - **base de validation** :  $\mathbb{D}_{validation}$
    - base de test :  $\mathbb{D}_{test}$

## 2- Régression

### Régression polynomiale en $D > 1$ dimensions

#### Régression polynomiale en $D$ dimension

- $D$  dimensions ?
  - le nombre de dimensions de l'observation  $x$
  - exemple :
    - $D = 1$  : on considère l'intensité globale d'une image
    - $D = 100$  : on considère l'intensité de chaque pixel d'une image de (10,10)

- Pour un **polynôme d'ordre  $M = 3$**

- on a  $1 + D + D^2 + D^3$  paramètres à estimer
- pour  $D = 1$  ( $x = [x_i]$ ), on a **4 paramètres** à estimer

$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + w_3x^3$$

- pour  $D = 3$  ( $x = [x_i, x_j, x_k]$ ), on a **40 paramètres** à estimer

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

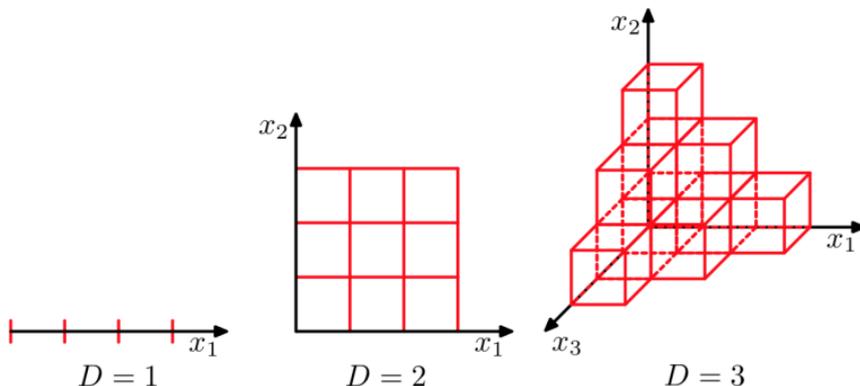
- pour  $D=100$  ( $x = [x_1, \dots, x_{100}]$ ), on a **1.010.101 paramètres** à estimer !!!

## 2- Régression

### Régression polynomiale en $D > 1$ dimensions

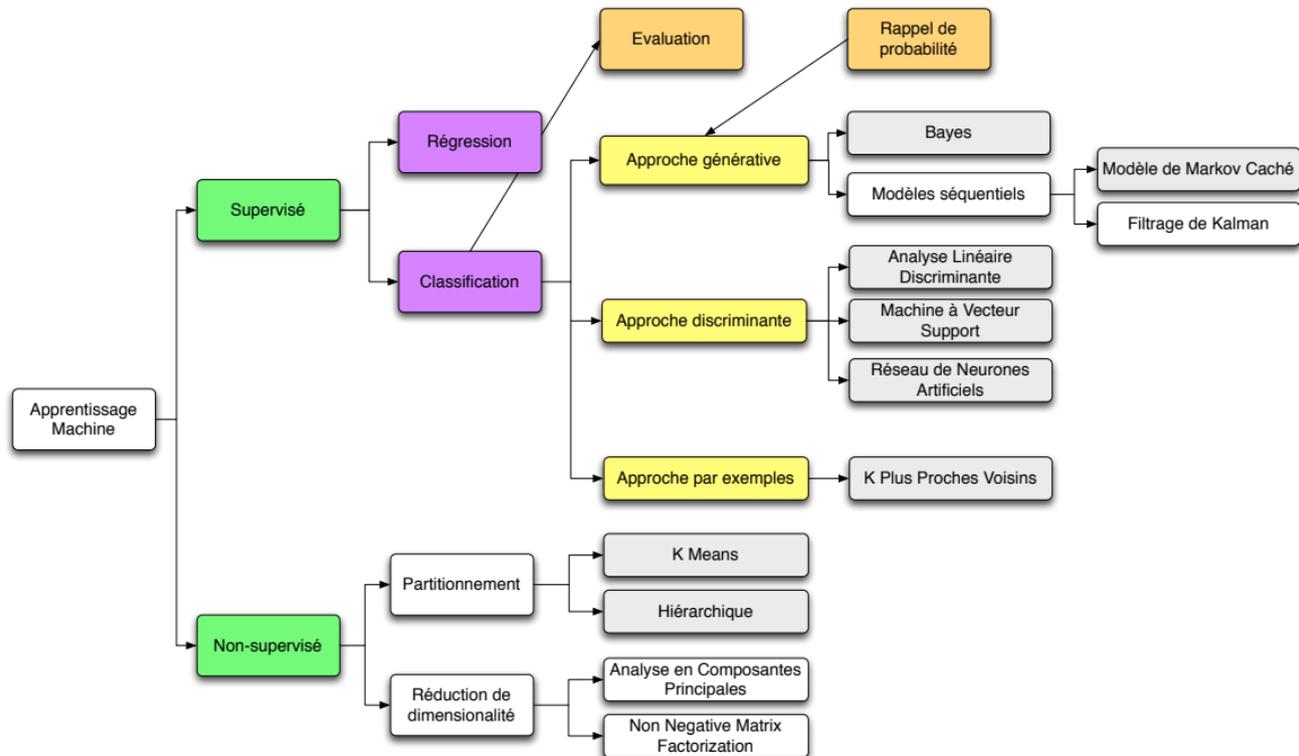
#### Malédiction de la dimension

- 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 (hypercubes)
  - quand  $D$  augmente, le nombre de régions augmente en  $O(3^D)$ 
    - Il devient impossible de garantir qu'on aura bien un exemple dans chaque région !!!



## 2- Régression

### Régression polynomiale en $D > 1$ dimensions



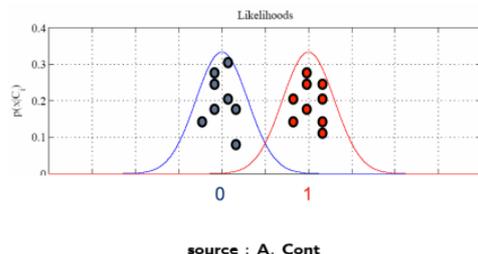
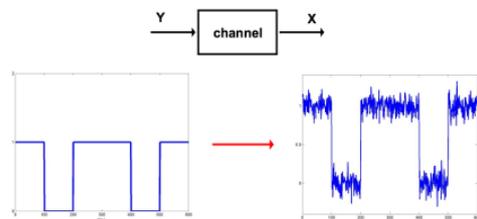
# Approche générative

# 3- Approche générative

## Décision Bayésienne

### Décision Bayésienne

- On souhaite trouver la classe  $Y = \{0, 1\}$  en fonction de  $x$
- En l'absence d'information, les deux probabilités sont équiprobables :
  - $p(Y = 0) = p(Y = 1) = 0.5$
  - $p(Y)$  est la **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 la **vraisemblance** (likelihood)



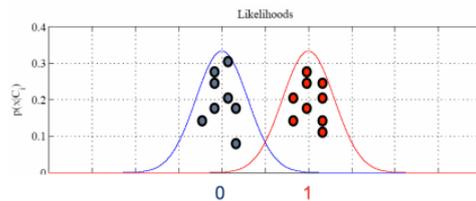
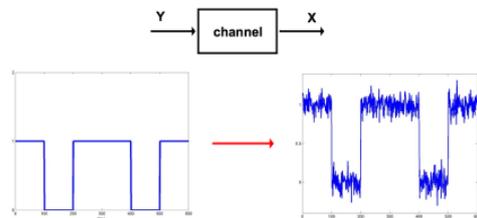
source : A. Cont

# 3- Approche générative

## Décision Bayésienne

### 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
  - $i^*(x) = \arg \max_i \{p(Y = y_i|X)\}$
- 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 Bayésienne



source : A. Cont

# 3- Approche générative

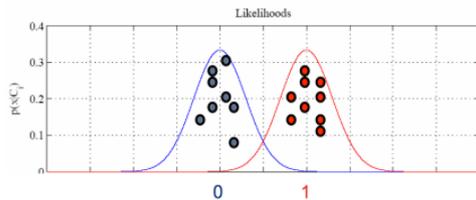
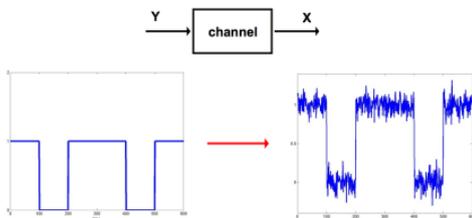
## Décision Bayésienne

### Inférence Bayésienne

- 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 Bayésienne :

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

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



source : A. Cont

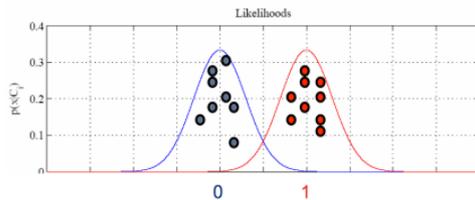
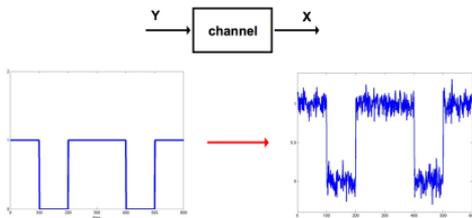
# 3- Approche générative

## Décision Bayésienne

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

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

$$i^*(x) = \arg \max_i \{\log(p(Y = y_i)) + \log(p(x|Y = y_i))\}$$



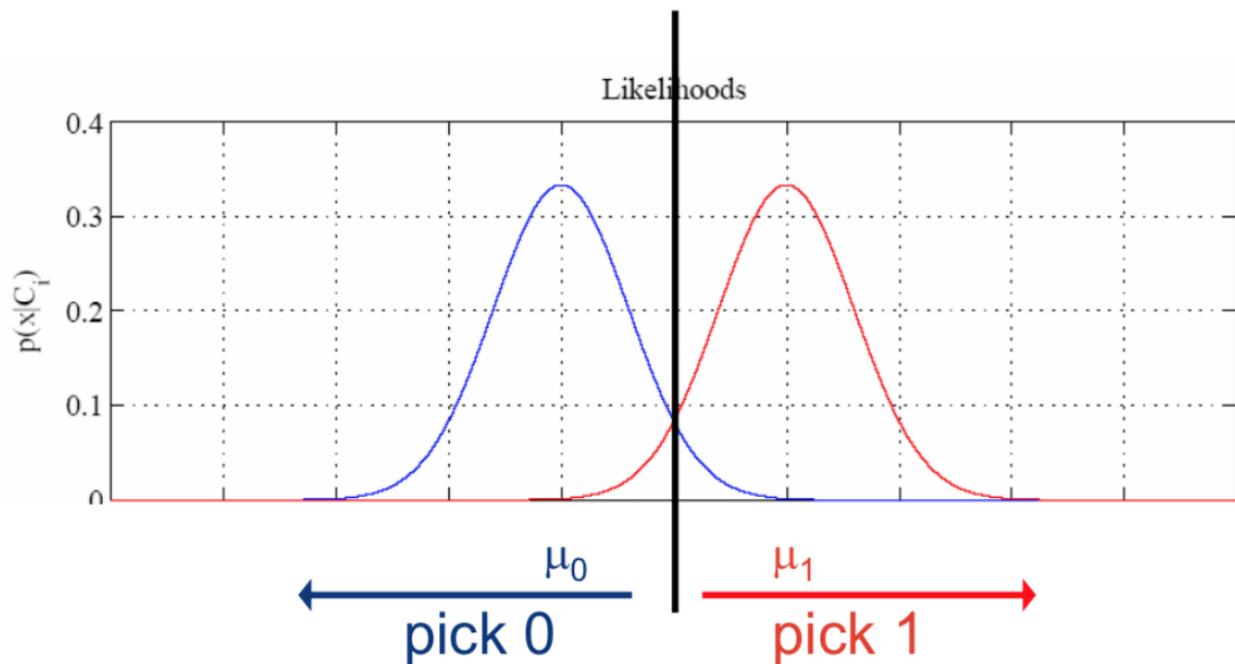
source : A. Cont

## Inférence Bayésienne dans le cas Gaussien : Cas simple

- Cas Simple ?
  - on considère toutes les classes **a priori** equi-probables :
    - $P(Y = 0) = P(Y = 1) = 0.5$
    - alors  $i^*(x) = \arg \max_i \{\log(p(x|Y = y_i))\}$
  - on considère  $D = 1$  et les variances égales ( $\sigma = \sigma_0 = \sigma_1$ ) alors
    - $P(X|Y = 0) \sim \mathcal{N}(\mu_0, \sigma)$
    - $P(X|Y = 1) \sim \mathcal{N}(\mu_1, \sigma)$
- On peut montrer que
  - $i^*(x) = \arg \min_i (w_i x + w_{i0})$ 
    - 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**  $g(x)$ 
  - $g(x) = \frac{\mu_1 + \mu_2}{2}$

### 3- Approche générative

#### Décision Bayésienne

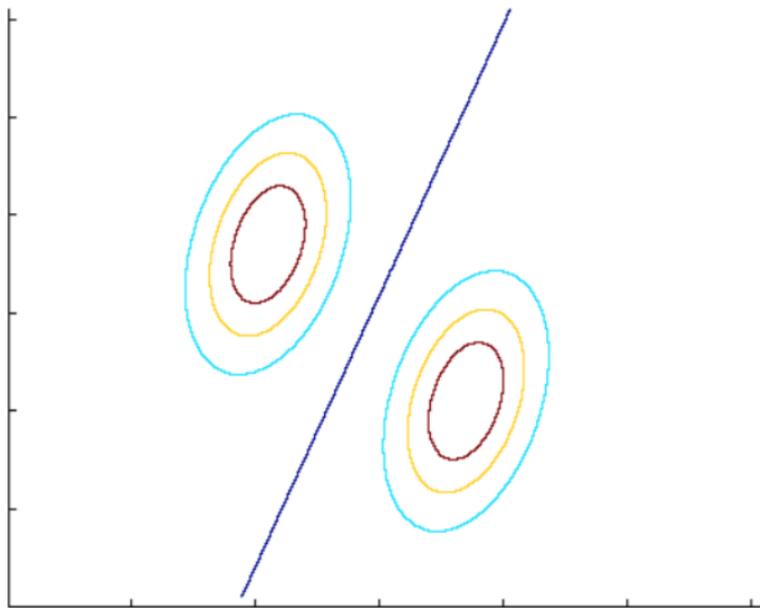


source : Arshia Cont

## Inférence Bayésienne dans le cas Gaussien : Cas général

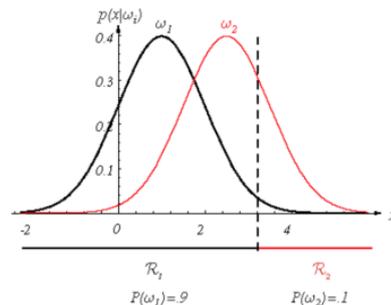
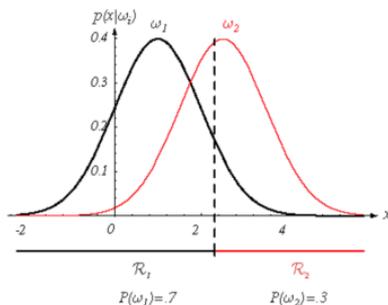
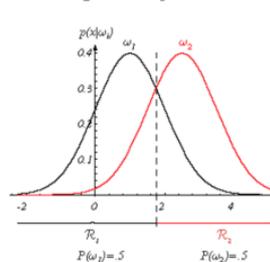
- Cas Général ?
  - on considère les classes **a priori** non equi-probables :
    - $P(Y = 0) \neq P(Y = 1)$
  - on considère  $D > 1$  et les matrices de co-variances égales ( $\Sigma = \Sigma_0 = \Sigma_1$ ) alors
    - $P(X|Y = 0) \sim \mathcal{N}(\mu_0, \Sigma)$  et
    - $P(X|Y = 1) \sim \mathcal{N}(\mu_1, \Sigma)$
- On peut montrer que
  - $i^*(x) = \arg \min_i \{ \mathbf{w}_i^T \mathbf{x} + \omega_{i0} \}$ 
    - avec  $\mathbf{w}_i = \Sigma^{-1} \mu_i$
    - avec  $\omega_{i0} = -\frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \log(P(Y = i))$
- On peut montrer que les classes  $i$  et  $j$  sont **equi-probables** pour  $\mathbf{x}$ 
  - $\mathbf{w}_i^T \mathbf{x} + \omega_{i0} = \mathbf{w}_j^T \mathbf{x} + \omega_{j0}$
- Ces valeurs de  $\mathbf{x}$  définissent la **frontière de décision**
  - $\mathbf{w}^T \mathbf{x} + w_0 = 0$ 
    - avec  $\mathbf{w} = \Sigma^{-1}(\mu_i - \mu_j)$
    - avec  $w_0 = -\frac{(\mu_i + \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j)}{2} + \log \frac{P(Y=i)}{P(Y=j)}$

*discriminant:*  
 $P_{Y|X}(1|\mathbf{x}) = 0.5$



### 3- Approche générative Décision Bayésienne

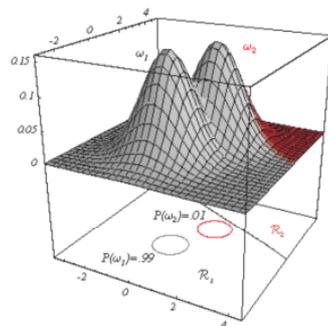
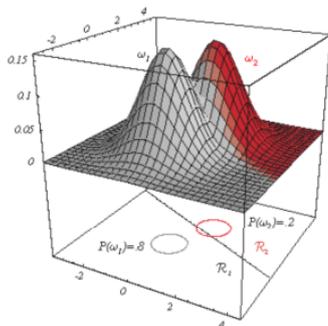
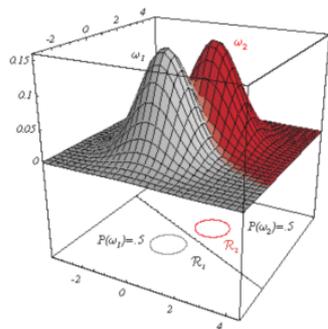
#### Illustration de l'influence de la probabilité a priori quand $D = 1$



source : Arshia Cont

### 3- Approche générative Décision Bayésienne

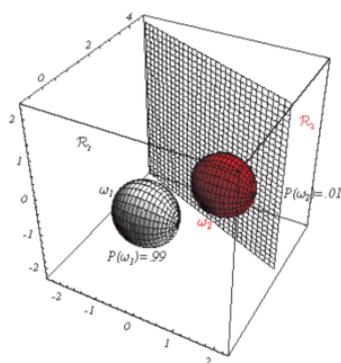
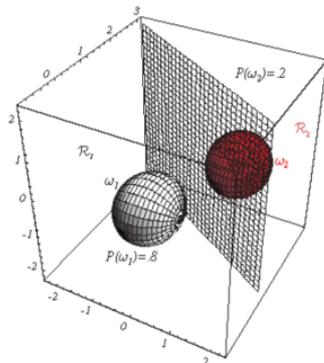
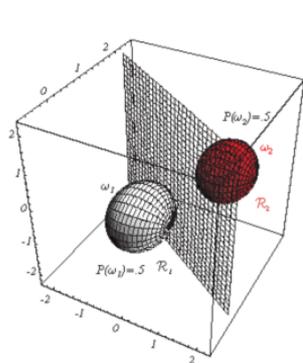
#### Illustration de l'influence de la probabilité a priori quand $D = 2$



source : Arshia Cont

### 3- Approche générative Décision Bayésienne

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



source : Arshia Cont

## 3- Approche générative

### Décision Bayésienne

#### Conclusion

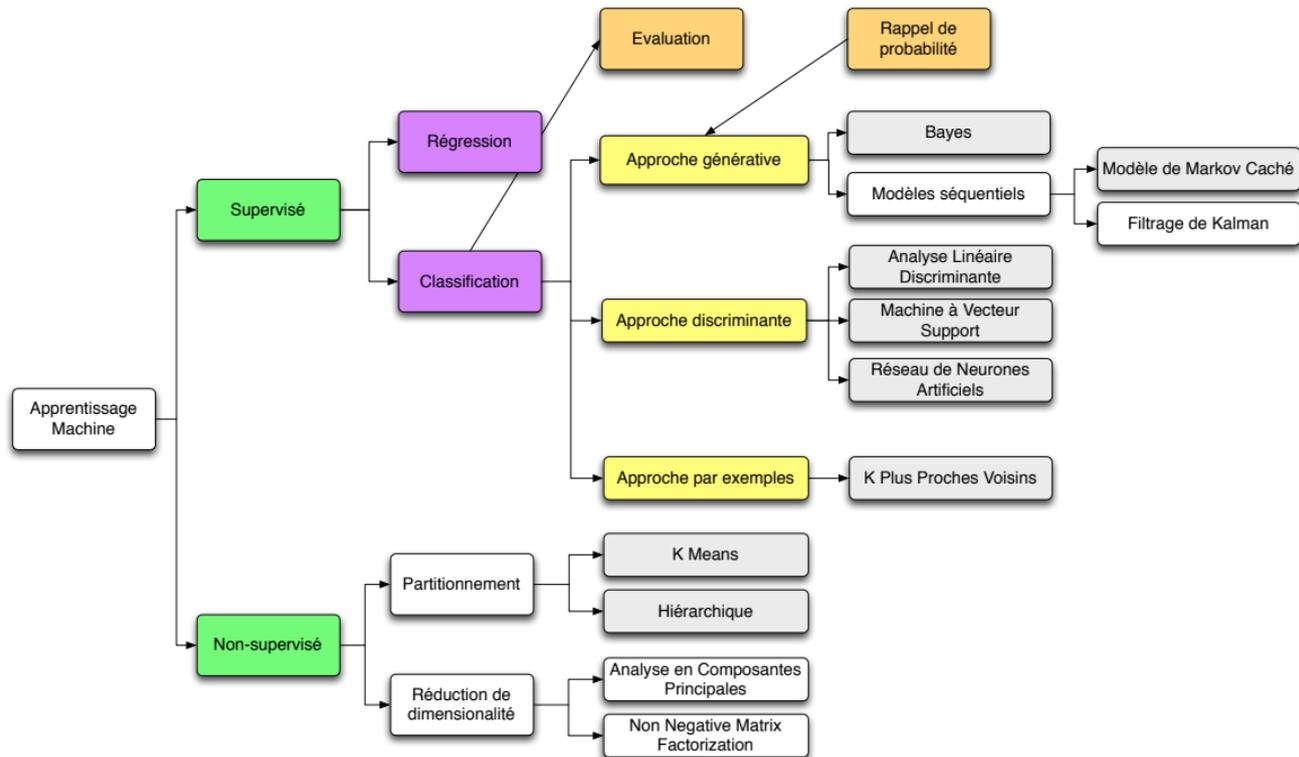
- Dans l'approche générative, nous apprenons les modèles  $P(X|Y)$  ayant engendré  $X$  en fonction de  $Y$
- Nous utilisons l'information a priori sur les classes  $Y : P(Y)$
- Nous combinons les deux pour obtenir une probabilité a posteriori  $P(Y|X)$  en utilisant la loi de Bayes
- La conséquence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

#### Approche discriminante

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

# 3- Approche générative

## Décision Bayésienne



# Approche générative- Modèles séquentiels

## 3- Approche générative

### Modèles séquentiels

#### Modèle de Markov

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



# 3- Approche générative

## Modèles séquentiels

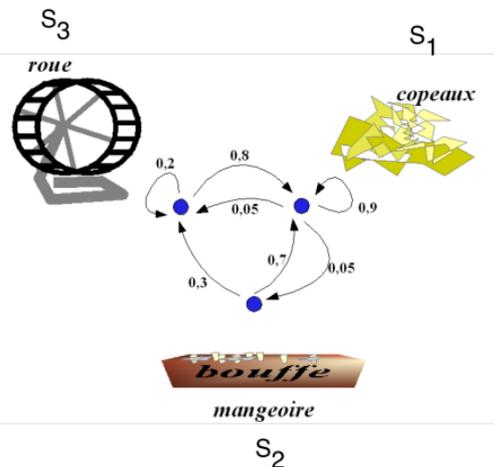
### Exemple : Doudou le hamster

- Doudou le hamster à 3 états dans sa journée :
  - il dort :
    - il est dans l'état  $S_1$  (copeaux)
  - il mange :
    - il est dans l'état  $S_2$  (mangeoire)
  - 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

$$T_{ij} = \begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.7^{(i=2,j=1)} & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{pmatrix} \quad (1)$$

$$T_{ij} = P(S_{t+1} = j | S_t = i) \quad (2)$$

$$\sum_i P(S_{t+1} = j | S_t = i) = 1 \quad (3)$$



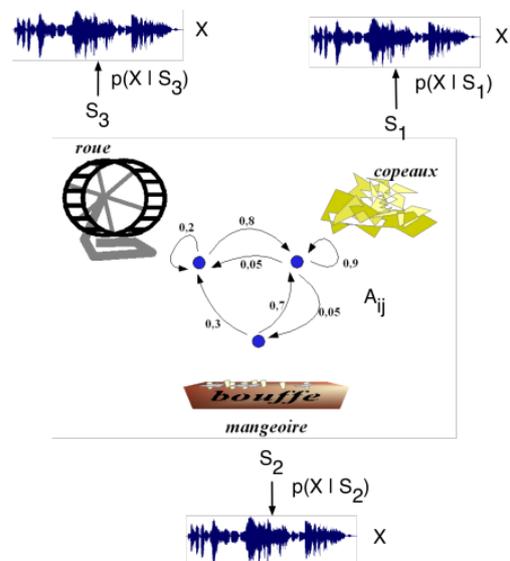
Modèle de Markov d'une journée de Doudou le hamster

# 3- Approche générative

## Modèles séquentiels

### Modèle de Markov **caché** ?

- Dans un modèle de Markov caché on observe pas directement les états  $S_i$ , ils sont "cachés"
- on observe 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 d'émission de  $X$  par  $S_i$  :  $p(X|S_i)$

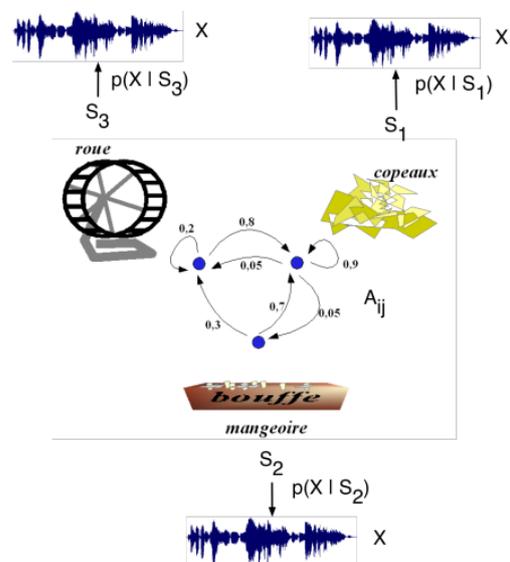


# 3- Approche générative

## Modèles séquentiels

### Définition d'un modèle de Markov caché

- Définition des **états**  $S_i$
- Définition des **observations**  $X$
- **Probabilités d'émission** de  $X$  par l'état  $S_i$  :
  - $A_i(X) = P(X_t = X | S_t = i)$
- **Probabilités de transiter** de  $S_i$  au temps  $t$  vers  $S_j$  au temps  $t + 1$  :
  - $T_{ij} = P(S_{t+1} = j | S_t = i)$
- La **probabilité qu'initialement** ( $t = 0$ ) le modèle se trouve dans l'état  $S_j$  :
  - $\pi_j = P(S_{t=0} = j)$
- On note  $\{\lambda\}$  l'ensemble de ces éléments



### 3- Approche générative

#### Modèles séquentiels

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  $S_t$  correspondant :  $S^* = \arg \max_S P(S|X, \lambda)$ 
  - 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 au modèle  $\{\lambda_2\}$  dormir/manger/travailler de Bill le salarié

- **Entraînement :**

- Etant donné une/des suites d'observations  $\{X_t\}$ , trouver le modèle  $\lambda^*$  qui maximise la vraisemblance des observations :  $\lambda^* = \arg \max_\lambda P(X|\lambda)$ 
  - Exemple : comment déterminer les paramètres  $\{\lambda\}$  du modèle de Doudou le hamster ?

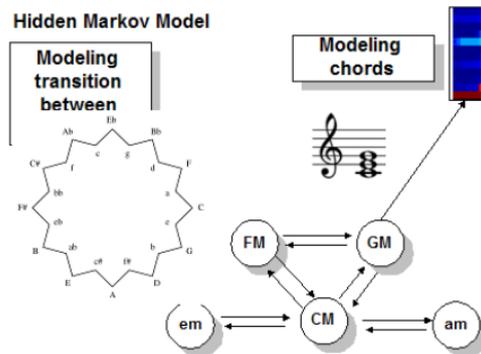
# 3- Approche générative

## Modèles séquentiels

### Exemple : La reconnaissance d'accords(1)

- **Objectif**
  - On veut estimer la suite d'accords  $\{C_M, C_{\#M}, \dots, C_m, \dots\}$  d'un morceau de musique à partir de l'observation de son signal audio
- Définition du modèle
  - **Etats**  $S_i =$ 
    - les différents accords  $\{C_M, C_{\#M}, \dots, C_m, \dots\}$
  - **Observations**  $X =$ 
    - un descripteur audio appelé Chroma ou Pitch Class Profile
  - **Probabilités d'émission** de  $X$  par l'état  $S_i =$ 
    - probabilité d'observer ces hauteurs de chroma pour cet accord

Hidden Markov Model



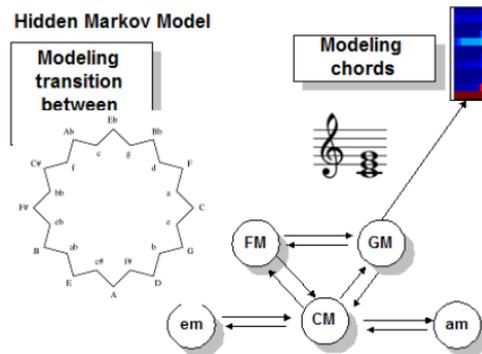
### 3- Approche générative

#### Modèles séquentiels

#### Exemple : La reconnaissance d'accords(2)

- **Probabilités de transiter** de  $S_i$  au temps  $t$  vers  $S_j$  au temps  $t + 1$  :
  - la matrice de transition entre accords respecte la théorie musicale
  - certains accords s'enchainent mieux ( $G_M$  vers  $C_M$  = consonance) que d'autres ( $G_M$  vers  $C\#_M$  = dissonance), cercle des quintes, relatifs majeur-mineur, ...
- **Solution**
  - On estime la suite d'accords par **décodage** d'un modèle de Markov Caché

Hidden Markov Model



# 3- Approche générative

## Modèles séquentiels

### Exemple : La reconnaissance de parole(1)

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

- 1) **Modèle de langage**
- 2) **Phonétiseur**
- 3) **Pré traitement audio**
- 4) **Modèle acoustique**

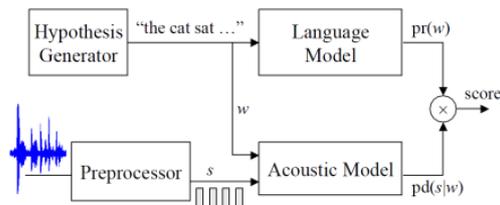


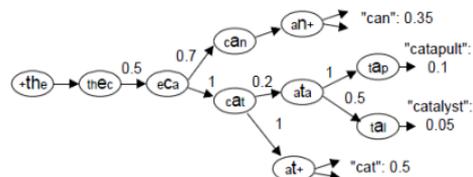
Schéma général d'un système de reconnaissance de parole  
(source : Mike Brookes)

# 3- Approche générative

## Modèles séquentiels

### Exemple : La reconnaissance de parole(2)

- 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)



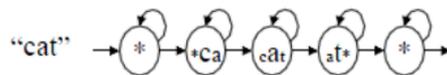
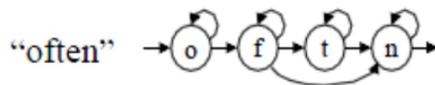
Modèle de langage (source : Mike Brookes)

## 3- Approche générative

### Modèles séquentiels

#### Exemple : La reconnaissance de parole(3)

- 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
    - **tri-phones**
      - 37<sup>3</sup> tri-phone en français

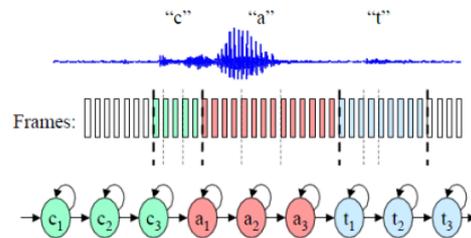


# 3- Approche générative

## Modèles séquentiels

### Exemple : La reconnaissance de parole(4)

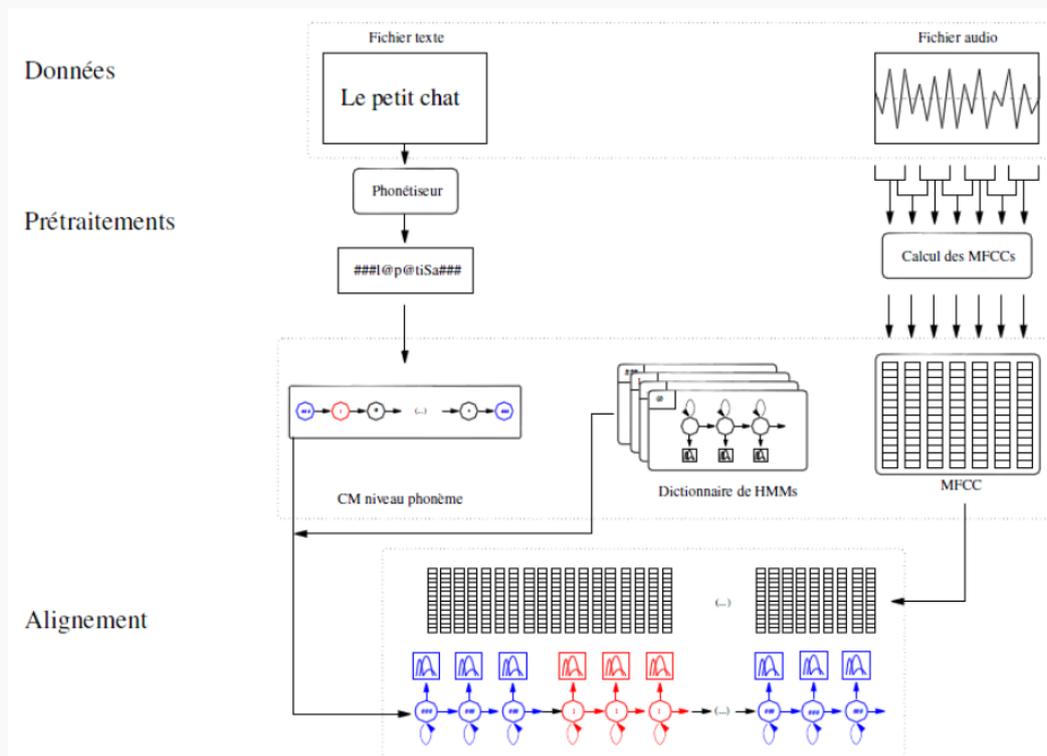
- 3) **Pré traitement audio** :
  - extrait des observations  $X$  pertinentes du signal audio
    - Généralement : MFCC +  $\Delta$  MFCC +  $\Delta\Delta$  MFCC (L=25ms, 40 bandes,  $3 \star 39$  coefficients)
- 4) **Modèle acoustique** :
  - définit un modèle de Markov caché permettant la jonction entre un phonème / tri-phonème et les différentes occurrences acoustiques (différents locuteurs)



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

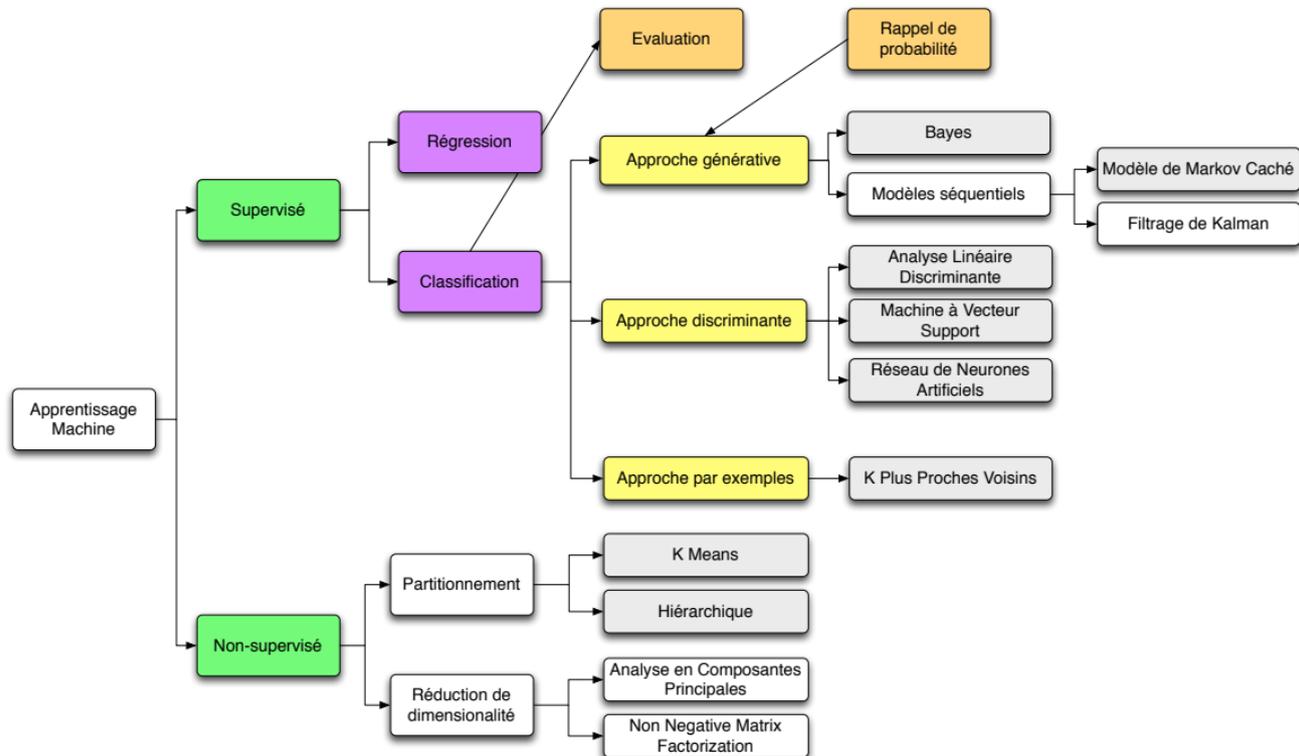
### 3- Approche générative Modèles séquentiels

#### Exemple : Alignement de parole à un texte



# 3- Approche générative

## Modèles séquentiels



# Approche discriminante

## 4- Approche discriminante

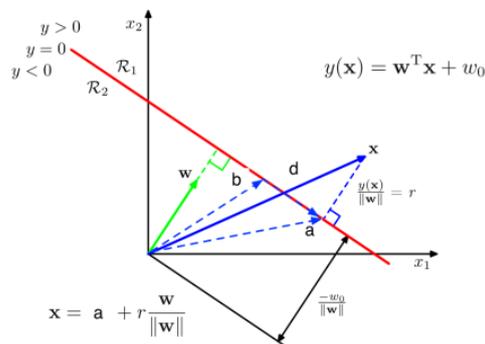
### Arbres de décision

## 4- Approche discriminante

### Frontière/surface de décision

#### Frontière/surface de décision

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



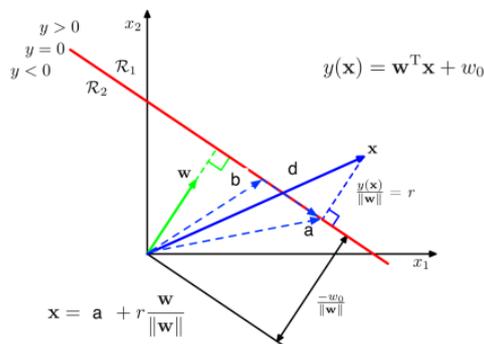
source : Hugo Larochelle

## 4- Approche discriminante

### Frontière/surface de décision

#### Frontière/surface de décision

- Equation :  $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$
- Surface de décision : ligne rouge
- La surface de décision est  $\perp$  à  $\mathbf{w}$
- **Preuve** :
  - $\mathbf{a}$  est sur la frontière donc  $y(\mathbf{a}) = 0$
  - on décompose  $\mathbf{a}$  comme  $\mathbf{a} = \mathbf{b} + \mathbf{d}$
  - $y(\mathbf{a}) = 0$ 
    - $= y(\mathbf{b} + \mathbf{d})$
    - $= \mathbf{w}^T (\mathbf{b} + \mathbf{d}) + w_0$
    - $= \mathbf{w}^T \mathbf{b} + w_0 + \mathbf{w}^T \mathbf{d}$
    - $= y(\mathbf{b}) + \mathbf{w}^T \mathbf{d}$ ,
    - $\mathbf{b}$  est sur la frontière donc  $y(\mathbf{b}) = 0$
    - $= \mathbf{w}^T \mathbf{d}$
    - donc  $\mathbf{w}$  est  $\perp$  à  $\mathbf{d}$  qui lui est sur la frontière

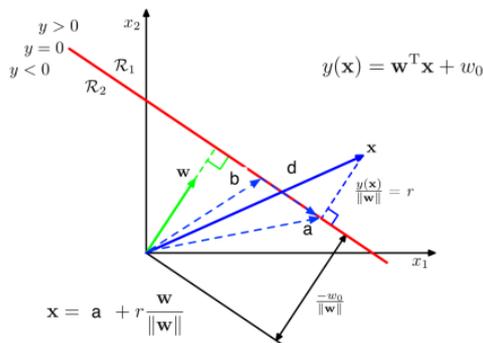


## 4- Approche discriminante

### Frontière/surface de décision

#### Frontière/surface de décision

- Equation :  $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$
- La distance entre  $\mathbf{x}$  et la frontière de décision est appelée  $r$
- $r$  est égale à  $r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$
- **Preuve :**
  - on décompose  $\mathbf{x}$  comme  $\mathbf{x} = \mathbf{a} + r \frac{\mathbf{w}}{\|\mathbf{w}\|}$
  - $y(\mathbf{x})$ 
    - $= y(\mathbf{a} + r \frac{\mathbf{w}}{\|\mathbf{w}\|})$
    - $= \mathbf{w}^T (\mathbf{a} + r \frac{\mathbf{w}}{\|\mathbf{w}\|}) + w_0$
    - $= \mathbf{w}^T \mathbf{a} + w_0 + r \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|}$
    - $= y(\mathbf{a}) + r \|\mathbf{w}\|$
    - $\mathbf{a}$  est sur la frontière donc  $y(\mathbf{a}) = 0$
    - $= r \|\mathbf{w}\|$



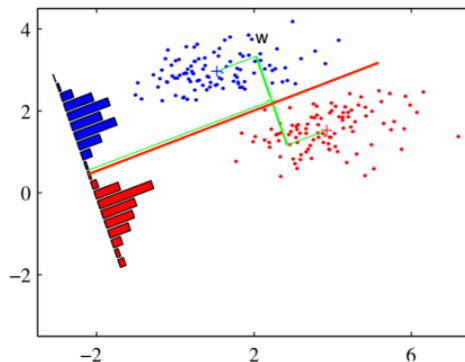


## 4- Approche discriminante

### Analyse Linéaire Discriminante (ALD)

#### Analyse Linéaire Discriminante (ALD)

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



source : Hugo Larochelle

## 4- Approche discriminante

### Analyse Linéaire Discriminante (ALD)

#### Analyse Linéaire Discriminante (ALD)

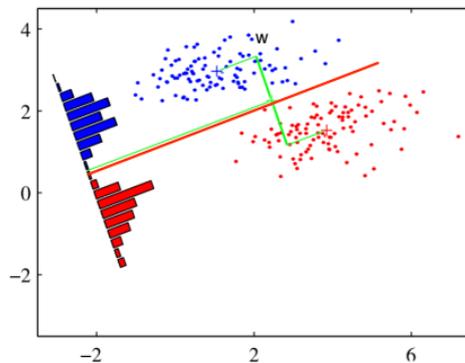
- Au total on maximise

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

**=  $\frac{\text{variance inter-classes}}{\text{variance intra-classe}}$**

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

$$S_w = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$$

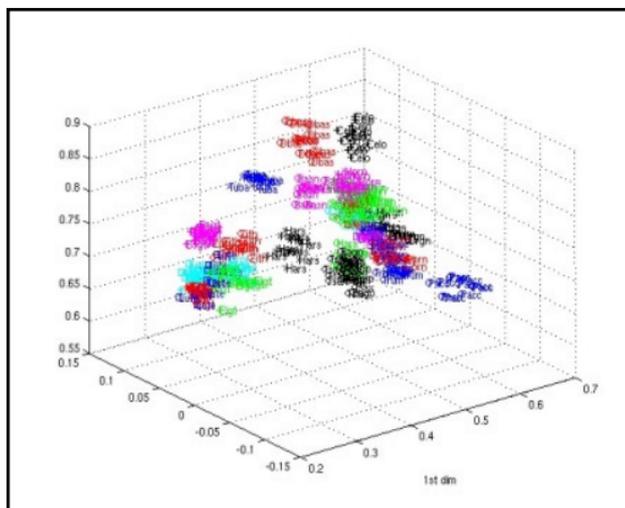
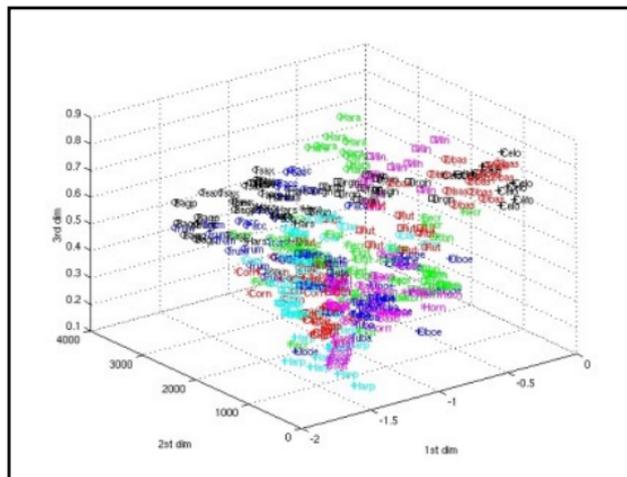


source : Hugo Larochelle

## 4- Approche discriminante

### Analyse Linéaire Discriminante (ALD)

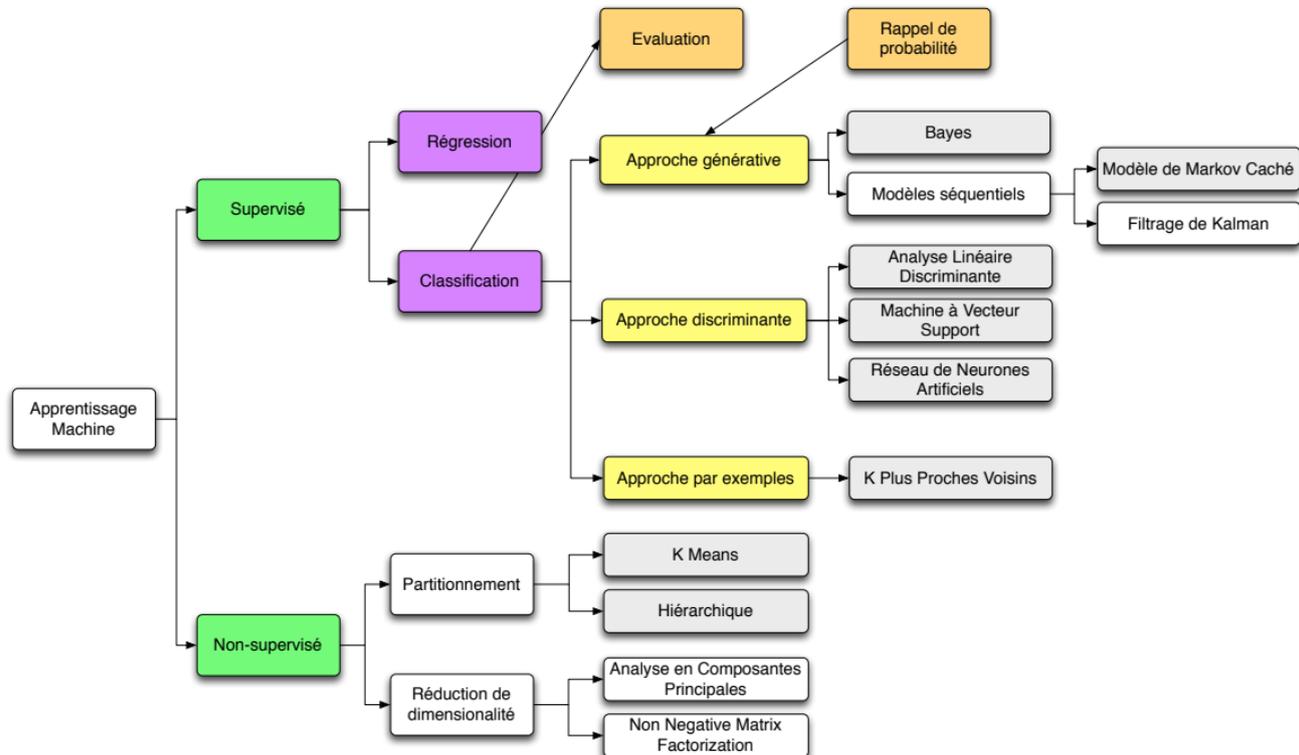
Exemple : application de l'ALD pour la reconnaissance des instruments



Apprentissage supervisé  
Approche générative  
Machine à Vecteurs Supports

# 4- Approche discriminante

## Machine à Vecteurs Supports (SVM)



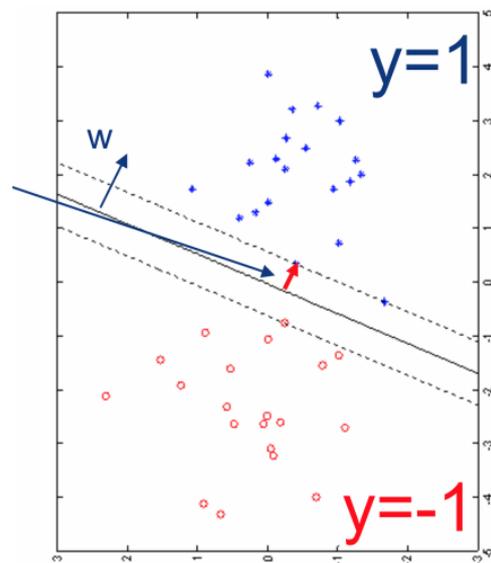


## 4- Approche discriminante

### Machine à Vecteurs Supports (SVM)

#### Maximisation d'une marge

- Il n'y aura pas d'erreur de classification si
  - $y_i \cdot (\mathbf{w}^T \mathbf{x}_i + w_0) > 0 \quad \forall i$
- Normalisation
  - pour éviter l'indétermination
  - on impose que la valeur de  $y$  au point le plus proche soit 1
    - $\min_i |\mathbf{w}^T \mathbf{x}_i + w_0| = 1$
    - donc  $\gamma = \frac{1}{\|\mathbf{w}\|}$
- Maximiser la marge  $\lambda$  revient à
  - minimiser  $\|\mathbf{w}\|$
- Il faut donc
  - $\min_{\mathbf{w}, w_0} \|\mathbf{w}\|^2$  sous la contrainte  $y_i \cdot (\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 \quad \forall i$

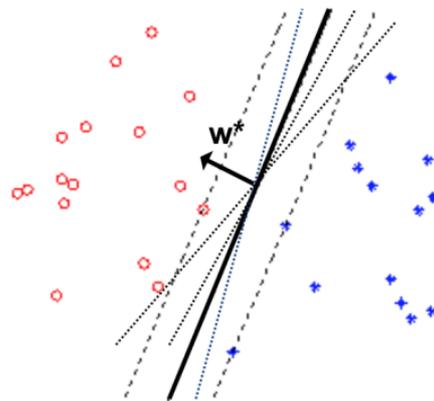


## 4- Approche discriminante

### Machine à Vecteurs Supports (SVM)

#### Maximisation d'une marge

- Parmi l'ensemble des hyper-plans  $w$  séparant les classes sans erreur
  - on cherche celui pour qui la marge  $\lambda$  (le sas de sécurité) est la plus grande
    - → celui pour lequel la généralisation sera la meilleure
  - celui qui a la complexité la plus réduite
    - on favorise les  $w$  avec beaucoup de coefficients à zéro



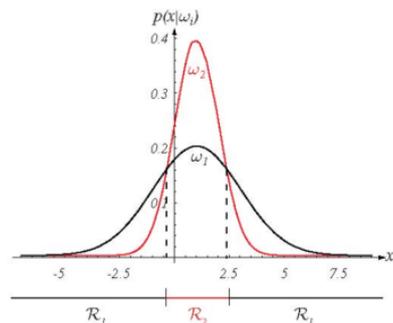
source : Arshia COnt

## 4- Approche discriminante

### Machine à Vecteurs Supports (SVM)

#### Cas de séparations non-linéaires

- Comment fait-on dans le cas suivant ?
  - Pas de séparation linéaire
- Il faudrait utiliser un modèle pour la frontière de décision d'ordre plus élevé
  - $\mathbf{x}^T \mathbf{W} \mathbf{x} + \mathbf{W}^T \mathbf{x} + \mathbf{w}_0 = 0$
- Besoin d'un polynôme d'ordre très élevé de façon général
  - Beaucoup de paramètres
  - Complexité très grande



source : Arshia Cont

## 4- Approche discriminante

### Machine à Vecteurs Supports (SVM)

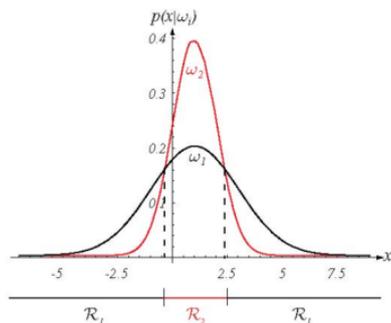
#### Cas de séparations non-linéaires

- Solution :
  - projeter les données dans un espace de dimension supérieur de manière à ce que dans cet espace les données soient linéairement séparables
    - $\phi : \mathcal{X} \rightarrow \mathcal{Z}$
    - $\dim(\mathcal{Z}) > \dim(\mathcal{X})$
  - Apprendre une frontière **linéaire** dans  $\mathcal{Z}$  est équivalent à apprendre une frontière **non-linéaire** dans  $\mathcal{X}$

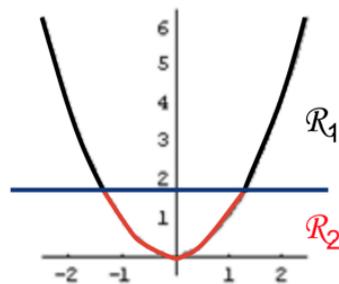
- Exemple :

$$\phi : \mathbb{R} \rightarrow \mathbb{R}^2$$
$$x \rightarrow (x, x^2)$$

- $\phi$  est appelé un **noyau** (kernel)



source : Arshia Cont



source : Arshia Cont

## 4- Approche discriminante

### Machine à Vecteurs Supports (SVM)

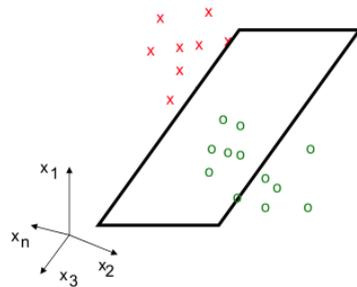
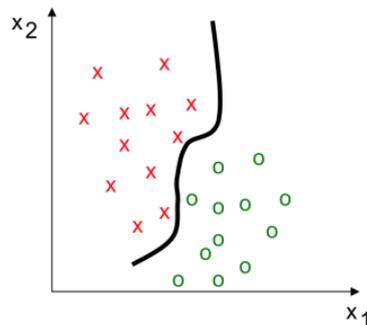
#### Cas de séparations non-linéaires

- Solution :
  - projeter les données dans un espace de dimension supérieur de manière à ce que dans cet espace les données soient linéairement séparables
    - $\phi : \mathcal{X} \rightarrow \mathcal{Z}$
    - $\dim(\mathcal{Z}) > \dim(\mathcal{X})$
  - Apprendre une frontière **linéaire** dans  $\mathcal{Z}$  est équivalent à apprendre une frontière **non-linéaire** dans  $\mathcal{X}$

- Exemple :

$$\phi : \mathbb{R} \rightarrow \mathbb{R}^2$$
$$x \rightarrow (x, x^2)$$

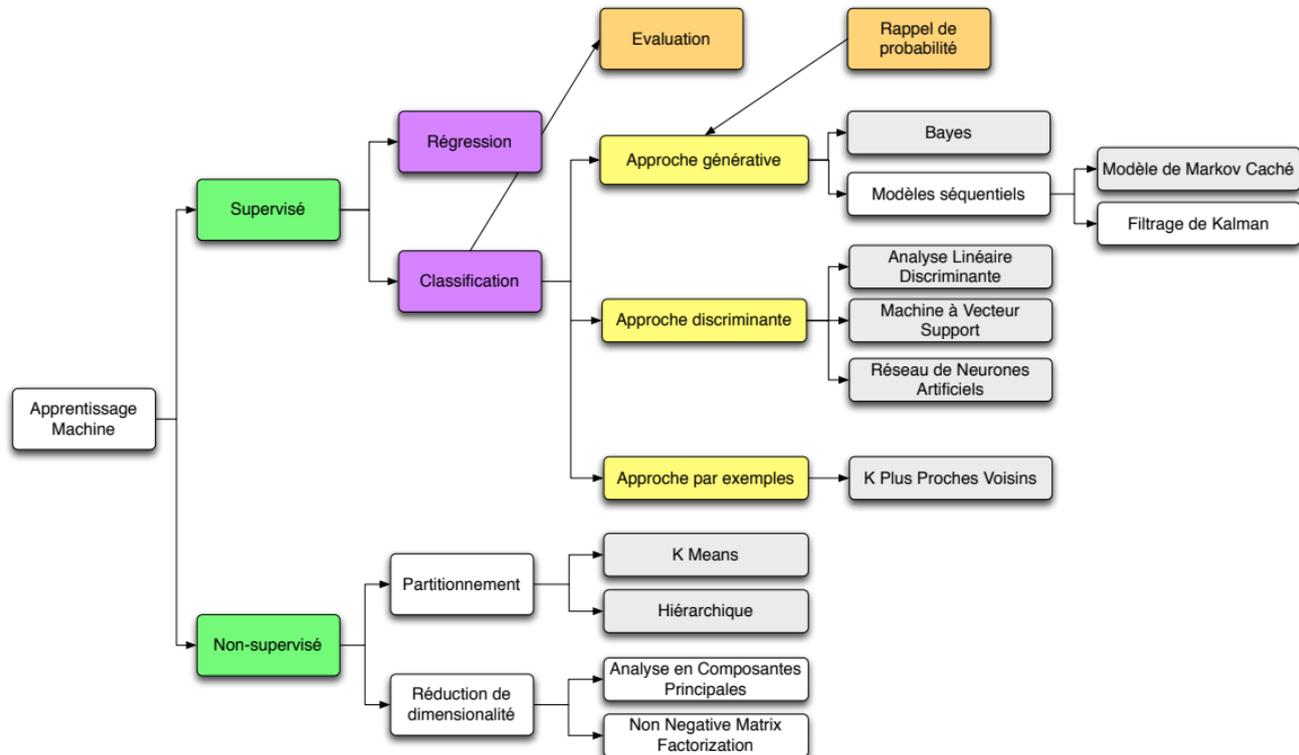
- $\phi$  est appelé un **noyau** (kernel)



source : Arshia Cont

# 4- Approche discriminante

## Machine à Vecteurs Supports (SVM)



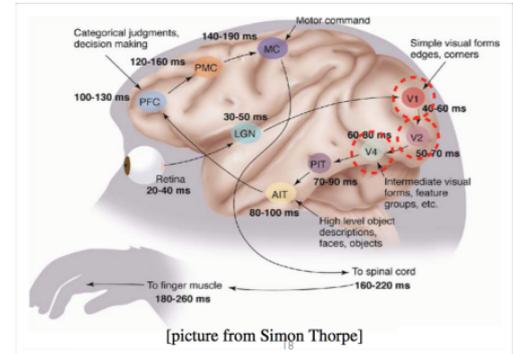
# Réseaux de neurones artificiels

## 4- Approche discriminante

### Réseaux de neurones artificiels

#### 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 traité par un réseau complexe de neurones inter-connectés
- Les neurones des différentes régions du cerveau sont spécialisés dans des traitements spécifiques



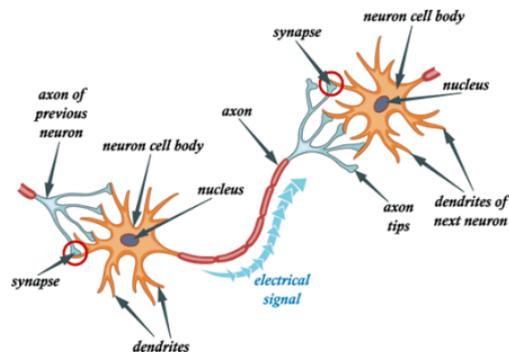
source : Hugo Larochelle

## 4- Approche discriminante

### Réseaux de neurones artificiels

#### Cerveau et neurones

- 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
- Chaque neurone
  - reçoit l'information à travers des **dendrites**
  - transforme l'information dans le corps de sa cellule (**soma**)
  - retourne un signal à travers un câble appelé **axon**
- le point de connection entre ce câble (axon) et les dendrites des autres neurones sont appelés **synapses**



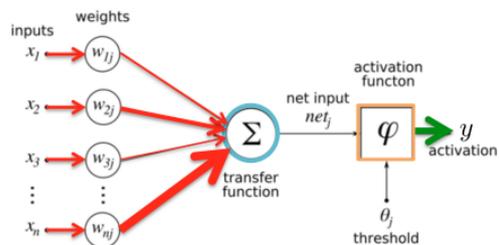
source : <http://biology.stackexchange.com/questions/25967/nerves-neurons-axons-and-dendrites-by-example>

## 4- Approche discriminante

### Réseaux de neurones artificiels

#### Réseau de neurones artificiels

- Un Réseau de Neurone Artificiel (Artificial Neural Network - ANN) reproduit l'interconnexion entre les différents neurones
- Chaque neurone est représenté par une fonction
  - prenant en **entrée** le signal des autres neurones
    - 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



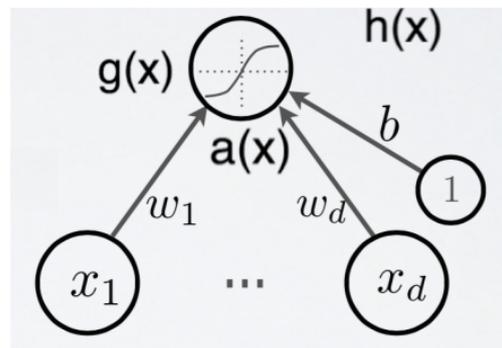
source : Ph. Esling

## 4- Approche discriminante

### Réseaux de neurones artificiels

#### Réseau de neurones artificiels

- Chaque neurone est représenté mathématiquement par
- 1 **Pre-activation** d'un neurone (activation des entrées)
  - $a(\mathbf{x}) = \sum_i w_i x_i + b = \mathbf{w}^T \mathbf{x} + b$ 
    - $\mathbf{w}$  sont les poids des connections (détermine quels neurones précédents apportent une information)
    - $b$  est le biais du neurone
- 2 **Activation** du neurone
  - $h(\mathbf{x}) = g(a(\mathbf{x})) = g(\mathbf{w}^T \mathbf{x} + b)$ 
    - $g$  est la fonction d'activation (généralement une fonction non-linéaire)



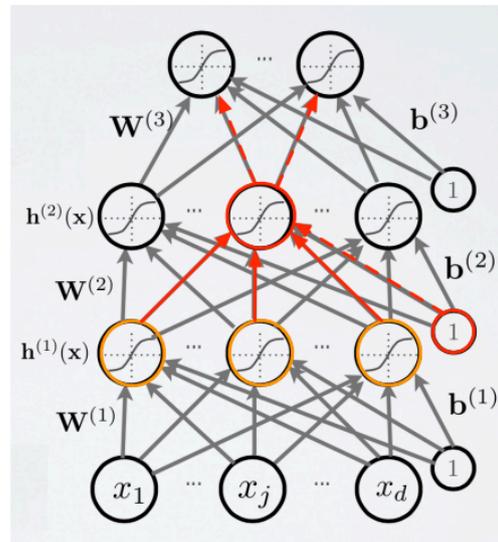
source : Hugo Larochelle

## 4- Approche discriminante

### Réseaux de neurones artificiels

#### Réseau de neurones multi-couches

- Les neurones artificiels sont organisés en couches (layer)
  - Multi-Layer-Perceptron (MLP)



source : Hugo Larochelle

## 4- Approche discriminante

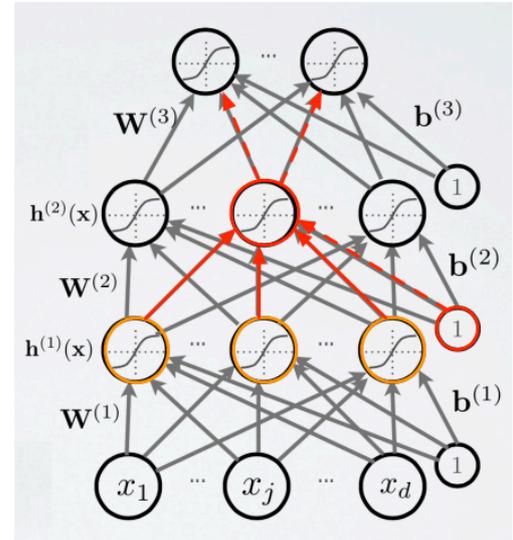
### Réseaux de neurones artificiels

#### Réseau de neurones multi-couches

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

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

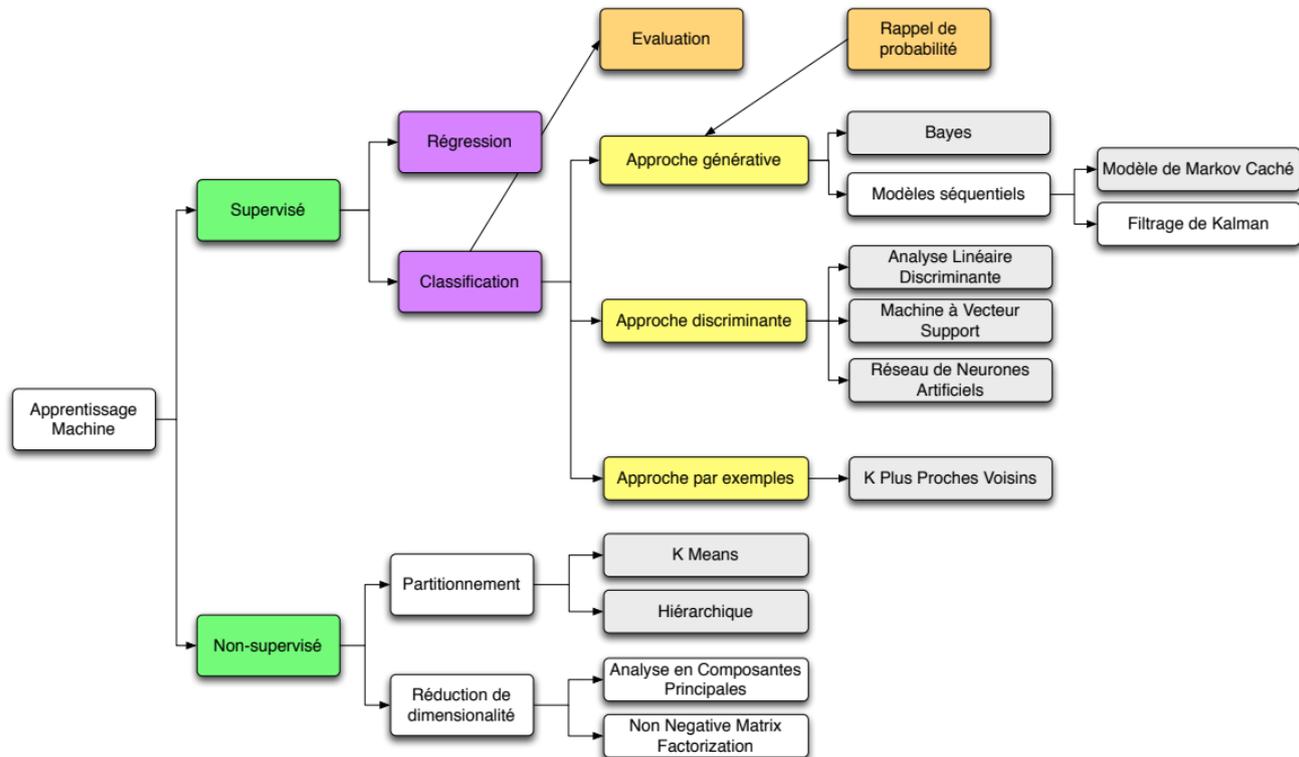
- beaucoup de paramètres à apprendre :  $\mathbf{w}^{(k)}, \underline{b}^{(k)}, \dots$
- on impose les valeurs à l'étage de sortie
- algorithme de **back propagation**
- descente de gradient



source : Hugo Larochelle

# 4- Approche discriminante

## Réseaux de neurones artificiels



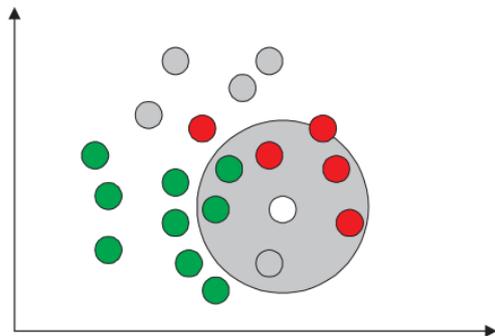
# Approche par exemplification

## 5- Approche par exemplification

### Algorithme des K Plus Proches Voisins

#### K Plus Proches Voisins

- Note : une observation  $\mathbf{x}_n$  est un point dans un espace à  $D$  dimensions, appelé espace des descripteurs
  - exemple :  $\mathbf{x}_n$  = la valeur des descripteurs audio à  $D$  dimensions
- **Entraînement** :
  - on remplit l'espace des descripteurs par l'ensemble des  $N$  "points" d'apprentissage :  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
  - à chaque point  $\mathbf{x}_n$  est associé sa classe  $t_n \in \{1, \dots, C\}$

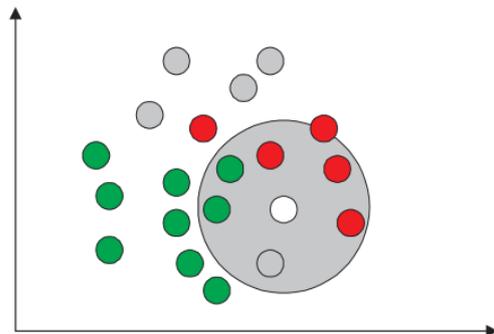


## 5- Approche par exemplification

### Algorithme des K Plus Proches Voisins

#### K Plus Proches Voisins

- **Evaluation :**
  - Soit  $x^*$  une observation nouvelle de classe  $t^*$  inconnue
  - On recherche dans l'espace des descripteurs les  $K$  points les plus proches de  $x^*$  selon une distance
    - généralement on utilise une distance euclidienne
  - On associe à  $x^*$  la classe majoritaire parmi celles assignées au  $K$  plus proche voisins  $\{t_k\}$ 
    - $t^* = \arg \max_{c \in \{1, \dots, C\}} \sum_{k=1}^K \delta(c, t_k)$

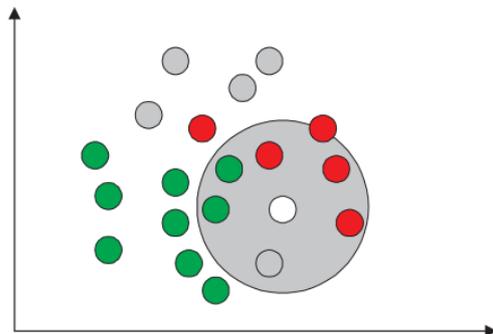


## 5- Approche par exemplification

### Algorithme des K Plus Proches Voisins

#### K Plus Proches Voisins

- **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  $x_m$  et tous les point  $x_k$ 
    - ce coût de calcul peut être très important



Algorithme 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.

## 5- Approche par exemplification

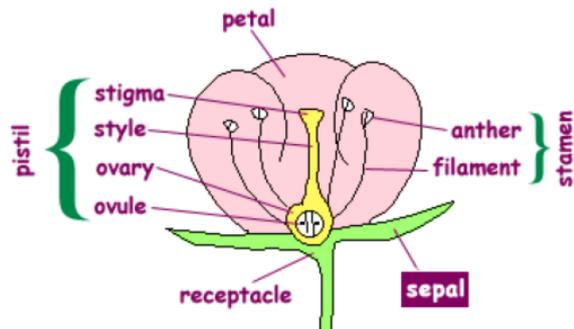
### Algorithme des K Plus Proches Voisins

## 6- Bases de référence pour l'entraînement et le test

### Base IRIS

#### Base IRIS

- $\mathbb{D}_{train,test} = \{(x_1, t_1), \dots, (x_N, t_N)\}$
- Observations  $\{x_1, \dots, x_N\}$ 
  - 1. sepal length in cm
  - 2. sepal width in cm
  - 3. petal length in cm
  - 4. petal width in cm
- Cibles  $\{t_1, \dots, t_N\}$ 
  - 1. Iris Setosa
  - 2. Iris Versicolour
  - 3. Iris Virginica



#### Base MNIST

- Task : handwritten digits recognition
- Images : 28x28 pixels grey-scale
- Training set : 60,000 examples,
- Test set : 10,000 examples



## 6- Bases de référence pour l'entraînement et le test

### Base CIFAR-10/ CIFAR-100

#### Base CIFAR-10/ CIFAR-100

- Task : image content recognition
- Images : 32x32 pixels colour
- CIFAR-10 : 10 classes (6000 images per class)
  - Training set : 50000 exemples
  - Test set : 10000 exemples
- CIFAR-100 : 100 classes (600 images per class) grouped into 20 superclasses
  - Training set : 50000 exemples
  - Test set : 10000 exemples

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



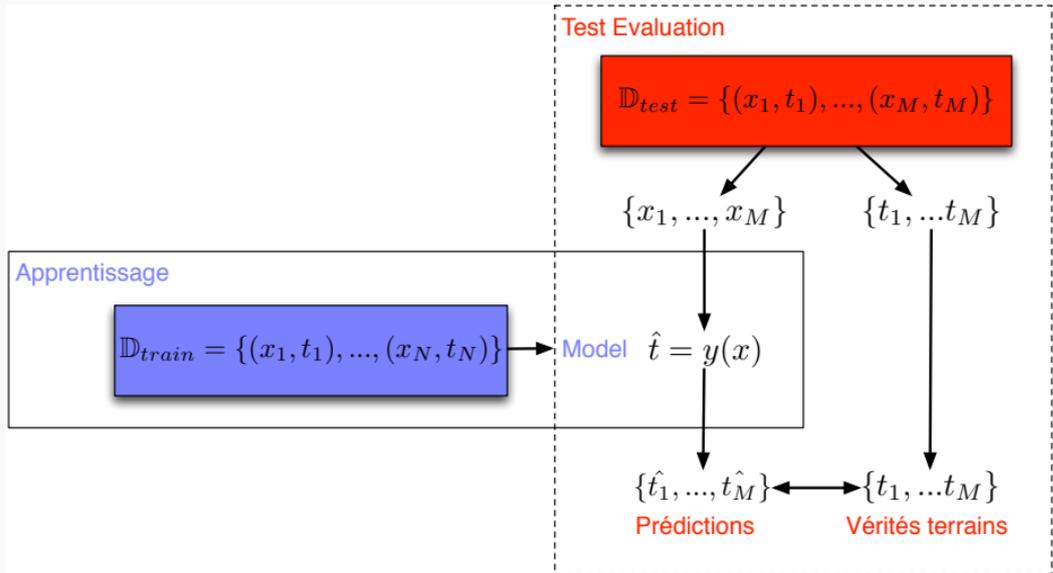
# Evaluation d'un système de classification

# 7- Evaluation d'un système de classification

## Indices de performance

### Evaluation des performances d'un système de classification

- L'ensemble de test  $\mathbb{D}_{test}$  doit être différent de l'ensemble d'entraînement  $\mathbb{D}_{train}$ 
  - On test la généralisation du modèle  $y(x)$
- On compare les prédictions de classes  $\hat{t}_m$  aux "vérités terrains" de classes  $t_m$ 
  - Comment compare-t'on ?



## 7- Evaluation d'un système de classification

### Indices de performance

#### Calcul des TP, FP, TN, FN

- Pour deux classes ( $c_1=+$ ,  $c_2=-$ )
- **True Positif** (TP) : # de données + détectées correctement (True) comme +
- **False Negatif** (FN) : # de données + détectées faussement (False) comme -
- **False Positif** (FP) : # de données - détectées faussement (False) comme +
- **True Negatif** (TN) : # de données - détectées correctement (True) comme -

#### Matrice de confusion

		Prédiction	
		Positif (Cancer)	Négatif (NoCancer)
Vérité	Positif (Cancer)	TP	FN
	Négatif (NoCancer)	FP	TN

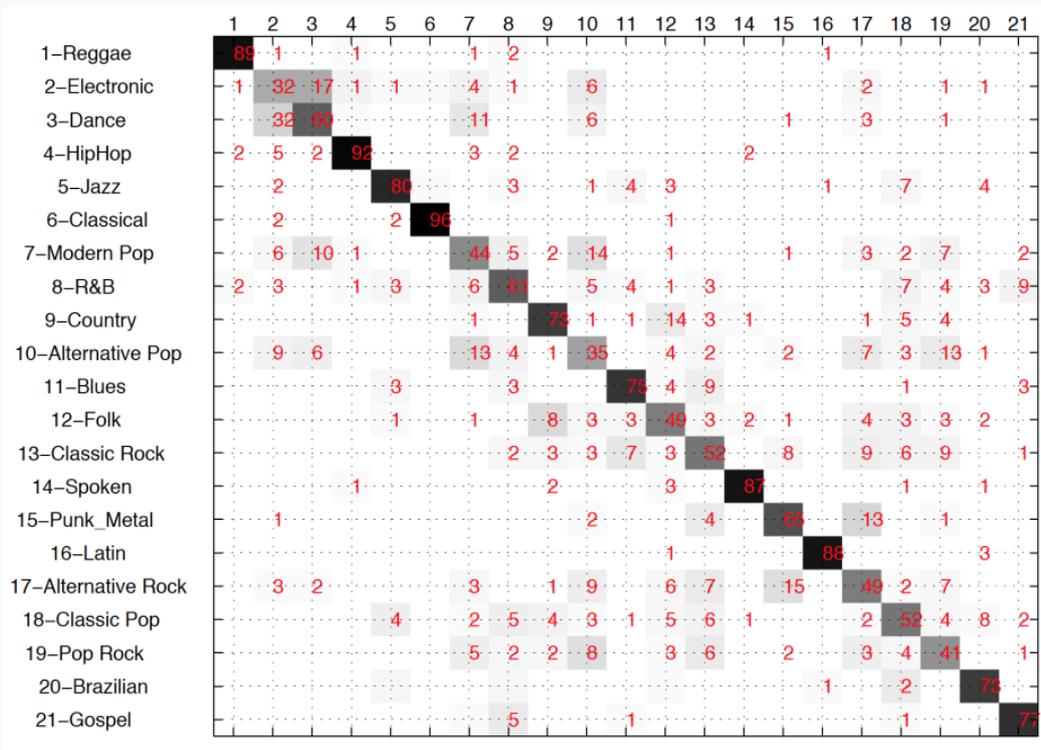
**Rappel** (encadré vert)

**Precision** (encadré rouge)

# 7- Evaluation d'un système de classification

## Indices de performance

### Exemple de matrice de confusion



## 7- Evaluation d'un système de classification

### Indices de performance

#### Calcul des indices résumés

$$\begin{aligned} \text{Rappel}(\text{Recall}) &= \text{Mesure la capacité à retrouver tous les } c_i \\ &= \frac{\# \text{ données détectées comme } c_i \text{ et étant réellement } c_i}{\# \text{ données étant réellement } c_i} \\ &= \frac{TP}{TP + FN} \end{aligned}$$

$$\begin{aligned} \text{Precision} &= \text{Mesure la capacité à retrouver uniquement des } c_i \text{ (moteur de recherche)} \\ &= \frac{\# \text{ données détectées comme } c_i \text{ et étant réellement } c_i}{\# \text{ données détectées comme } c_i \text{ (correctes ou fausses)}} \\ &= \frac{TP}{TP + FP} \end{aligned}$$

*F – measure* = Prend en compte simultanément le Rappel et la Précision

$$\begin{aligned} \text{Accuracy} &= \text{Mesure les performances globales indépendamment de la distribution des classes} \\ &= \frac{TP + TN}{TP + FP + TN + FN} \end{aligned}$$

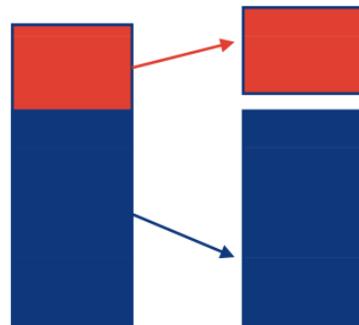
(4)

# 7- Evaluation d'un système de classification

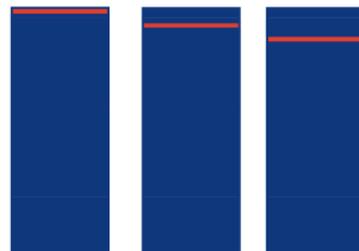
## Séparer ensemble d'entraînement et de test

### Séparer ensemble d'entraînement et de test

- 1. Deux ensembles séparés
- 2. Un ensemble unique
  - 2.1 **N-Fold Cross Validation** (validation à N-plis croisés)
    - $\mathbb{D}$  est divisé en  $N$  sous-ensemble  $\mathbb{D}_n$ 
      - test : on choisit un sous-ensemble
      - entraînement : on utilise les  $N - 1$  autres
    - on réitère en choisissant un nouveau sous-ensemble de test parmi les  $N$  possibles
    - on calcul la moyenne des indices
  - 2.2. **Leave-one-out Cross-Validation** :
    - cas limite quand  $N = \text{le nombre de données}$ 
      - test : on choisit une donnée
      - entraînement : on utilise toutes les  $N - 1$  autres données
    - on réitère ...
    - on calcul la moyenne des indices



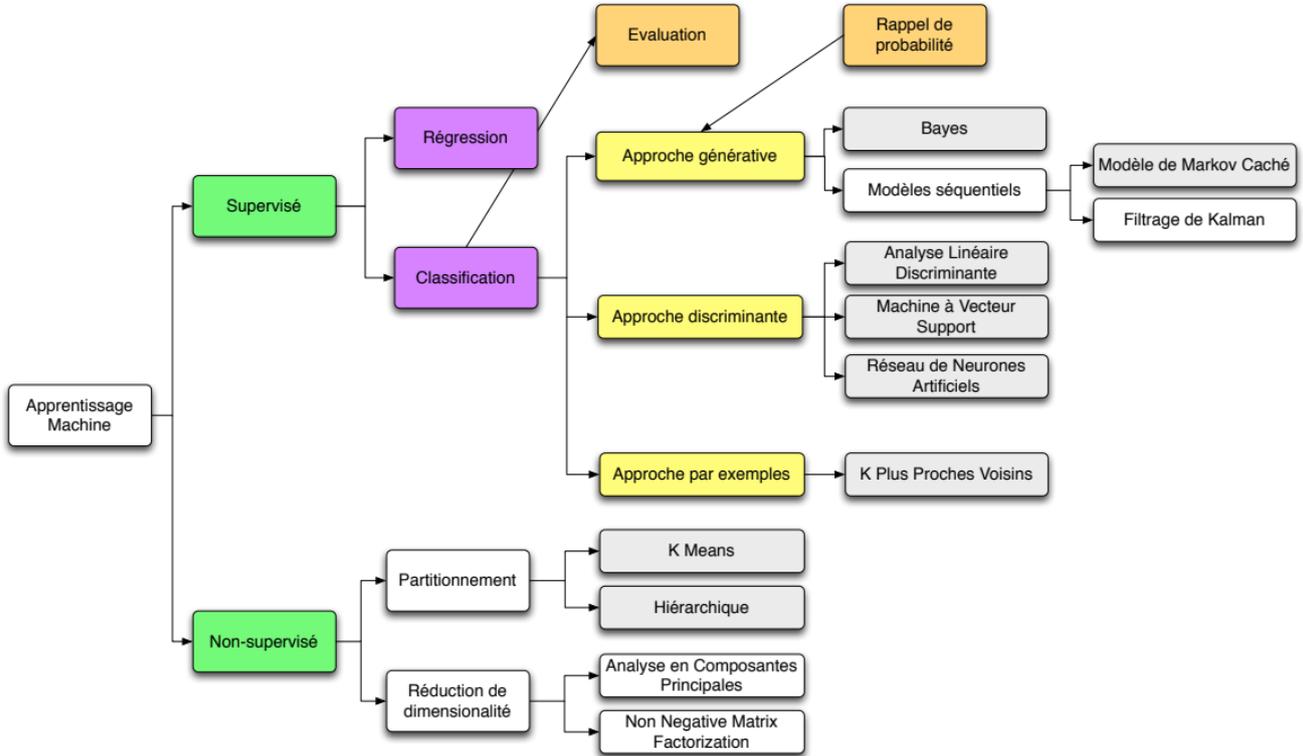
source : Arshia Cont



source : Arshia Cont

# 7- Evaluation d'un système de classification

Séparer ensemble d'entraînement et de test



# Apprentissage non-supervisé

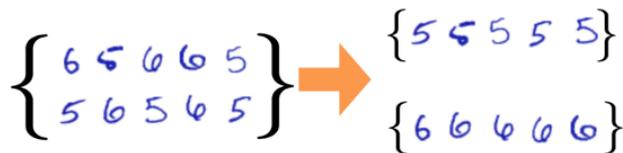
# 8- Apprentissage non-supervisé

## Introduction

Nous considérons un ensemble d'**observations** (entrées du système)  $\{x_1, \dots, x_N\}$

### Apprentissage non-supervisé

- Nous ne donnons pas à la machine les cibles
- $D = \{x_1, \dots, x_N\}$
- L'objectif de la machine est de créer un modèle de  $x$ , un partitionnement (clustering) des données
  - peut être utilisé pour analyser les données, prendre des décisions



source : Hugo Larochelle

# 8- Apprentissage non-supervisé

## Introduction

### Apprentissage non-supervisé = Algorithme de clustering

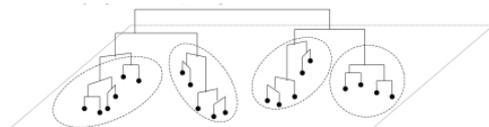
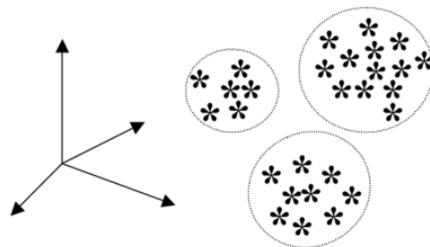
- Algorithme de clustering
  - divise un groupe hétérogène de données, en sous-groupes de manière que
    - les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire
    - les données considérées comme différentes se retrouvent dans d'autres groupes distincts
- **Objectif**
  - trouver la structure sous-jacente à un ensemble de données

# 8- Apprentissage non-supervisé

## Introduction

### Grandes classes d'algorithmes

- Méthodes **non-hiérarchiques**
  - on considère toutes les données simultanément
  - exemple : k-means
- Méthodes **hiérarchiques**
  - **ascendantes** :
    - on **agglomère** progressivement les données deux à deux
    - exemple : algorithme agglomératif hiérarchique
  - **descendantes** :
    - on **divise** à chaque étage les données en deux groupes
    - exemple : KD-tree



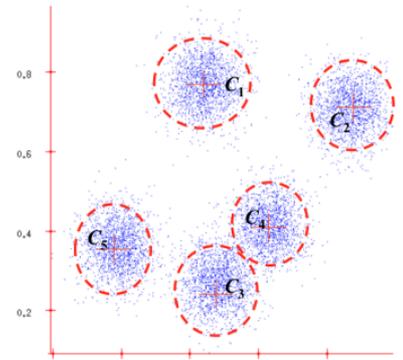
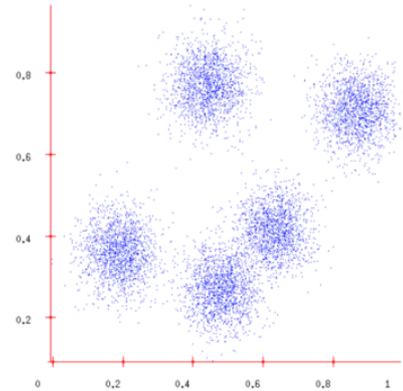
source : Christine Decaestecker, Marco Saerens

## 8- Apprentissage non-supervisé

### Algorithmes des K-Means (nuées dynamiques)

#### Algorithmes des K-Means (nuées dynamiques)

- Méthode non-hiérarchique
- On cherche à **organiser** l'ensemble de données  $\mathbb{D} = \{x_1, \dots, x_N\}$  en  $K$  **ensembles** (clusters)
- On cherche à **représenter** l'ensemble de données  $\mathbb{D} = \{x_1, \dots, x_N\}$  par les **prototypes**  $\mu_k$  des  $K$  clusters
- **Assignment** :
  - $r_{nk}=1$  si donnée  $n$  appartient au cluster  $k$
  - $r_{nk}=0$  sinon
- **Erreur commise** :
  - $J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$
- **Objectif** : minimiser l'erreur
- **Comment** : algorithme Expectation-Maximization (EM)

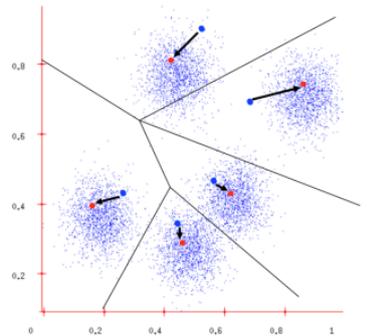
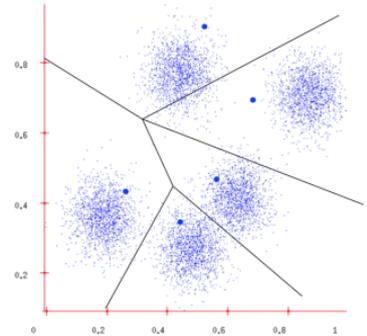


# 8- Apprentissage non-supervisé

## Algorithmes des K-Means (nuées dynamiques)

### Algorithme Expectation-Maximisation (EM)

- **Initialisation :**
  - choisir  $K$  clusters (random, KD-tree)
  - calculer le centroïde  $\mu_k$  de chaque cluster à partir des objets attribués à  $k$
  - $$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$
- **Etape E (Expectation) :**
  - attribuer chaque objet au cluster dont il est le plus proche (distance euclidienne)
  - $r_{nk} = 1$  si  $k = \arg \min_j \|x_n - \mu_k\|^2$ ,
  - $r_{nk} = 0$  sinon
- **Etape M (Maximization) :**
  - étant donné la nouvelle attribution des objets aux clusters, recalculer les centroïdes
  - $$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$
- **Itération :**
  - répéter jusqu'à ce que l'erreur ne se réduise plus

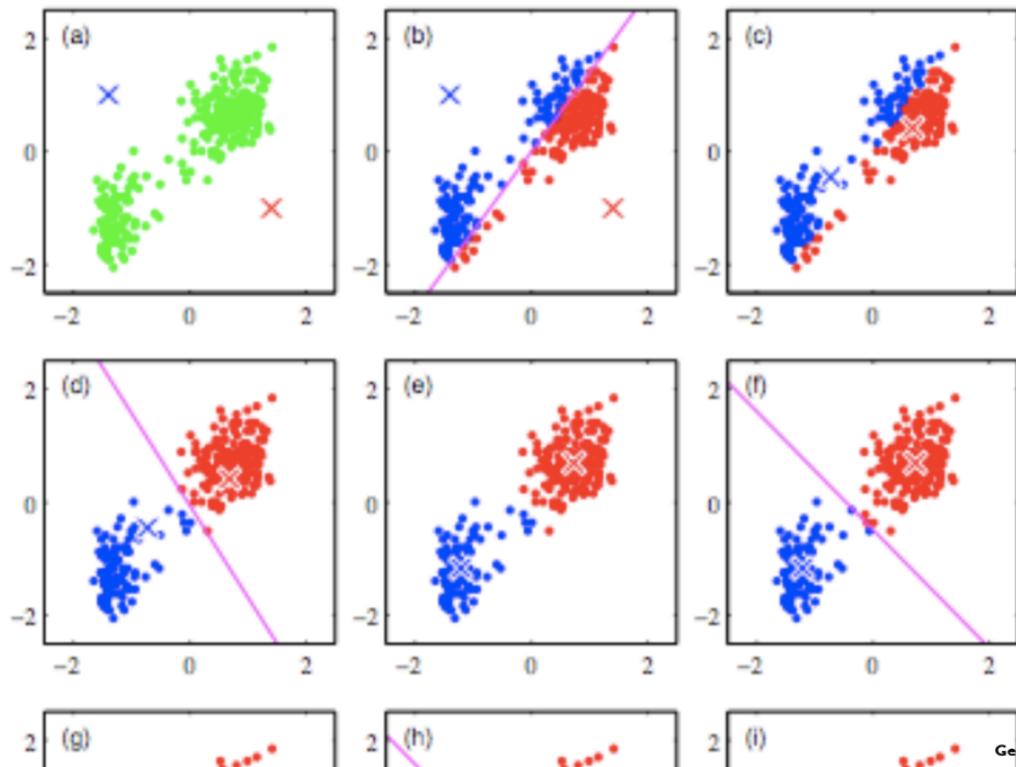


source : Philippe Esling

## 8- Apprentissage non-supervisé

### Algorithmes des K-Means (nuées dynamiques)

#### Algorithmes des K-Means

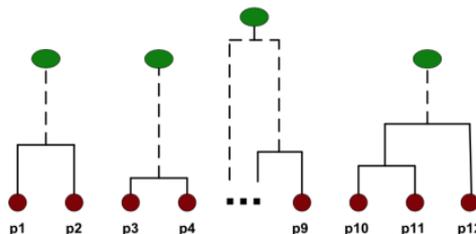
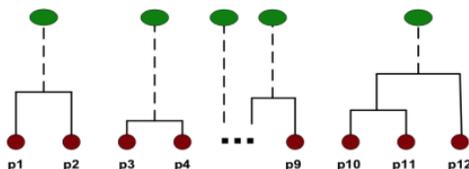


# 8- Apprentissage non-supervisé

## Algorithme hiérarchique par agglomération

### Algorithme hiérarchique par agglomération

- Méthode hiérarchique (ascendante par agglomération)
- **Initialisation :**
  - chaque objet constitue un cluster
- **Itération :**
  - regroupement des deux objets les plus proches
    - **distance** entre objets
      - euclidienne, Minkowski, cosin
    - ou d'un objet à un cluster
    - ou des deux clusters les plus proches
      - **linkage** entre objets
- **Condition d'arrêt :**
  - on arrive au sommet de l'arbre, ou bien on a obtenu K clusters



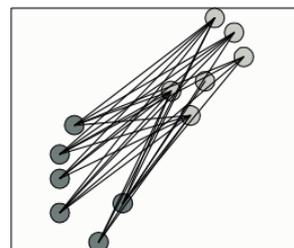
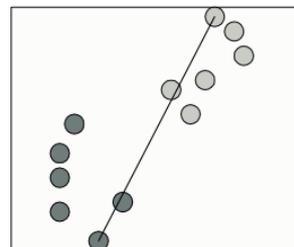
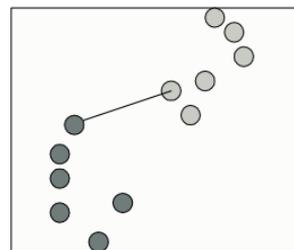
source : Philippe Esling

## 8- Apprentissage non-supervisé

### Algorithme hiérarchique par agglomération

#### Linkage entre objets d'un cluster

- **Single :**
  - plus petite distance entre les objets dans les deux groupes —  
 $d(r, s) = \min(\text{dist}(r_i, s_j)) \quad i \in r \quad j \in s$
- **Complete :**
  - plus grande ...
  - $d(r, s) = \max(\text{dist}(r_i, s_j)) \quad i \in r \quad j \in s$
- **Average :**
  - moyenne des distances entre toutes les paires d'objets entre les deux groupes
  - $d(r, s) = \text{mean}(\text{dist}(r_i, s_j)) \quad i \in r \quad j \in s$
- **Centroid :**
  - distance entre les centroides des deux groupes
  - $d(r, s) = \text{dist}(\bar{r}, \bar{s})$
- **Ward :**
  - représente l'augmentation de l'inertie intra-groupe due à la réunion des groupes  $r$  et  $s$
  - $d(r, s) = \frac{n_r n_s}{n_r + n_s} \text{dist}(\bar{r}, \bar{s})$

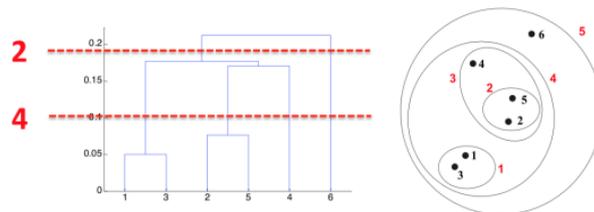


## 8- Apprentissage non-supervisé

### Algorithme hiérarchique par agglomération

#### Dendrogramme

- Représente graphiquement sous forme d'arbre binaire les différentes connections entre objets/clusters.
- La hauteur de la connexion (distance Cophenetic) entre deux objets/clusters représente la distance entre les deux objets/clusters connectés.



source : Philippe Esling

# Elements de probabilité

# 9- Elements de probabilité

## 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

### 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** :
  - $P(X = x_i) \geq 0$
  - $\sum_i P(X = x_i) = 1$
- Fonction de **répartition** (distribution des probabilités cumulées) :
  - $t \rightarrow F_X(t) = P(X < t)$

#### Exemples de loi de probabilité discrètes

- Loi **uniforme discrète**

- $P(X = x_1) = P(X = x_2) = \dots = P(X = x_n) = \frac{1}{n}$

- 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**

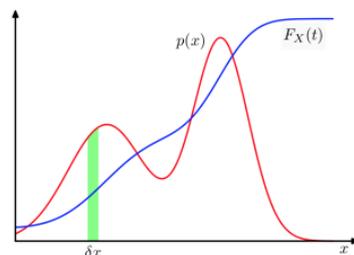
- nombre  $k$  de succès obtenus à l'issue de  $n$  épreuves de Bernoulli indépendantes de paramètre  $p \in [0, 1]$
  - $P(X = k) = \binom{n}{k} p^k (1 - p)^{n - k}$

- Loi **géométrique**

- loi qui modélise le temps d'attente du premier succès dans une série d'épreuves de Bernoulli indépendantes à probabilité de succès  $p \in [0, 1]$
  - $P(X = k) = (1 - p)^{k-1} p$

### Variable aléatoire continue

- $X$  peut prendre toutes les valeurs **continues** dans un intervalle donné (borné ou non borné)
  - Exemple : le nombre de Joules du cerveau
- **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 associée à une intervalle  $[a, b]$ 
    - $P(a \leq X \leq b) = \int_a^b f(x) dx$
- **Propriétés** :
  - $f(x) \geq 0 \quad \forall x \in \mathbb{R}$
  - $\int_{-\infty}^{\infty} f(x) dx = 1$
- Fonction de **répartition** :
  - $t \rightarrow F_X(t) = P(X < t) = \int_{-\infty}^t f(x) dx$
  - $P(a \leq X \leq b) = F_X(b) - F_X(a) = \int_a^b p(x) dx$



source : Hugo Larochelle

#### 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
  - $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

#### 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 :  $E(X) = \int_{x=-\infty}^{\infty} xf(x)dx$
- **Propriétés**
  - $E(X + Y) = E(X) + E(Y)$
  - $E(aX) = aE(X) \quad \forall a \in \mathbb{R}$

#### Variance $V(X)$

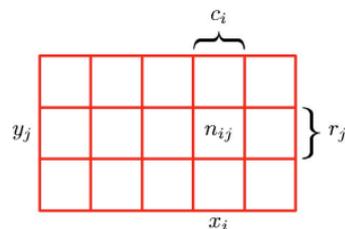
- La variance  $V(X)$  mesure la **dispersion** de  $X$  autour de son espérance
  - on appelle écart-type  $\sigma(A) = \sqrt{V(X)}$
- **Calcul**
  - espérance mathématique du carré de l'écart à l'espérance mathématique.
- Cas discret :  $V(X) = \sum_{n=1}^N \{x_n - E(X)\}^2 P(X = x_n)$
- Cas continu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

## 9- Elements de probabilité

### Cas de deux variables aléatoires

#### Cas de deux variables aléatoires

- Loi **jointe**, probabilité **jointe** :
  - Probabilité d'observer simultanément  $X = x_i$  et  $Y = y_j$
  - Discret :  $p_{xy} = P(X = x_i, Y = y_j)$
  - 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  $V(X + Y) = V(X) + V(Y)$
- Mesures** :
  - Variance  $V(X) = E(\{X - E(X)\}\{X - E(X)\})$
  - Covariance  $cov(X, Y) = E(\{X - E(X)\}\{Y - E(Y)\})$
  - Corrélation  $R(X, Y) = \frac{cov(X, Y)}{\sigma(X)\sigma(Y)}$
- Propriétés :
  - si  $X$  et  $Y$  sont deux variables aléatoires indépendantes alors  $cov(X, Y) = 0$



source : Arshia Cont

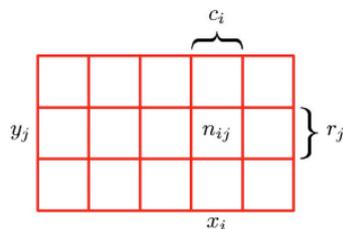
## 9- Elements de probabilité

### Cas de deux variables aléatoires

#### Cas de deux variables aléatoires

- 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

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

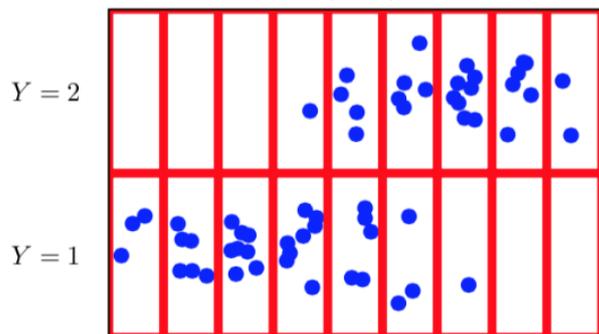


source : Arshia Cont

# 9- Elements de probabilité

## Cas de deux variables aléatoires

$p(X, Y)$

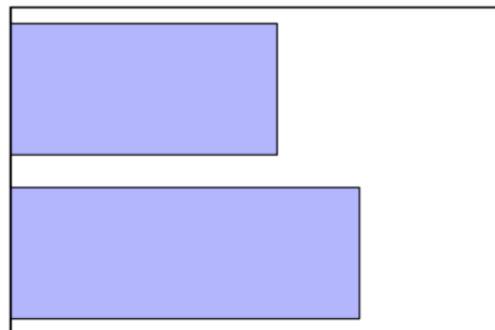


$Y = 2$

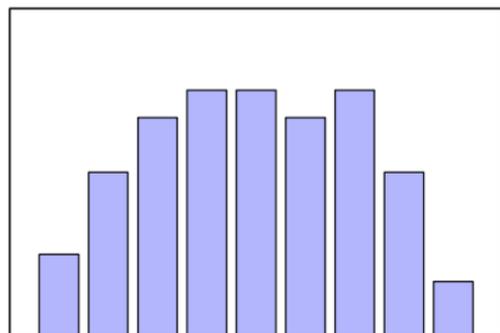
$Y = 1$

$X$

$p(Y)$

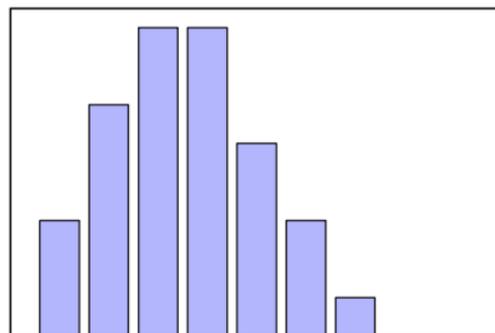


$p(X)$



$X$

$p(X|Y = 1)$



$X$

### 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**

## 9- Elements de probabilité

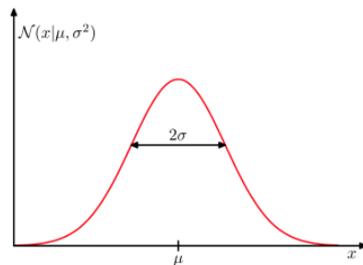
### La loi normale ou loi gaussienne

#### La loi normale ou loi gaussienne

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

#### Formulation à $D = 1$ dimension

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



source : Hugo Larochelle

## 9- Elements de probabilité

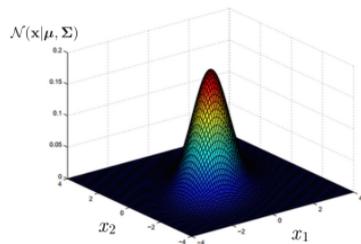
### La loi normale ou loi gaussienne

#### La loi normale ou loi gaussienne

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

#### Formulation à $D > 1$ dimension

- $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}$
- $E(\mathbf{x}) = \boldsymbol{\mu}$
- $cov(\mathbf{x}) = \boldsymbol{\Sigma}$
- Remarque :
  - $\boldsymbol{\Sigma}$  est une matrice!!!
  - permet de représenter les dépendances (corrélation) entre les dimensions  $d$  de  $X$



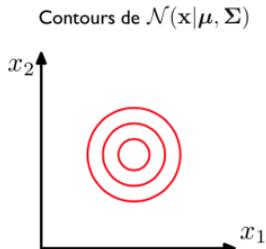
source : Hugo Larochelle

## 9- Elements de probabilité

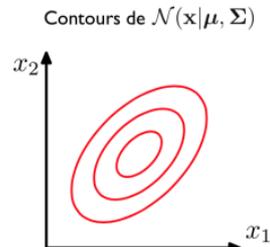
### La loi normale ou loi gaussienne

#### Exemples de dépendances représentées par $\Sigma$

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



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



source : Hugo Larochelle

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

