

Séminaire Classification

Jean-Yves Tourneret[†]

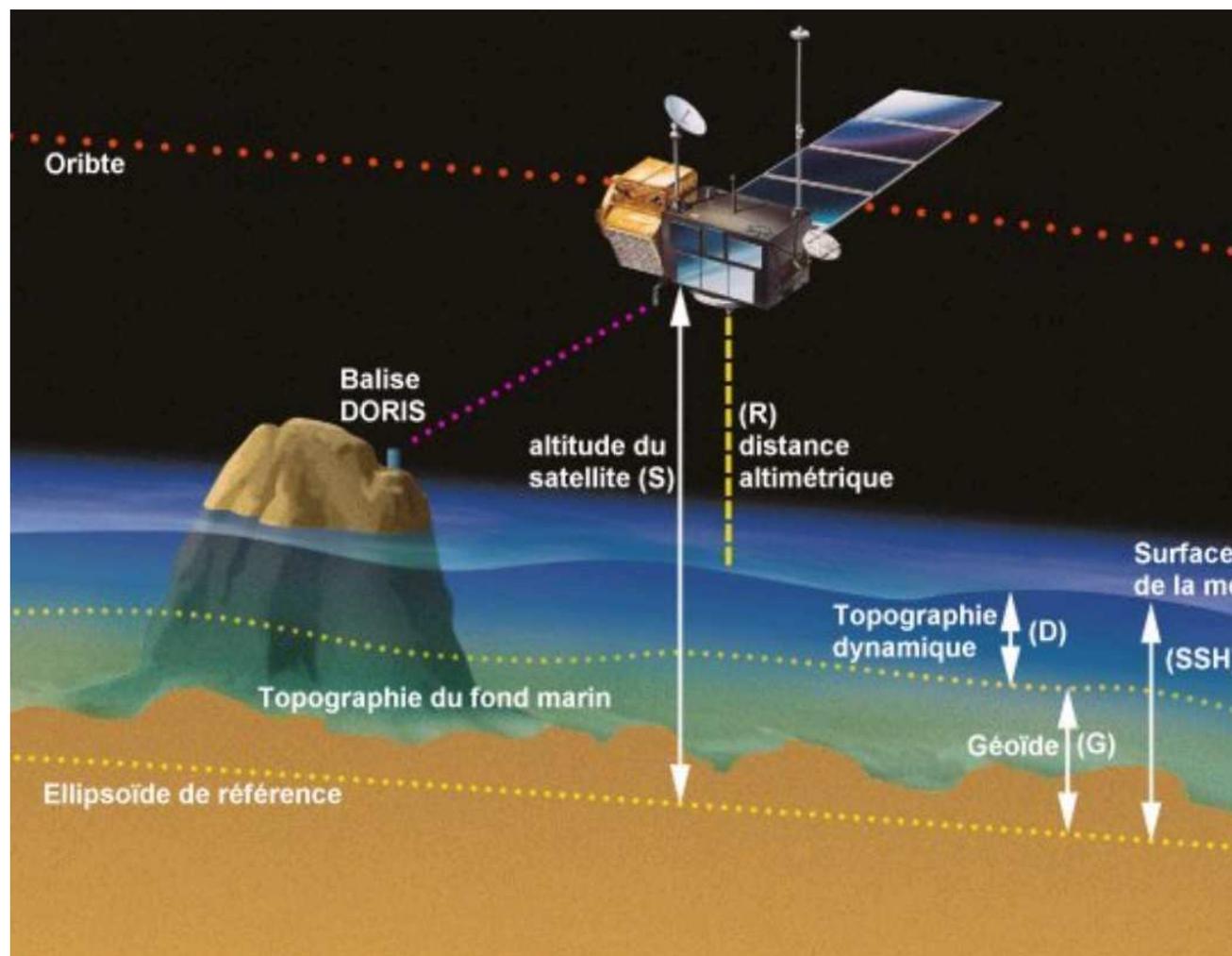
IRIT - ENSEEIHT - TéSA

Thème 1 : Analyse et synthèse de l'information

Plan de l'exposé

1. Introduction
2. Prétraitement
3. Méthodes Statistiques
4. Fonctions discriminantes
5. Classification non supervisée

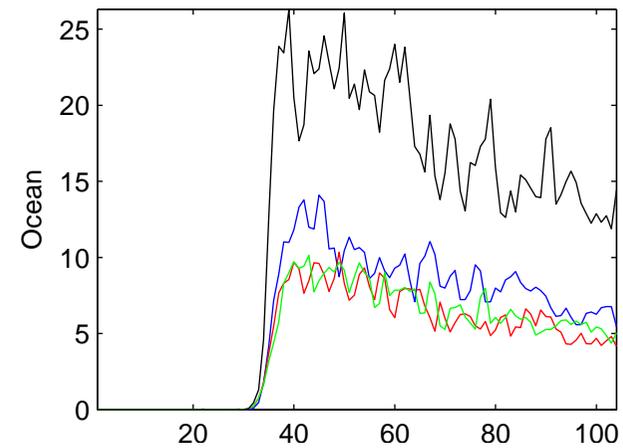
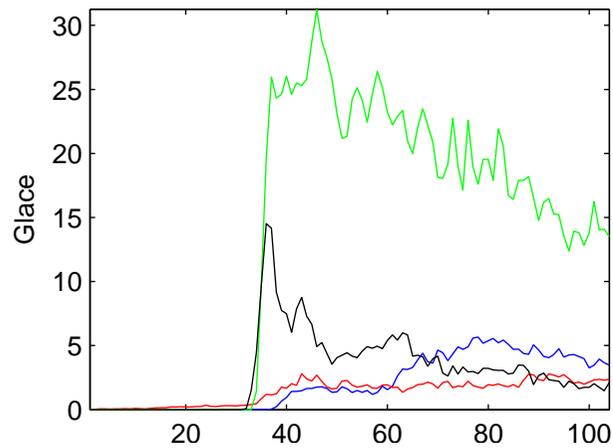
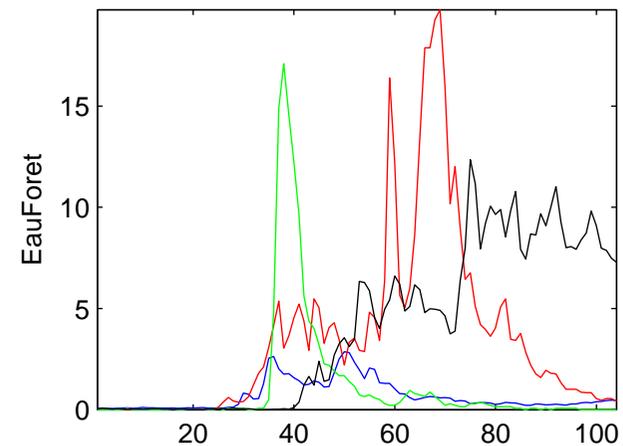
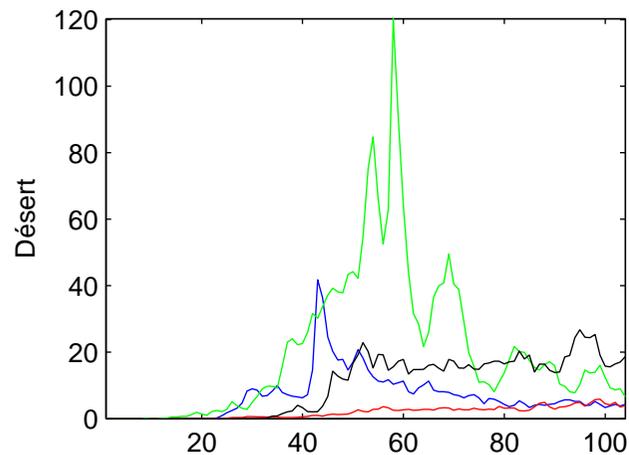
Principe de l'altimétrie



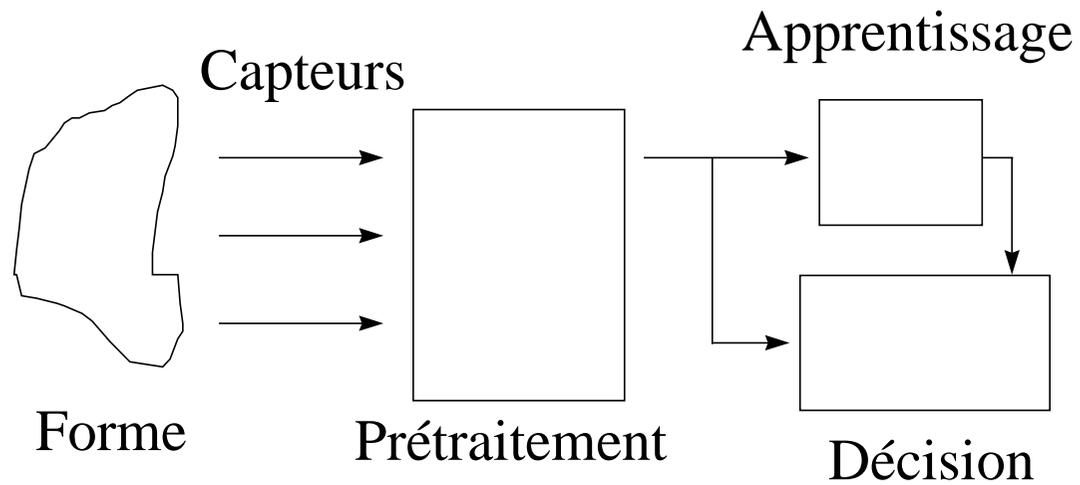
Exemple

- Classe ω_1 : océans
- Classe ω_2 : déserts (Algérie, Lybie, Afrique du Sud)
- Classe ω_3 : eaux et forêts (Amazonie, Canada, Congo, Russie)
- Classe ω_4 : glaces (glace continentale arctique, glace continentale Groenland, glace mer antarctique, glace mer arctique)

Formes d'onde altimétriques



Chaîne de Reconnaissance



Plan de l'exposé

1. Introduction
2. **Prétraitement**
3. Méthodes Statistiques
4. Fonctions discriminantes
5. Classification non supervisée

Prétraitement

- **Extraction de données**

Modélisation et Représentation

Analyse en Composantes Principales (ACP linéaire ou non-linéaire)

Analyse Factorielle Discriminante (AFD linéaire ou non-linéaire)

- **Sélection de données**

Recherche exhaustive

Utilisation de l'algorithme "Branch and Bound"

Méthodes sous-optimales

Analyse en Composantes Principales

Hypothèse : N individus de \mathbb{R}^p (p variables) notés $X_i = (X_i(1), \dots, X_i(p))^T$,
 $i = 1, \dots, N$

Problème : recherche de \tilde{X}_i qui

$$\text{minimise } \frac{1}{n} \sum_{i=1}^n \|X_i - \tilde{X}_i\|^2 \quad \text{ou} \quad \text{maximise } \frac{1}{n} \sum_{i=1}^n \|\tilde{X}_i\|^2 \quad (1)$$

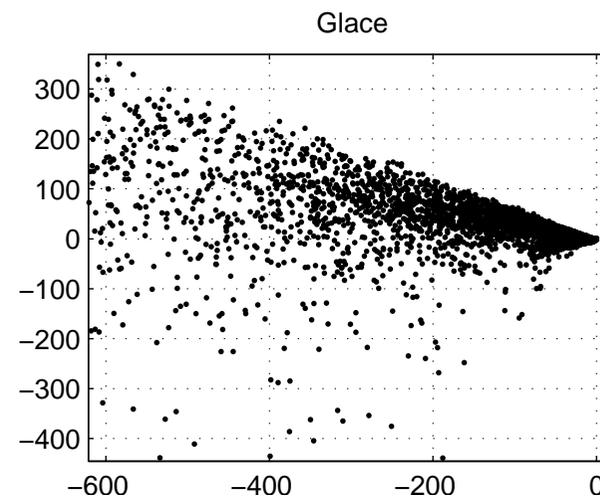
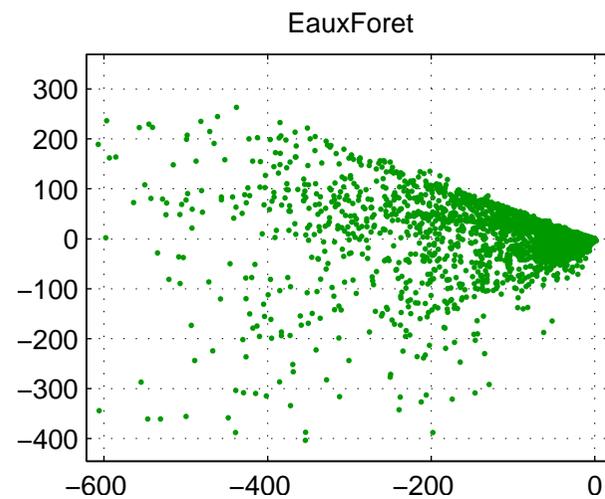
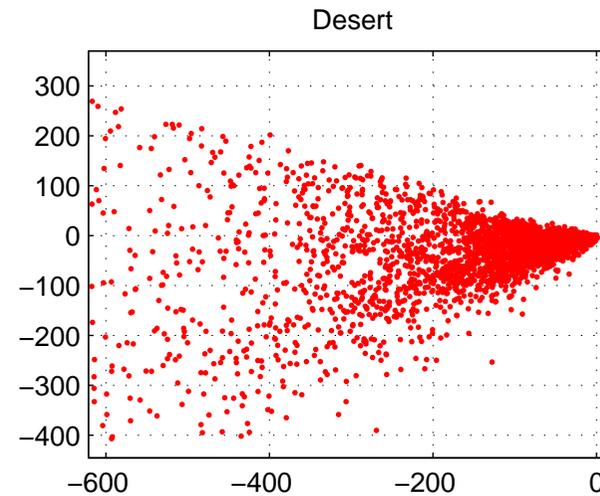
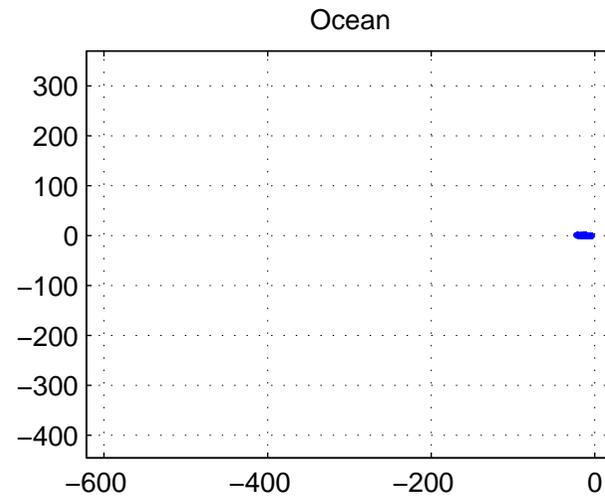
Théorème : les solutions sont “emboîtées”.

\Rightarrow Détermination des axes principaux de proche en proche (par exemple le premier axe principal s’obtient à l’aide du Lagrangien $L(u) = u^T C u - \lambda u^T u$ avec $C = \frac{1}{n} \sum_{i=1}^n X_i X_i^T$ matrice de covariance des données)

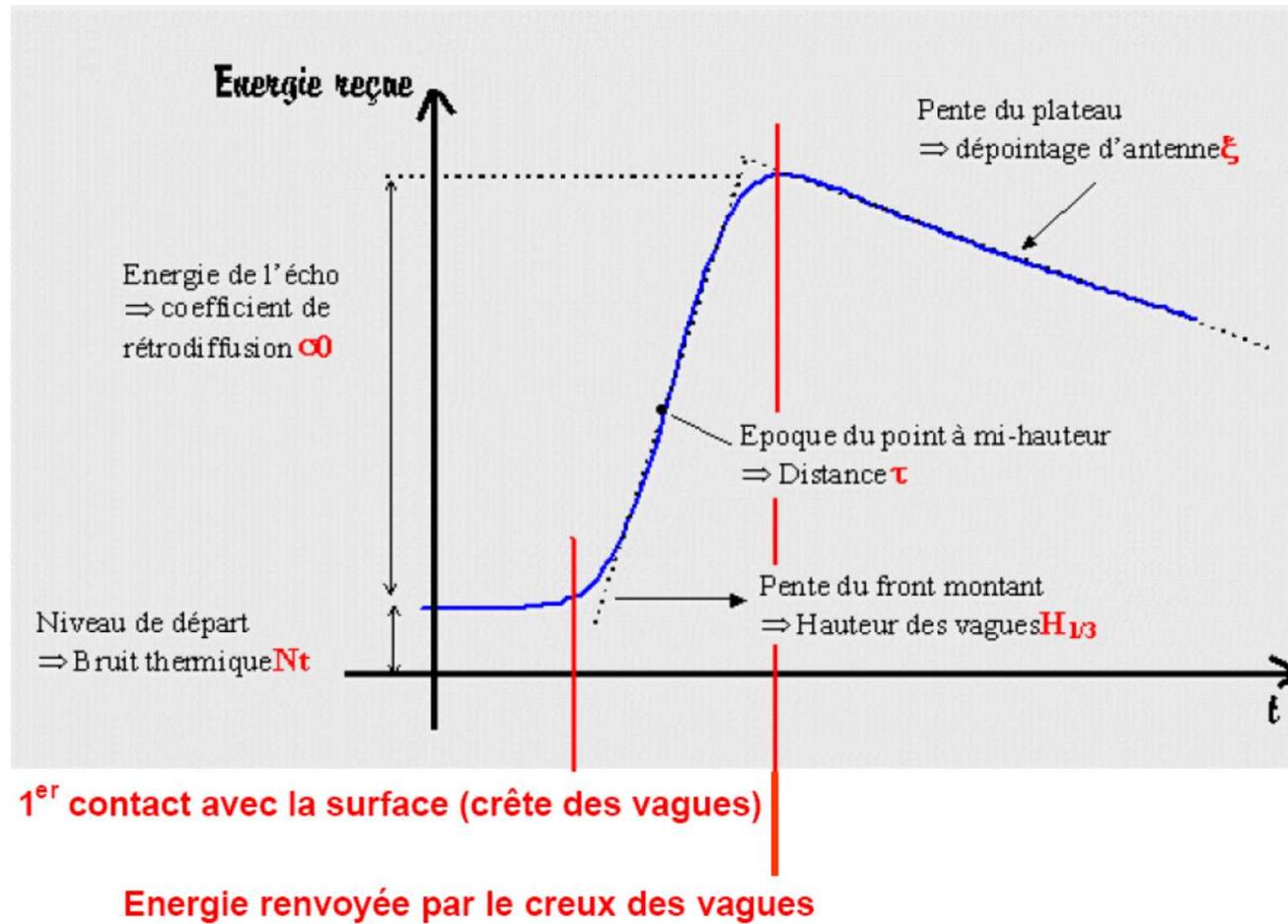
Les axes principaux sont les vecteurs propres de la matrice C

Remarque : **ACP des variables**

ACP des individus (Formes d'Onde)



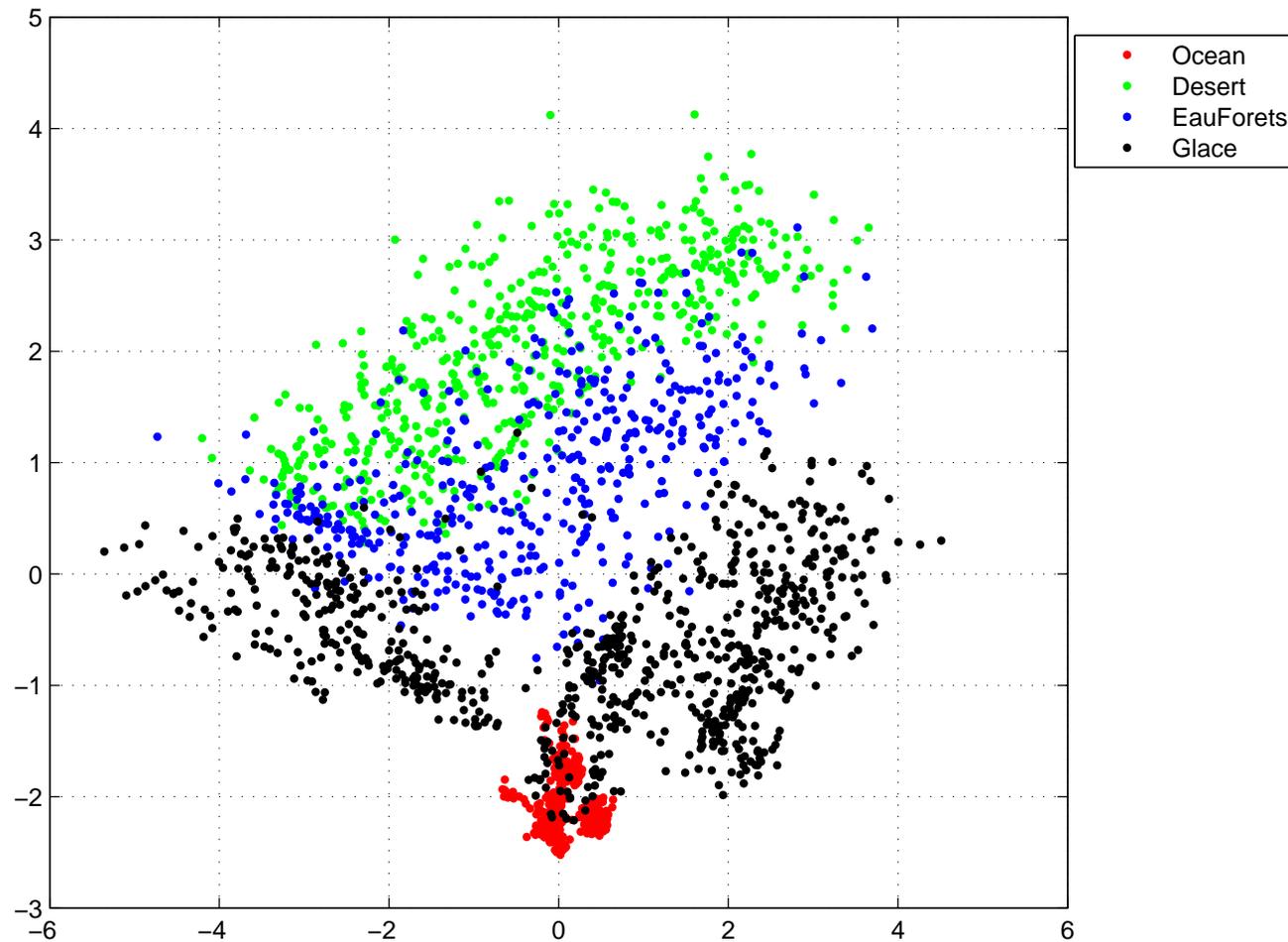
Modèle Analytique



Extraction de paramètres pertinents

- Paramètres estimés à partir des formes d'onde (Retracking) :
 - `ku_swh_1Hz` : Significant Wave Height of ku band
 - `ku_std_swh_1Hz` : Standard deviation on swh 18 Hz
 - `ku_sigma0_1Hz` : Backscatter coefficient ku band
 - `s_sigma0_1Hz` : Backscatter coefficient s band
 - `elevation_1Hz` : Élévation au dessus de l'ellipsoïde de référence
 - `ku_peakiness_1Hz` : Peakiness des formes d'ondes ku band
 - `ku_ice2_LEwidth_1Hz` : Largeur du front de montée
- Paramètres mesurés par radiométrie :
 - `tb23_1Hz` : Température de brillance 23 GHz
 - `tb36_1Hz` : Température de brillance 36 GHz

ACP des individus (retracking)



Analyse factorielle discriminante

Critère de Fisher (2 classes)

$$J(u) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{u^T B u}{u^T S u} \quad (2)$$

où B et S sont les matrices de dispersion *interclasse* et *intraclasse*

Généralisation à $K > 2$ classes

$$S = \sum_{i=1}^K S_i = \sum_{i=1}^K \sum_{x \in W_i} (x - m_i)(x - m_i)^T \quad B = \sum_{i=1}^K n_i (m_i - m)(m_i - m)^T \quad (3)$$

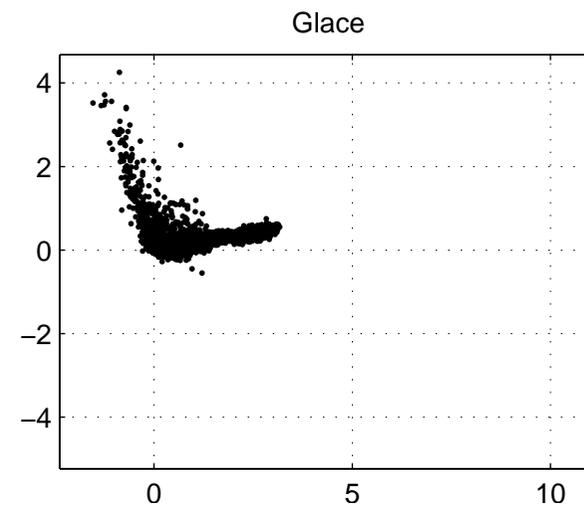
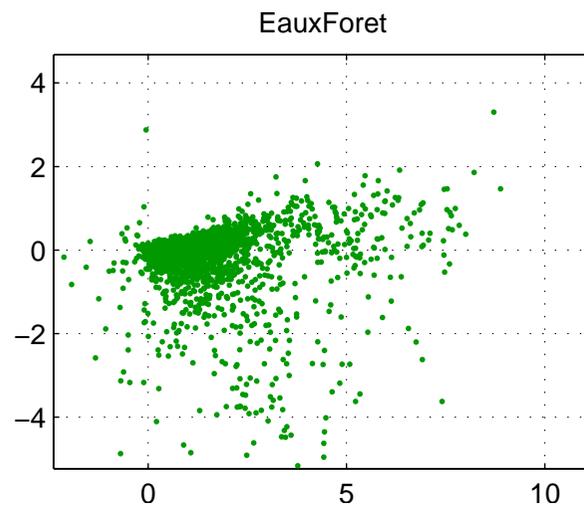
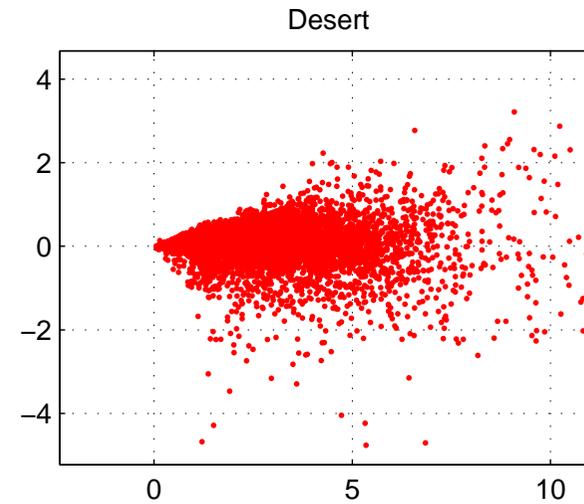
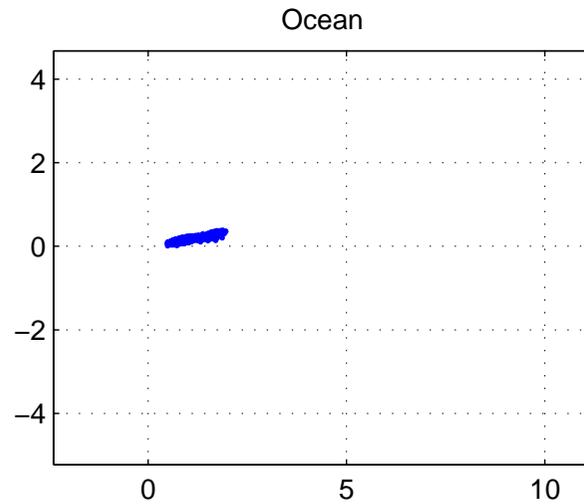
Détermination des axes factoriels discriminants de proche en proche (par exemple, le Lagrangien pour le premier axe est $L(u) = u^T B u - \lambda(u^T S u)$):

Les axes factoriels sont les vecteurs propres de $S^{-1}B$ ou de $T^{-1}B$

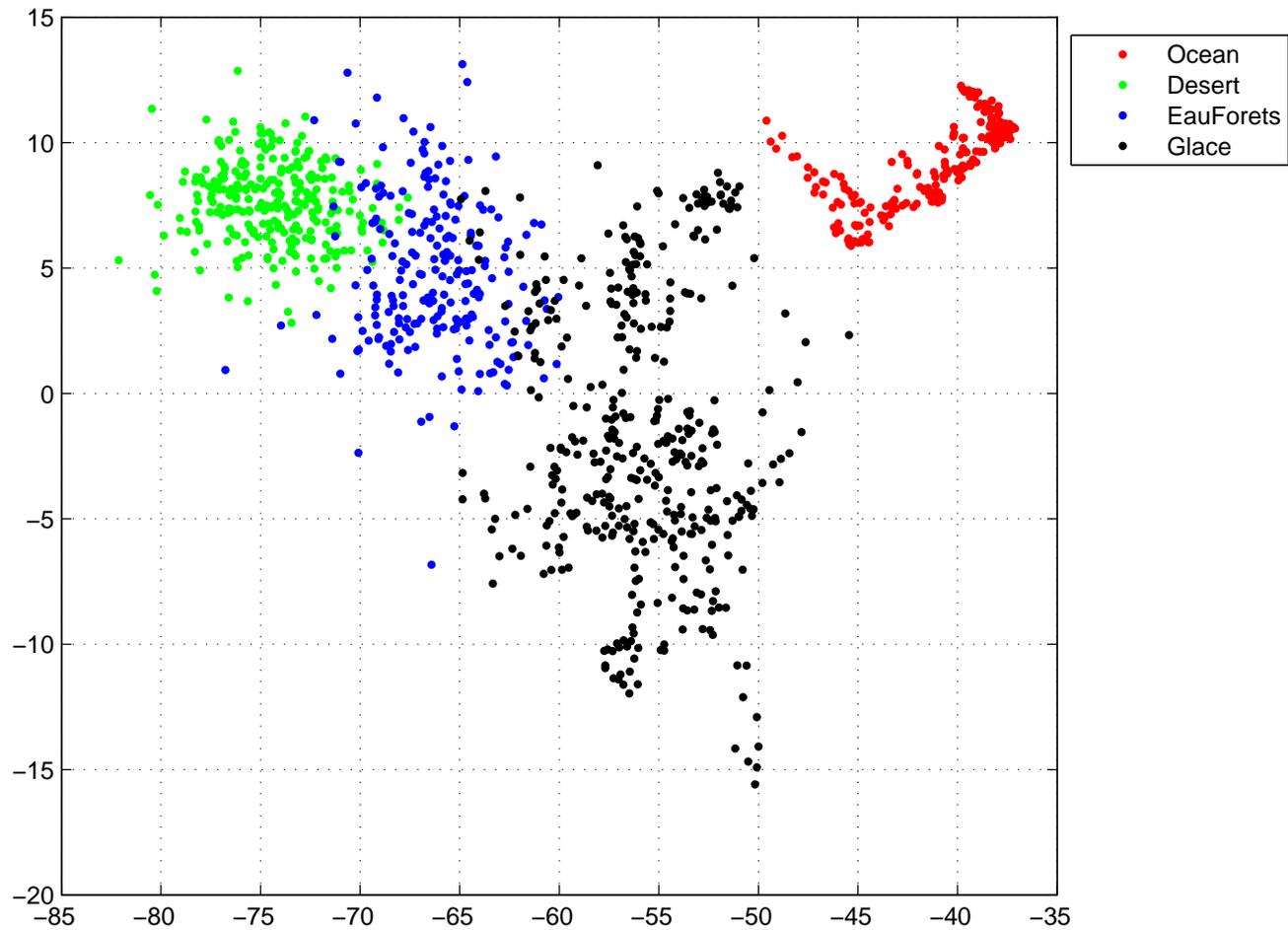
Remarques :

- ACP et AFD non-linéaires
- Poursuite de projection

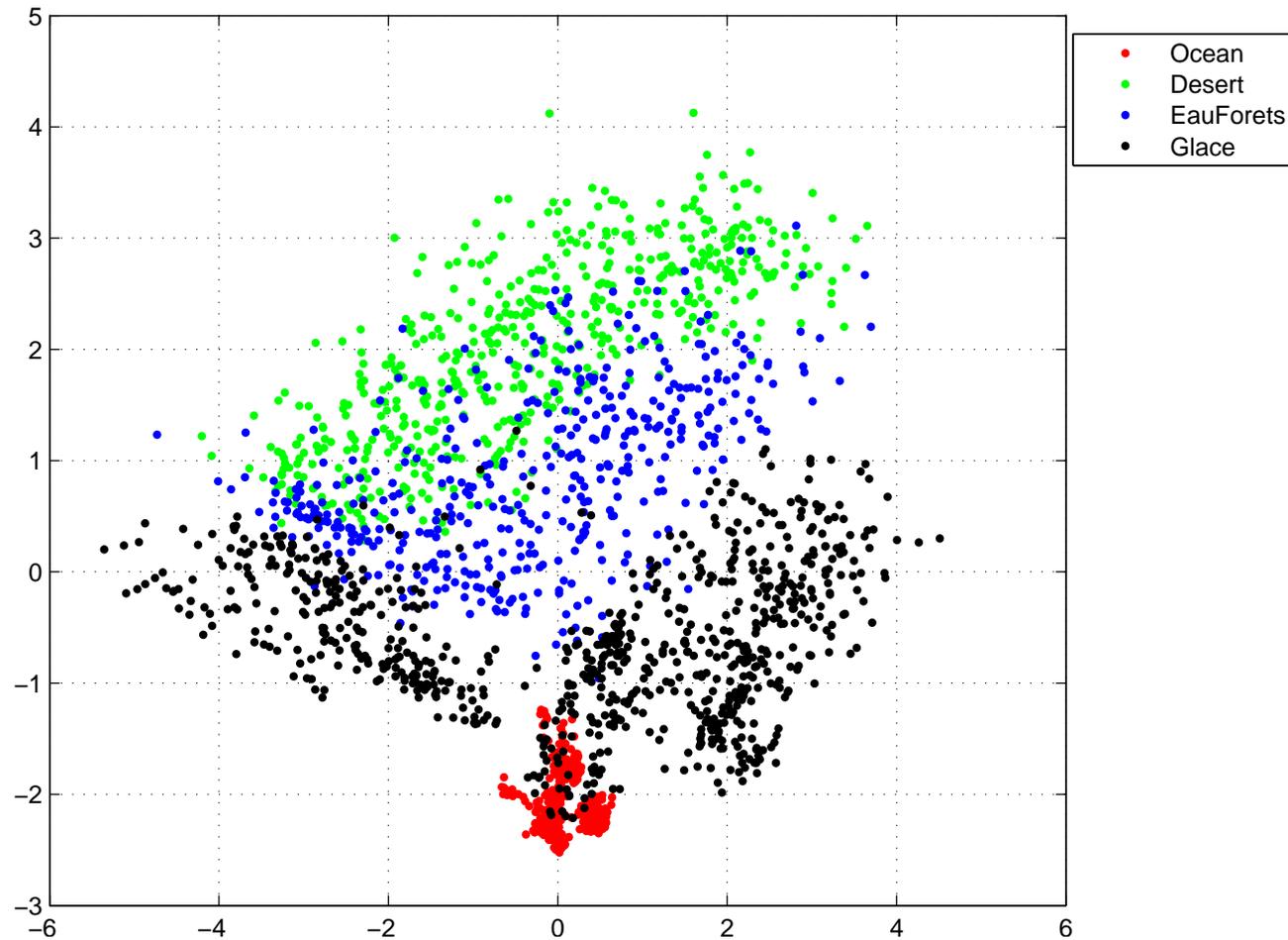
AFD (Formes d'Onde)



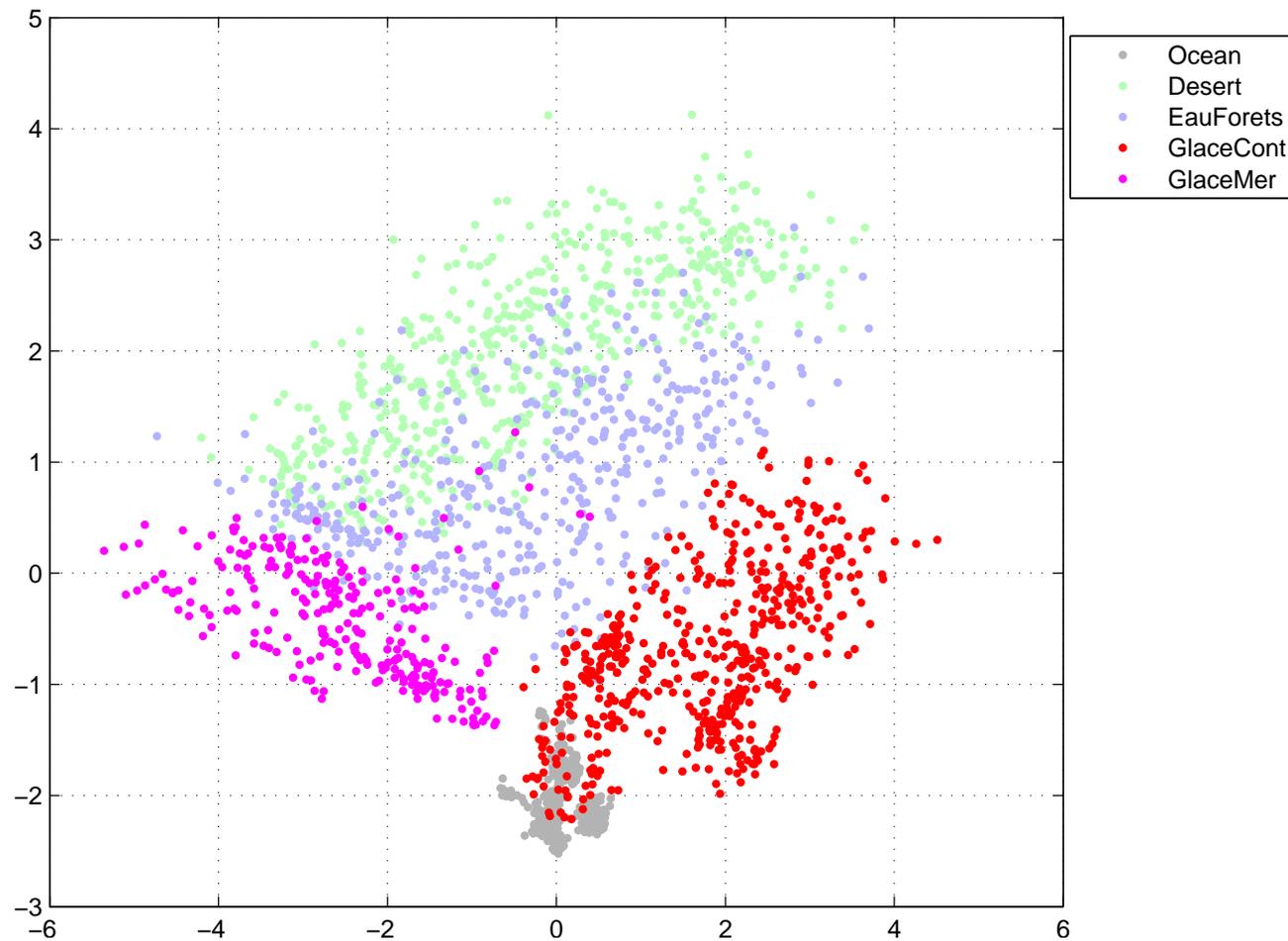
AFD (retracking)



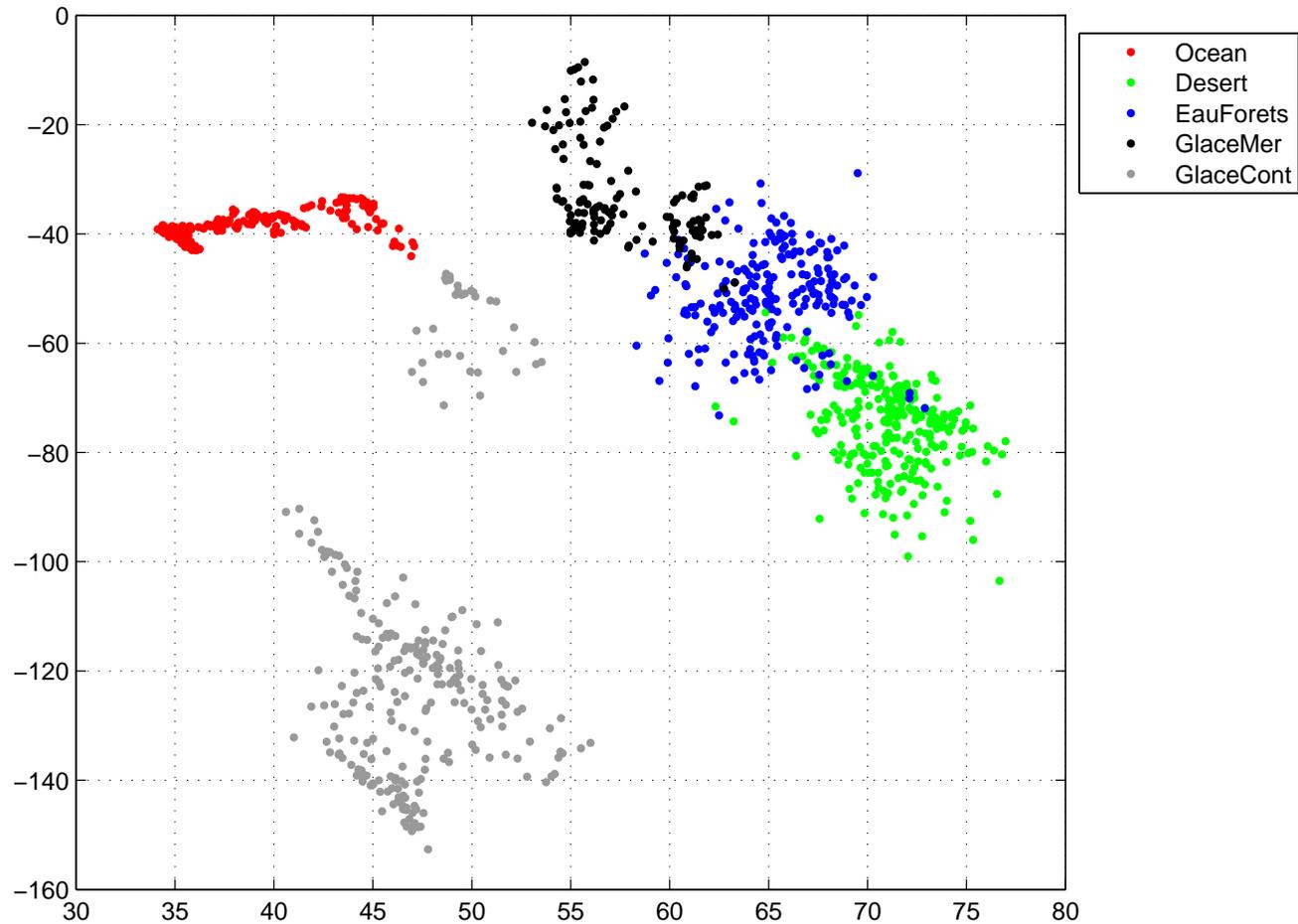
Un autre Problème de Classification



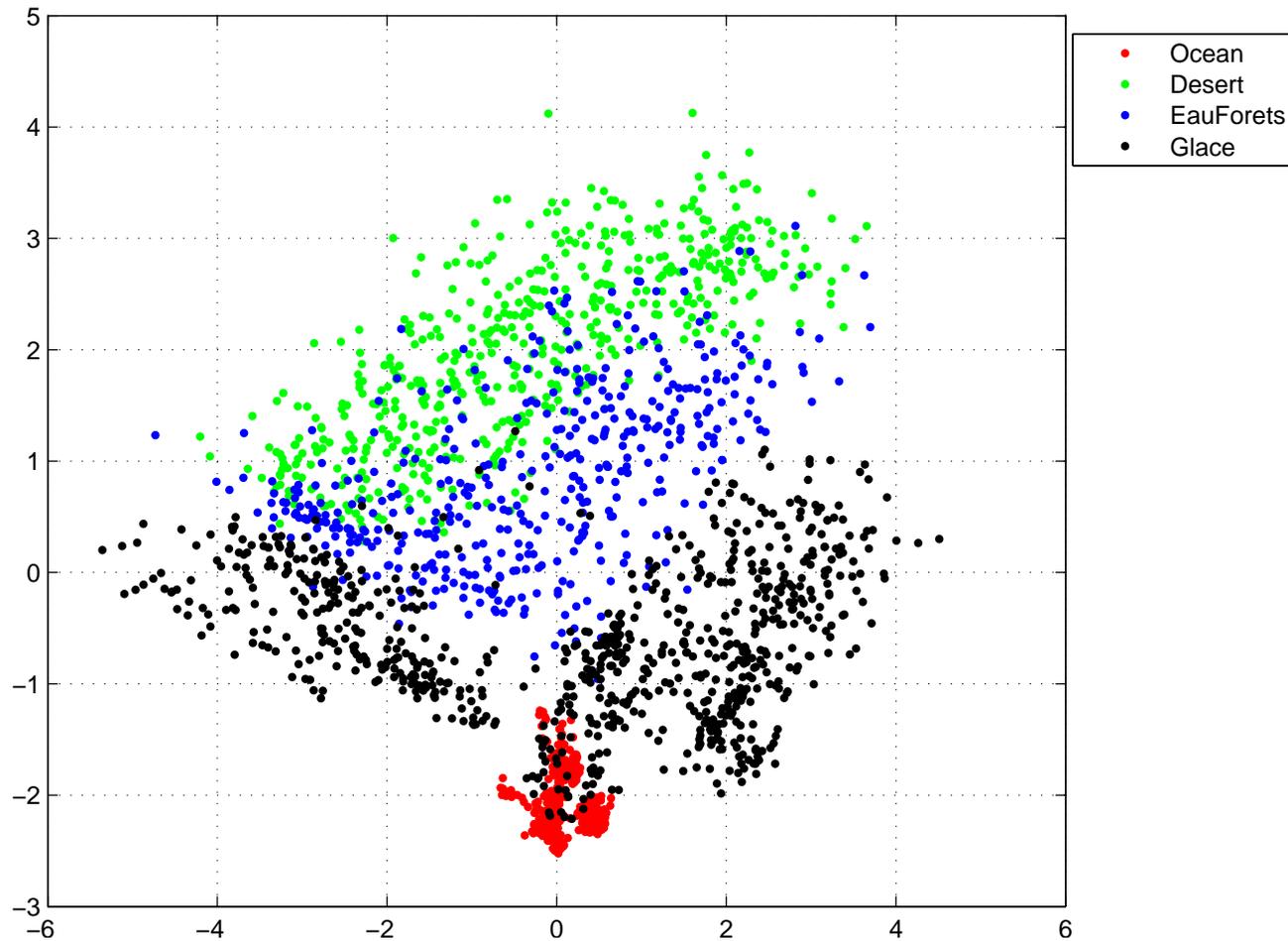
Classification des Glaces



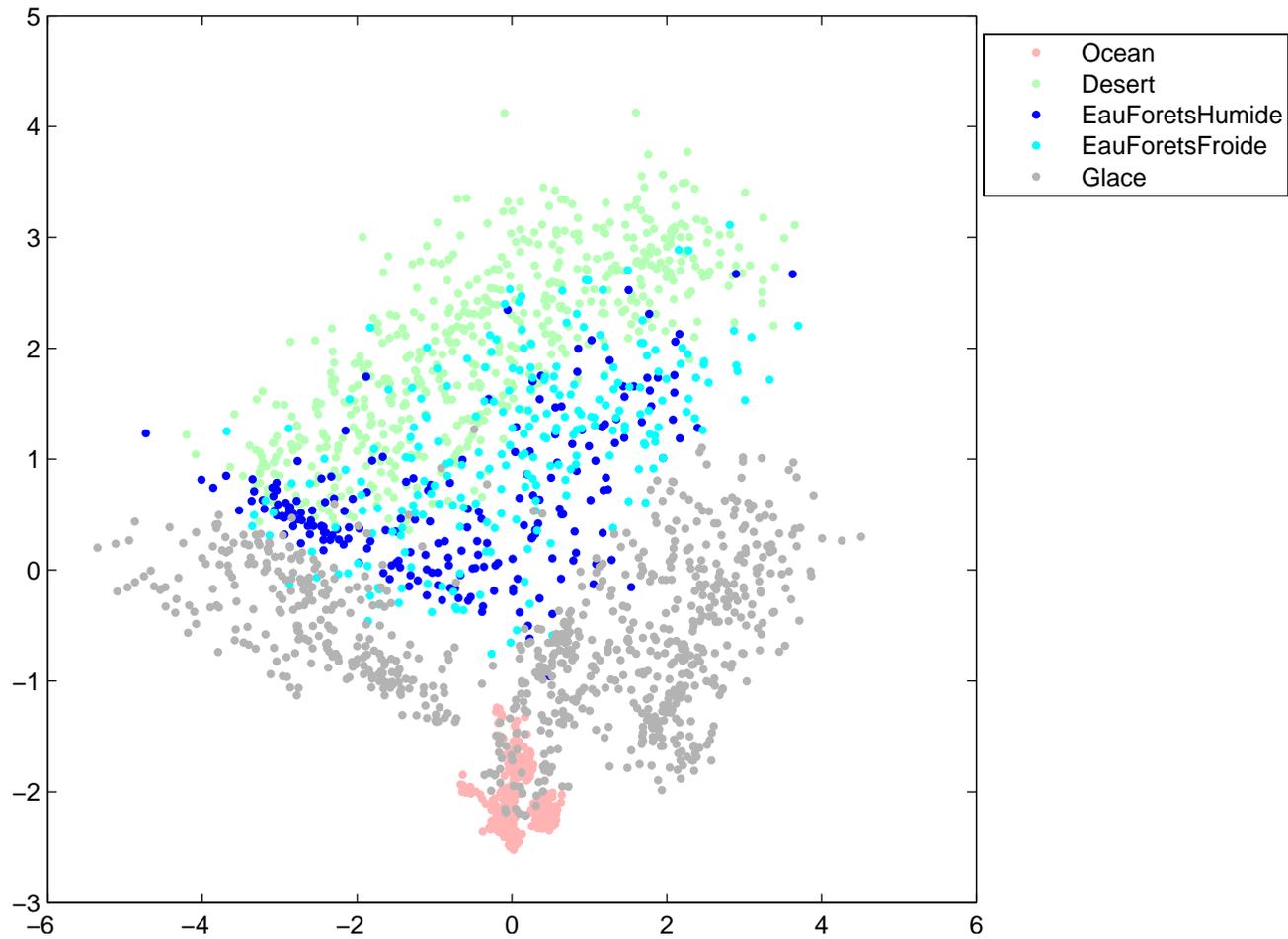
AFD avec cinq Classes



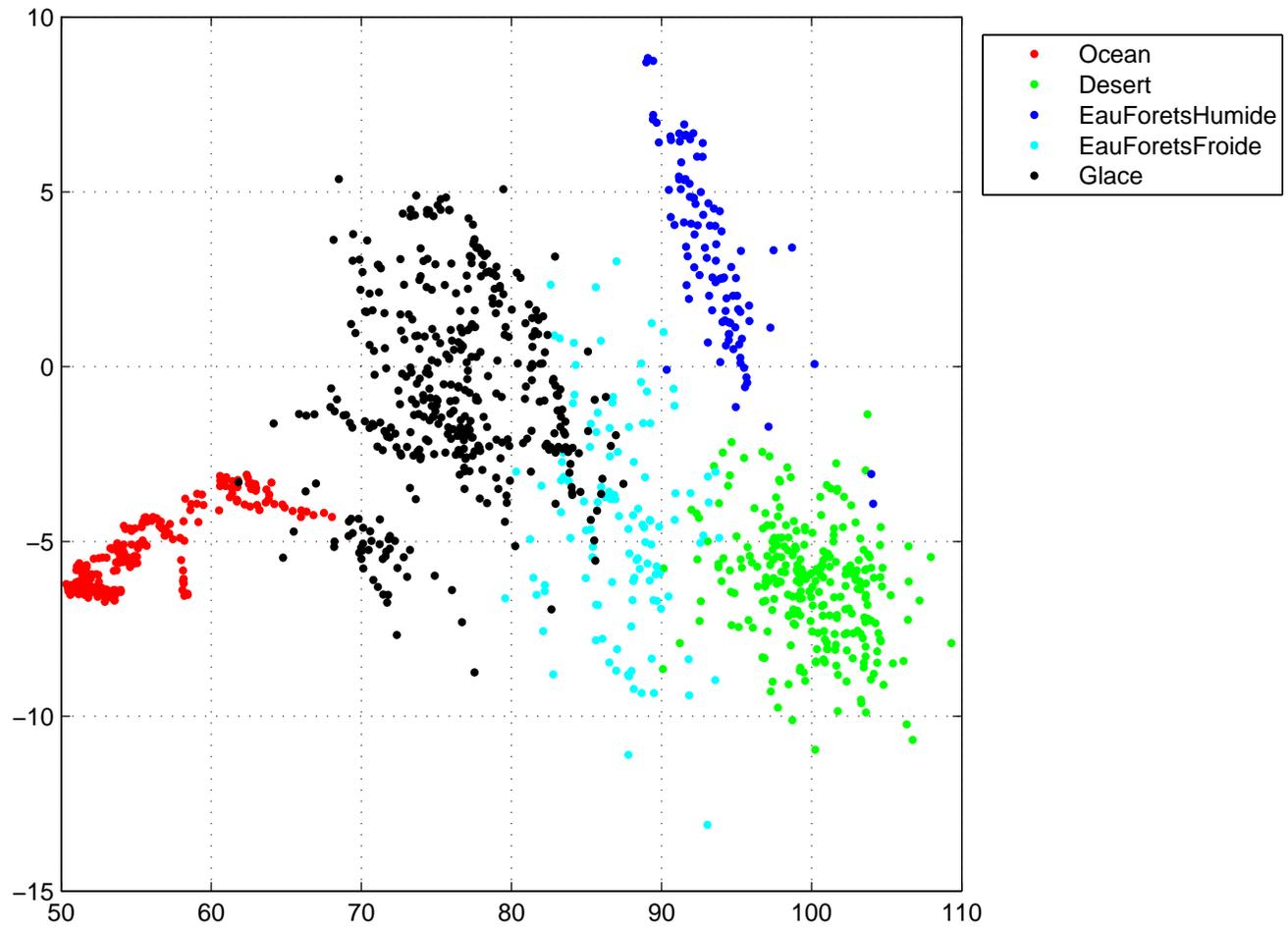
Un autre Problème de Classification



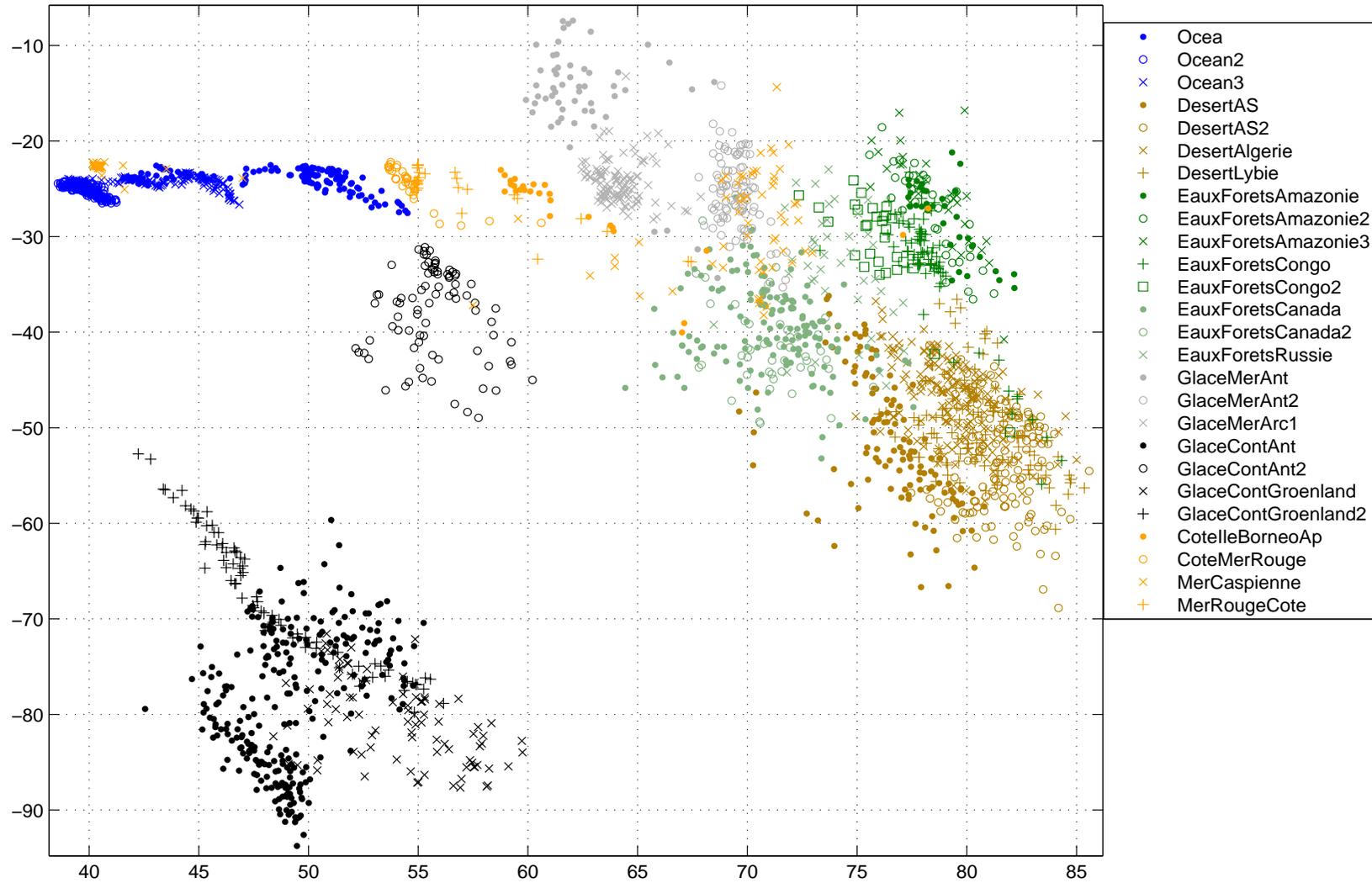
Classification des Eaux et Forêts



AFD avec cinq Classes



Problème à 7 classes

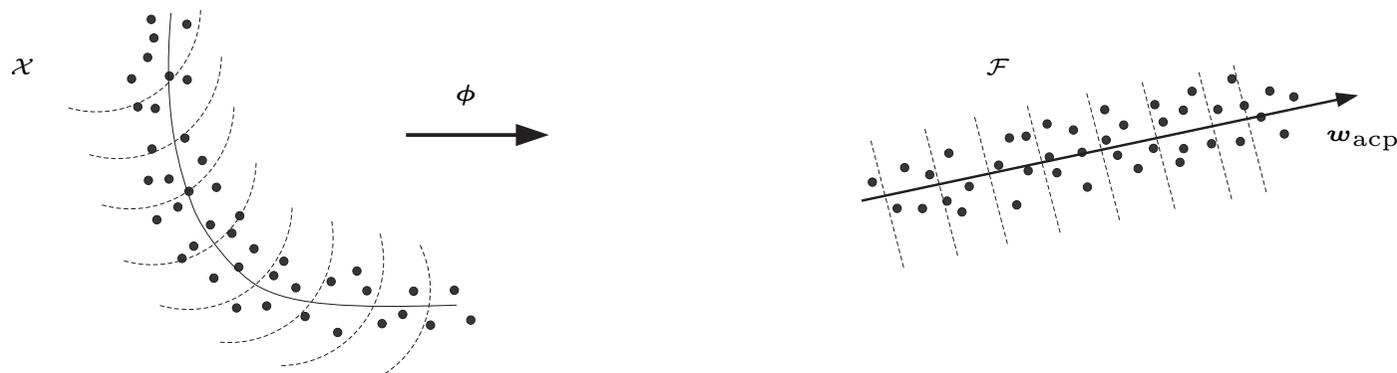


Extension au cas non-linéaire

Plongement dans un nouvel espace de représentation à l'aide d'une application non-linéaire

$$\begin{aligned}\phi : \mathcal{X} &\longrightarrow \mathcal{F} \\ \mathbf{x} &\longmapsto \phi(\mathbf{x})\end{aligned}$$

On applique l'une des méthodes précédentes aux données transformées $\phi(\mathbf{x}_k)$.



$$\mathbf{x} = [x(1), x(2)]^T$$

$$\phi(\mathbf{x}) = [1, \sqrt{2}x(1), \sqrt{2}x(2), \sqrt{2}x(1)x(2), x(1)^2, x(2)^2]^T$$

L'astuce du noyau

Partant d'un constat d'échec...

Choix de $\phi(\mathbf{x})$ délicat ou calcul de $\lambda(\mathbf{x}) = \mathbf{u}^T \phi(\mathbf{x})$ (ou de tout autre produit scalaire dans \mathcal{F}) coûteux ...

Exemple : non-linéarité polynômiale de degré 5 appliquée à des images (16×16)
 $\Rightarrow \dim \mathcal{F} \approx 10^{10}$!!!

... le remède

Calculer un produit scalaire dans \mathcal{F} ne nécessite pas d'exhiber $\phi(\mathbf{x})$.

Exemple : cas quadratique

$$\begin{aligned} \phi(\mathbf{x})^T \phi(\mathbf{y}) &= \left[1, \sqrt{2} x(1), \sqrt{2} x(2), \sqrt{2} x(1)x(2), x(1)^2, x(2)^2 \right] \\ &\quad \left[1, \sqrt{2} y(1), \sqrt{2} y(2), \sqrt{2} y(1)y(2), y(1)^2, y(2)^2 \right]^T \\ &= (1 + \mathbf{x}^T \mathbf{y})^2 = \kappa(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Noyaux de Mercer

Condition de Mercer

Toute fonction symétrique $\kappa(\mathbf{x}_1, \mathbf{x}_2)$ de $\mathcal{L}_2(\mathcal{X}^2)$ vérifiant la condition ci-dessous définit un produit scalaire dans un espace fonctionnel hilbertien réel \mathcal{F} :

$$\iint \kappa(\mathbf{x}_1, \mathbf{x}_2) \varphi(\mathbf{x}_1) \varphi(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2 \geq 0, \quad \forall \varphi \in \mathcal{L}_2(\mathcal{X}).$$

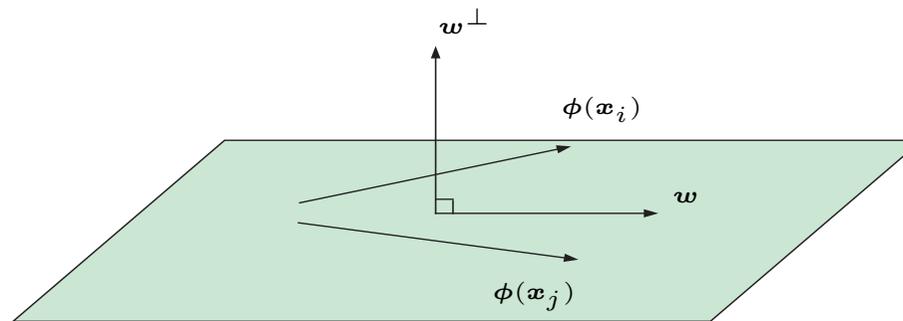
Exemples de noyaux

- noyau **polynômial** : $\kappa(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^T \mathbf{x}_2)^q$
- noyau **sigmoïdal** : $\frac{1}{\eta_0} \tanh(\beta_0 \mathbf{x}_1^T \mathbf{x}_2 - \alpha_0)$
- noyau **gaussien** : $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2 / \beta_0)$
- noyau **exponentiel** : $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\| / \beta_0)$
- ...

Representer theorem

La fonction recherchée étant définie par $\lambda(\mathbf{x}) = \mathbf{u}^T \phi(\mathbf{x})$, le domaine de recherche de \mathbf{u} peut être limité à l'espace linéaire induit par $\phi(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$. Selon le **Representer Theorem**, on a :

$$\mathbf{u} = \sum_{i=1}^n \alpha(i) \phi(\mathbf{x}_i) = \mathbf{Q} \boldsymbol{\alpha}$$



$$\lambda(\mathbf{x}) = \mathbf{u}^T \phi(\mathbf{x}) = \sum_{i=1}^n \alpha(i) \phi(\mathbf{x})^T \phi(\mathbf{x}_i) = \sum_{i=1}^n \alpha(i) \kappa(\mathbf{x}, \mathbf{x}_i)$$

Kernel ACP

Principe

On cherche à maximiser le critère défini par $J(\mathbf{u}) = \mathbf{u}^T \Sigma^\phi \mathbf{u}$, avec

$$\Sigma^\phi = \frac{1}{n} \sum_{k=1}^n (\phi(\mathbf{x}_k) - \mathbf{m}^\phi)(\phi(\mathbf{x}_k) - \mathbf{m}^\phi)^T,$$

soit à résoudre le problème aux valeurs propres : $\Sigma^\phi \mathbf{u} = \gamma \mathbf{u}$.

Stratégie

On utilise $\mathbf{u} = \sum_{i=1}^n \alpha(i) \phi(\mathbf{x}_i) = \mathbf{Q} \boldsymbol{\alpha}$, puis on multiplie à gauche par \mathbf{Q}^T :

$$\mathbf{Q}^T \Sigma^\phi \mathbf{Q} \boldsymbol{\alpha} = \gamma \mathbf{Q}^T \mathbf{Q} \boldsymbol{\alpha}$$

Dans le cas de [données centrées](#) dans \mathcal{F} , on aboutit au problème suivant

$$\mathbf{K}^2 \boldsymbol{\alpha} = n \mathbf{K} \boldsymbol{\alpha}$$

soit à diagonaliser la matrice de Gram $(\mathbf{K})_{ij} = \kappa(i, j)$.

Kernel AFD

Principe

On cherche à maximiser le critère de Fisher défini par $J(\mathbf{u}) = \frac{\mathbf{u}^T \Sigma_B^\phi \mathbf{u}}{\mathbf{u}^T \Sigma_S^\phi \mathbf{u}}$, avec

$$\begin{aligned}\Sigma_B^\phi &= (\mathbf{m}_1^\phi - \mathbf{m}_0^\phi)(\mathbf{m}_1^\phi - \mathbf{m}_0^\phi)^T \\ \Sigma_S^\phi &= \frac{1}{n} \sum_{i=0,1} \sum_{\mathbf{x} \in \omega_i} (\phi(\mathbf{x}) - \mathbf{m}_i^\phi)(\phi(\mathbf{x}) - \mathbf{m}_i^\phi)^T\end{aligned}$$

Numérateur

$$\begin{aligned}\mathbf{u}^T \mathbf{m}_i &= \frac{1}{n_i} \sum_{k=1}^n \sum_{\mathbf{x} \in \omega_i} \alpha(k) \phi(\mathbf{x})^T \phi(\mathbf{x}_k) = \frac{1}{n_i} \sum_{k=1}^n \sum_{\mathbf{x} \in \omega_i} \alpha(k) \kappa(\mathbf{x}, \mathbf{x}_k) \\ &= \boldsymbol{\alpha}^T \boldsymbol{\mu}_i\end{aligned}$$

avec $\mu_i(k) = \frac{1}{n_i} \sum_{\mathbf{x} \in \omega_i} \kappa(\mathbf{x}, \mathbf{x}_k)$, $k \in \{1, \dots, n\}$. Il en résulte que :

$$\mathbf{u}^t \Sigma_B^\phi \mathbf{u} = \boldsymbol{\alpha}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \mathbf{S}_B \boldsymbol{\alpha}$$

Kernel AFD

Dénominateur

$$\begin{aligned}
 \mathbf{u}^T \Sigma_S^\phi \mathbf{u} &= \frac{1}{n} \sum_{i=0,1} \sum_{\mathbf{x} \in \omega_i} \mathbf{u}^T (\phi(\mathbf{x}) - \mathbf{m}_i^\phi) (\phi(\mathbf{x}) - \mathbf{m}_i^\phi)^T \mathbf{u} \\
 &= \frac{1}{n} \sum_{i=0,1} \sum_{\mathbf{x} \in \omega_i} \boldsymbol{\alpha}^T (\boldsymbol{\kappa}(\mathbf{x}) - \boldsymbol{\mu}_i) (\boldsymbol{\kappa}(\mathbf{x}) - \boldsymbol{\mu}_i)^T \boldsymbol{\alpha}
 \end{aligned}$$

où $\boldsymbol{\kappa}(\mathbf{x})$ désigne la colonne de la matrice de Gram $(\mathbf{K})_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ correspondant à \mathbf{x} . On réécrit finalement :

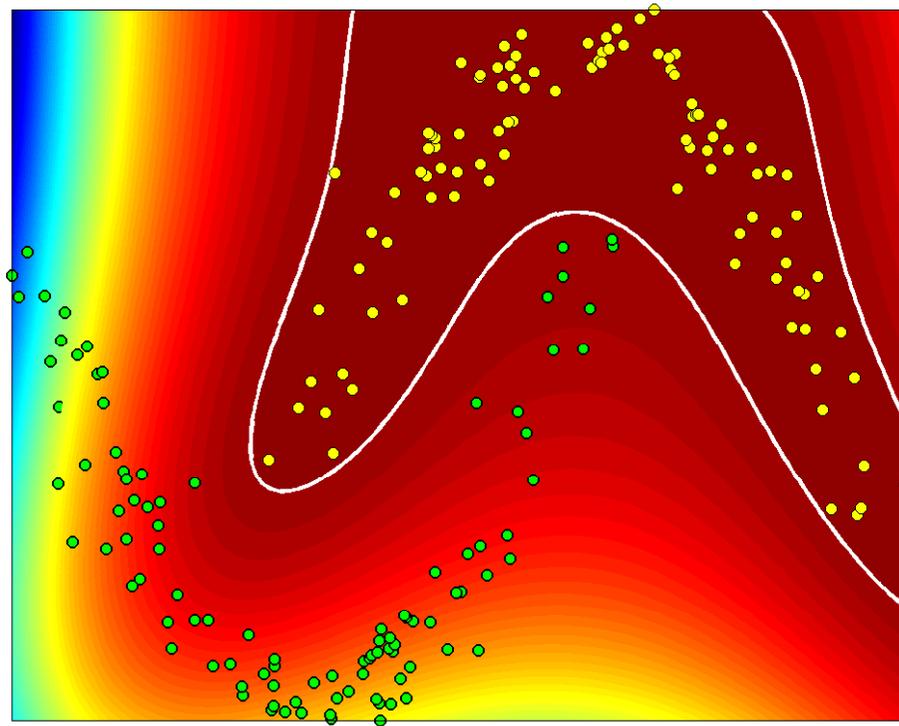
$$\mathbf{u}^T \Sigma_S^\phi \mathbf{u} = \boldsymbol{\alpha}^T \mathbf{S}_S \boldsymbol{\alpha}$$

Critère de Fisher

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{S}_B \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{S}_S \boldsymbol{\alpha}}$$

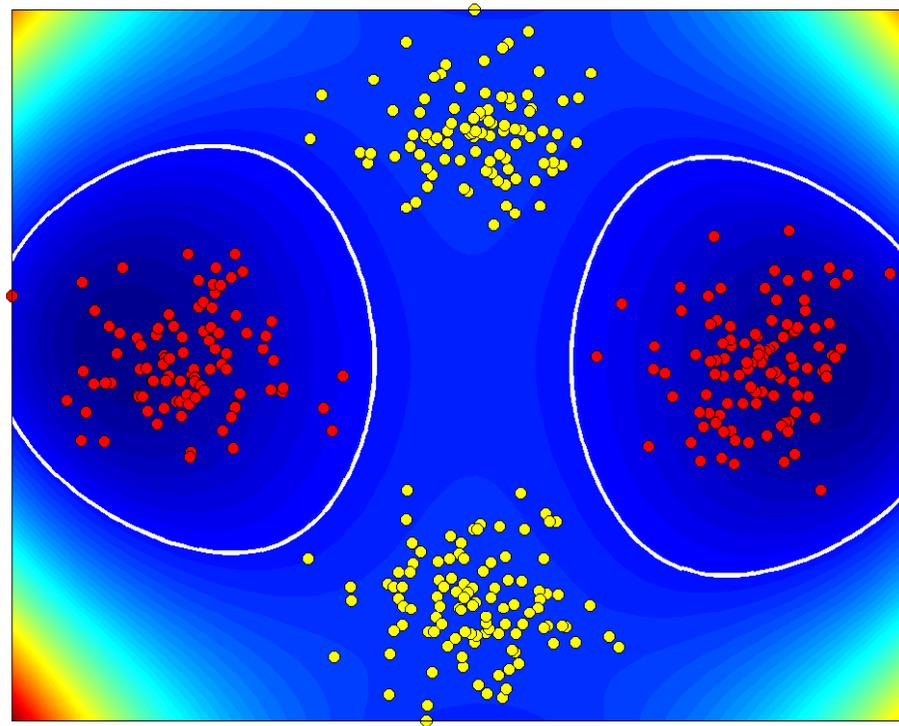
Détecteur polynomial de degré 3

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^T \mathbf{x}_2)^3$$



Détecteur à noyau gaussien

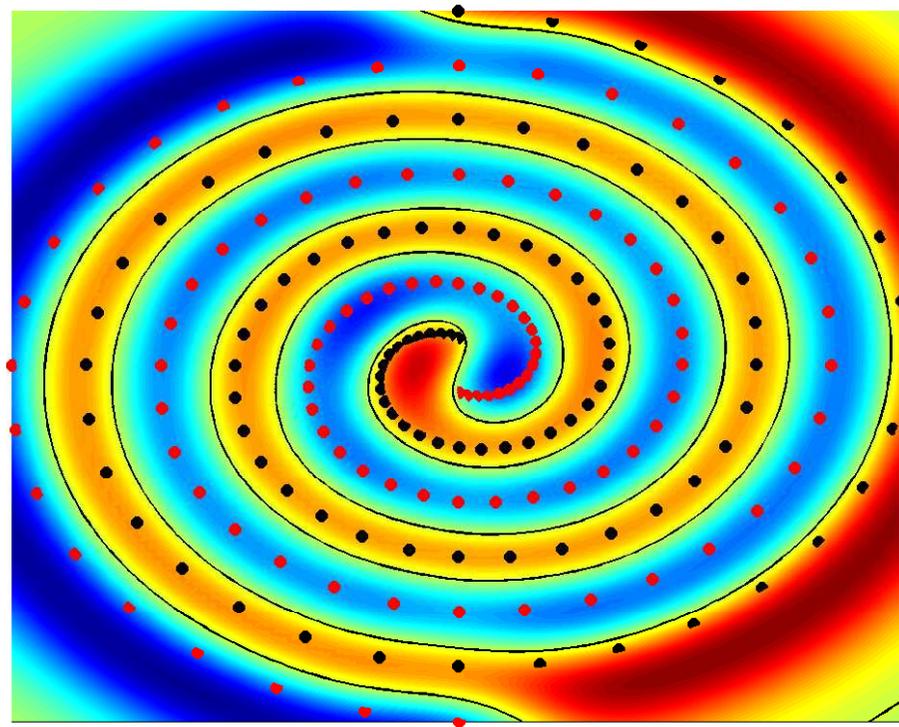
$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2)$$



Détecteur à noyau gaussien

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2)$$

pour le problème de Lang et Witbrock, 1988.



Plan de l'exposé

1. Introduction
2. Prétraitement
3. Méthodes Statistiques
4. Fonctions discriminantes
5. Classification non supervisée

Classifieur Bayésien (théorie)

Hypothèses :

- Probabilité d'apparition à priori de la classe ω_i : $P(\omega_i)$
- Densité de probabilité conditionnelle (à la classe ω_i) du vecteur x : $f(x|\omega_i)$

Règle de Bayes / Théorème des Probabilités Totales :

- Probabilité a posteriori que x appartienne à la classe ω_i : $P(\omega_i|x) = \frac{f(x|\omega_i)P(\omega_i)}{f(x)}$
- Densité de probabilité du vecteur x : $f(x) = \sum_{i=1}^K f(x|\omega_i)P(\omega_i)$

Conclusions : Le classifieur du maximum a posteriori (MAP)

$$d^*(x) = a_j \Leftrightarrow P(\omega_j|x) \geq P(\omega_k|x), \forall k \in \{1, \dots, K\} \quad (4)$$

Remarques : Probabilité d'erreur, Distance de Mahalanobis, Classification Bayésienne avec rejet

Apprentissage Supervisé

Méthodes paramétriques

- Estimation ponctuelle
- Estimation Bayésienne

Cas Gaussien : règle de décision linéaire ou quadratique

Méthodes non-paramétriques

- Fenêtres de Parzen
- Méthode des plus proches voisins
- Apprentissage Bayésien

Remarques :

- Règle des k plus proches voisins
- Inégalités : $P^* \leq P_1 \leq P^* \left(2 - \frac{K}{K-1} P^*\right) \dots$

Apprentissage Bayésien

- en traitement du signal

$$p(x|\omega) = \int p(x|\theta)p(\theta|\omega)d\theta$$

$$p(\theta|\omega) \propto \prod_{j=1}^n p(x_j|\theta)p(\theta)$$

- en traitement de la parole

$$P(\omega_i|x_1, \dots, x_p) \propto f(x_1, \dots, x_p|\omega_i)P(\omega_i)$$

avec $f(x_1, \dots, x_p|\omega_i)$ calculé à partir du canal acoustique (modèle de Markov caché ou mélange de Gaussiennes) et $P(\omega_i)$ dépend du modèle de langage ou du locuteur

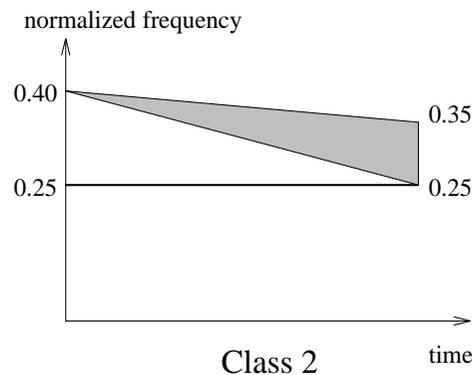
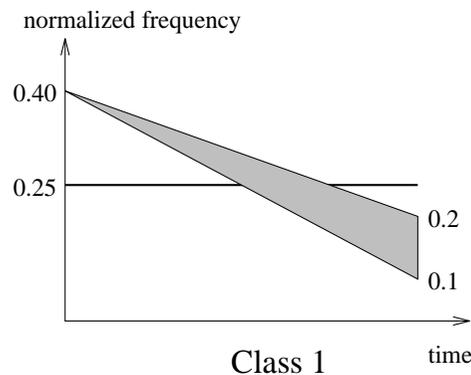
- en classification d'images

Utilisation des Champs de Markov

Application : Classification de chirps

$$\mathbf{x}[i] = \sum_{j=1}^2 a_j \cos(2\pi [\phi_j + f_j i + s_j i^2]) + \epsilon[i]$$

avec $i = \{0, \dots, N - 1\}$, $\epsilon[i] \sim \mathcal{N}(0, \sigma_\epsilon^2)$, $N = 128$ (nombre de points), $a_1 = a_2 = 1$ (amplitudes des composantes), $\sigma_\epsilon^2 = 2$ ($SNR = 10 \log_{10} \left(\frac{a_1^2 + a_2^2}{\sigma_\epsilon^2} \right) = 0$ dB) (variance du bruit), $f_1 = 0.25$, $f_2 = 0.4$ (Fréquences), phases ϕ_1 et ϕ_2 uniformes sur $[0, 1]$, $s_1 = 0$ (1ère composante = harmonique pure)



Définition des classes :

- Classe ω_1 : $p(s_2) = \mathcal{U}\left(\frac{-0.30}{2N}, \frac{-0.20}{2N}\right)$
- Classe ω_2 : $p(s_2) = \mathcal{U}\left(\frac{-0.15}{2N}, \frac{-0.05}{2N}\right)$

Résultat de l'apprentissage

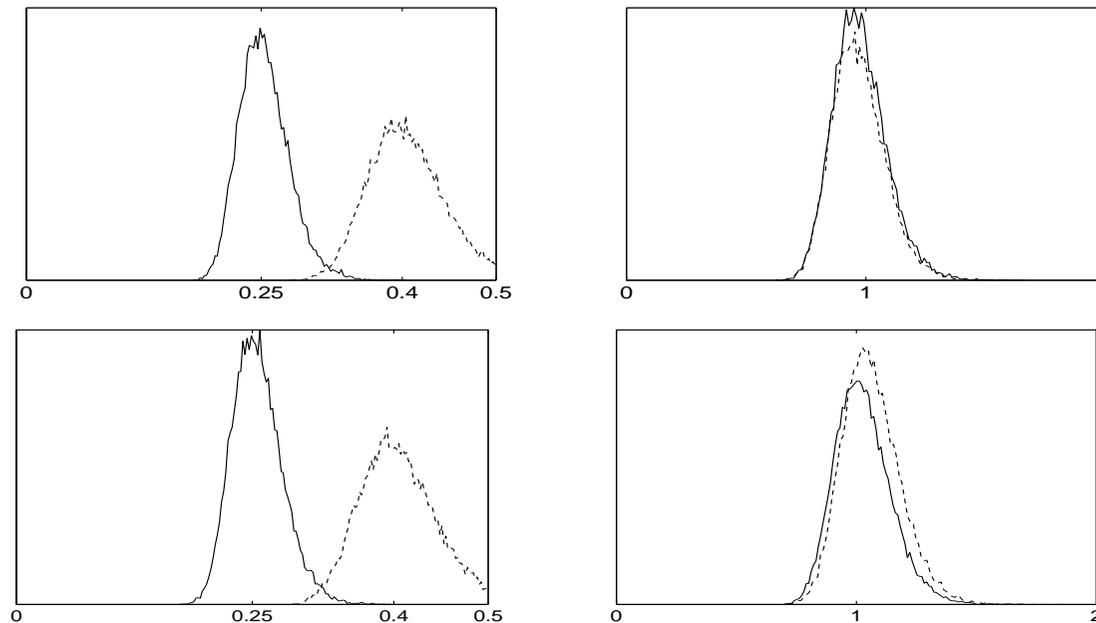


Figure 1: Histogrammes associés à $p(\mathbf{m}_f[1]|\mathcal{X})$ (ligne continue), $p(\mathbf{m}_f[2]|\mathcal{X})$ (ligne pointillée), $p(\mathbf{m}_a[1]|\mathcal{X})$ (ligne continue), $p(\mathbf{m}_a[2]|\mathcal{X})$ (ligne pointillée), pour les classes ω_1 (figures du dessus) et ω_2 (figure du dessous).

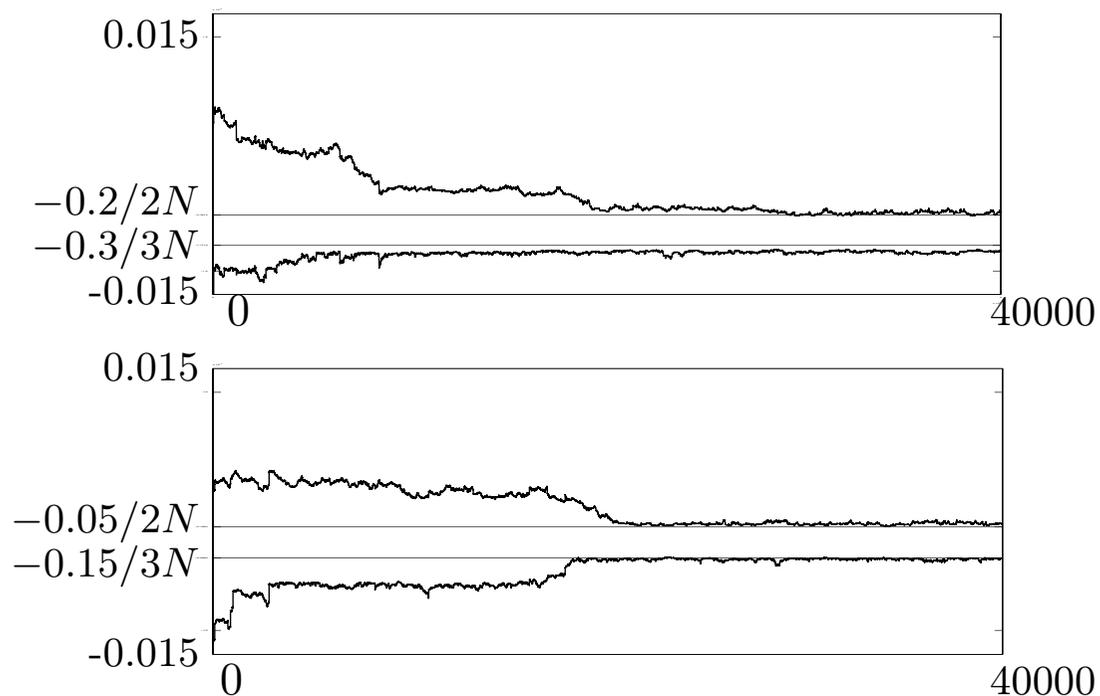


Figure 2: Eléments d'une chaîne de Markov générée selon $p(\mathbf{s}_{\min}[2]|\mathcal{X})$, $p(\mathbf{s}_{\max}[2]|\mathcal{X})$ (ligne du dessus - classe ω_1) et $p(\mathbf{s}_{\min}[2]|\mathcal{X})$, $p(\mathbf{s}_{\max}[2]|\mathcal{X})$ (ligne du dessous - classe ω_2).

Plan de l'exposé

1. Introduction
2. Prétraitement
3. Méthodes Statistiques
4. Fonctions discriminantes
5. Classification non supervisée

Fonctions discriminantes

- Fonctions discriminantes linéaires
- Machines à vecteurs supports (SVM)
- Réseaux neuronaux

Fonctions discriminantes linéaires

Classifieur MAP (2 classes) :

Classe ω_1 si $\hat{f}(x|\omega_1)P(\omega_1) \geq \hat{f}(x|\omega_2)P(\omega_2) \Leftrightarrow g(x) \geq 0$

On peut imposer une **forme linéaire** pour $g(x)$ i.e. $g(x) = w^T x$ avec $x = (x_1, \dots, x_p, 1)^T$ et se fixer la règle de classification

$$\begin{aligned}d(x) &= \omega_1 \Leftrightarrow g(x) > 0 \\d(x) &= \omega_2 \Leftrightarrow g(x) < 0\end{aligned}\tag{5}$$

Problème : comment déterminer w à partir d'une base d'apprentissage $\mathcal{B} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ où x_1, \dots, x_n sont n vecteurs de \mathbb{R}^p et y_1, \dots, y_n sont des variables binaires telles que

$$\begin{aligned}y_i &= 1 \text{ si } x_i \in \omega_1 \\y_i &= -1 \text{ si } x_i \in \omega_2\end{aligned}$$

Algorithme du perceptron

Hypothèse : classes linéairement séparables

Problème : minimisation de

$$J(w) = \sum_{x \in Y} (-\delta_x w^t x) \quad (6)$$

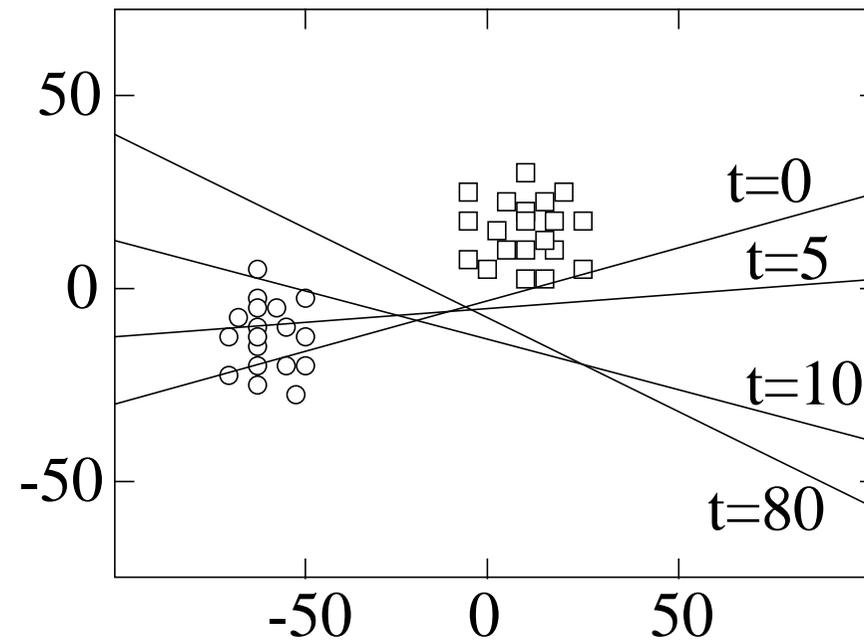
avec $\delta_x = 1$ si $x \in \omega_1$ et $\delta_x = -1$ si $x \in \omega_2$ et Y est l'ensemble des vecteurs de la base d'apprentissage mal classés par la règle (5).

Mise à jour de w :

$$w_k(n+1) = w_k(n) + \mu \sum_{x \in Y} (\delta_x x) \quad (7)$$

Remarques : paramètre d'apprentissage μ ou $\mu(n)$, **convergence après un nombre fini d'itérations.**

Exemple d'école



Algorithmes de descente

- **Filtre optimal de Wiener-Hopf**

$$J(w) = \frac{1}{2} E [e^2(n)] = \frac{1}{2} E [(d(n) - w^t x(n))^2] \quad (8)$$

- **Méthode de la plus profonde descente**

$$w_k(n+1) = w_k(n) - \mu \left. \frac{\partial J(w)}{\partial w_k} \right|_{w_k = w_k(n)} \quad (9)$$

- **Algorithme LMS**

$$\hat{w}_k(n+1) = \hat{w}_k(n) + \mu x_k(n) [d(n) - y(n)] \quad (10)$$

Machines à vecteurs supports (SVMs)

Marge d'un couple et d'une base d'apprentissage :

$$\gamma_i(\tilde{w}) = \frac{y_i (w^T x_i - b)}{\|w\|} \quad \gamma_{\mathcal{B}}(\tilde{w}) = \min_{i \in \{1, \dots, n\}} \frac{y_i (w^T x_i - b)}{\|w\|} \quad (11)$$

avec $\tilde{w} = (w, b)$.

Hyperplan optimal

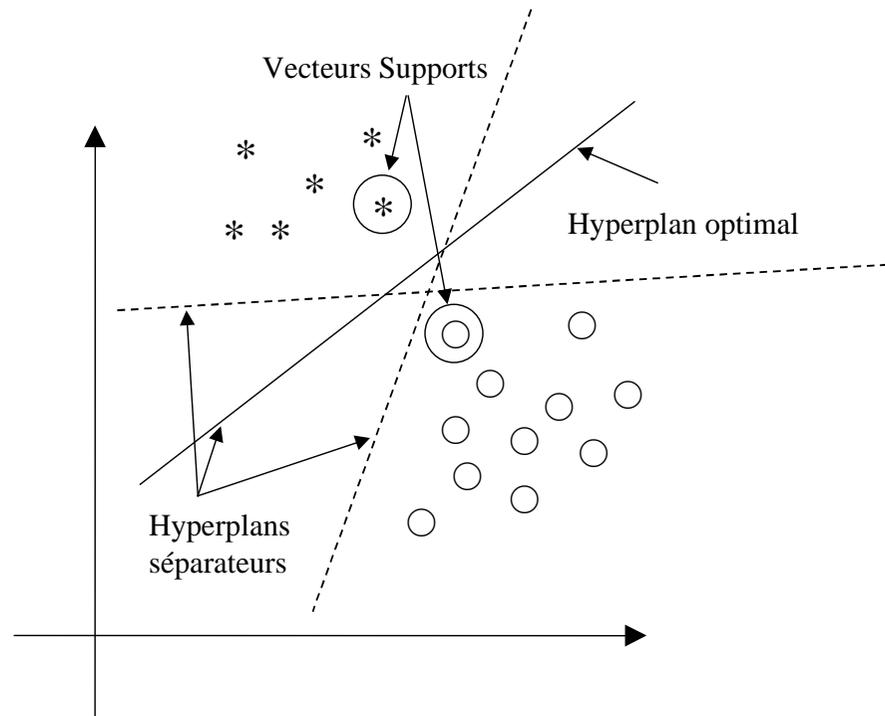
L'hyperplan optimal maximise la marge de la base d'apprentissage $\gamma_{\mathcal{B}}(\tilde{w})$

Vecteurs supports

Ce sont les vecteurs qui sont à distance minimale de l'hyperplan séparateur tels que

$$\gamma_i(\tilde{w}) = \gamma_{\mathcal{B}}(\tilde{w})$$

Illustration



Détermination de l'hyperplan optimal

Problème d'optimisation sous contraintes

$$w = \arg \max \gamma_{\mathcal{B}}(\tilde{w}) \text{ sous les contraintes } \min_{i \in \{1, \dots, n\}} y_i (w^T x_i - b) = 1$$

Remarques :

- si $y_i (w^T x_i - b) = a$, en changeant \tilde{w} en \tilde{w}/a , on a $y_i (w^T x_i - b) = 1$.
- **Problème équivalent**

$$\text{Minimiser } \frac{1}{2} \|w\|^2 \text{ sous les contraintes } y_i (w^T x_i - b) \geq 1, \forall i$$

- **Cas non séparable**

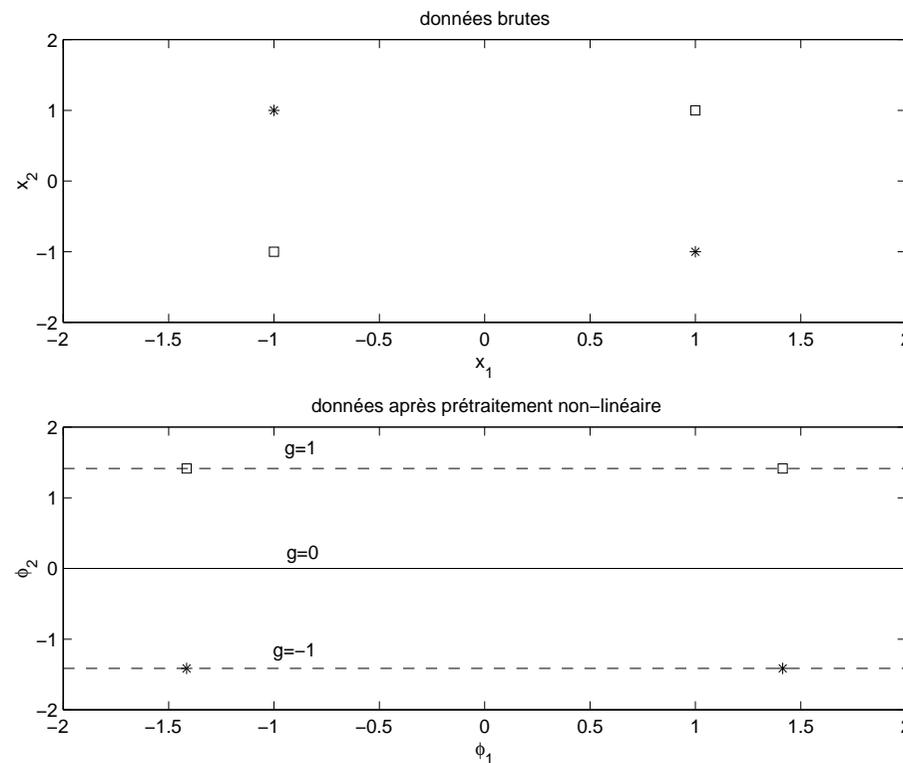
$$\text{minimiser } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \text{ sous les contraintes } y_i (w^T x_i - b) \geq 1 - \xi_i, \forall i$$

Prétraitement non-linéaire

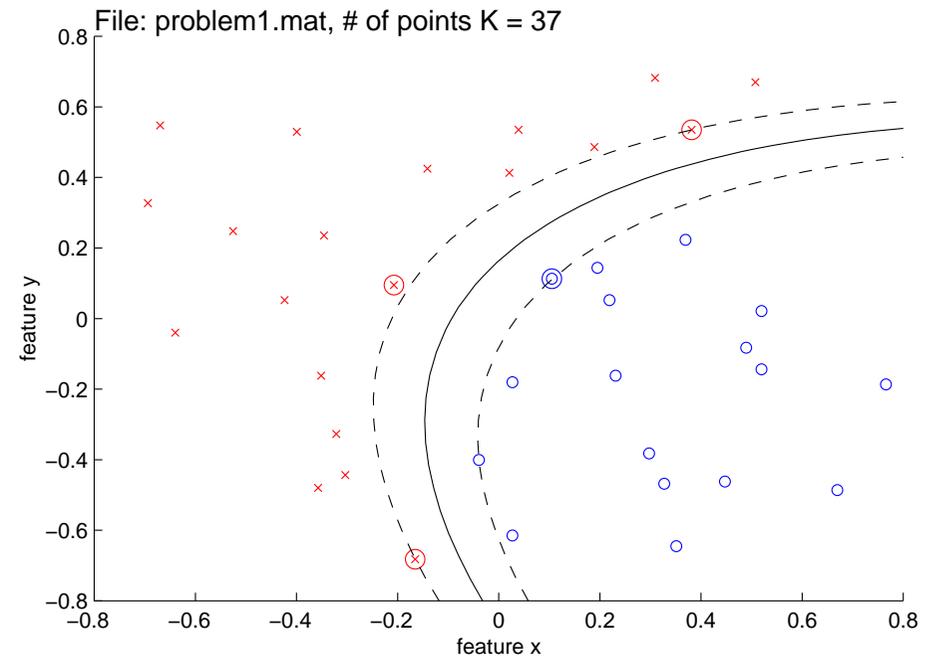
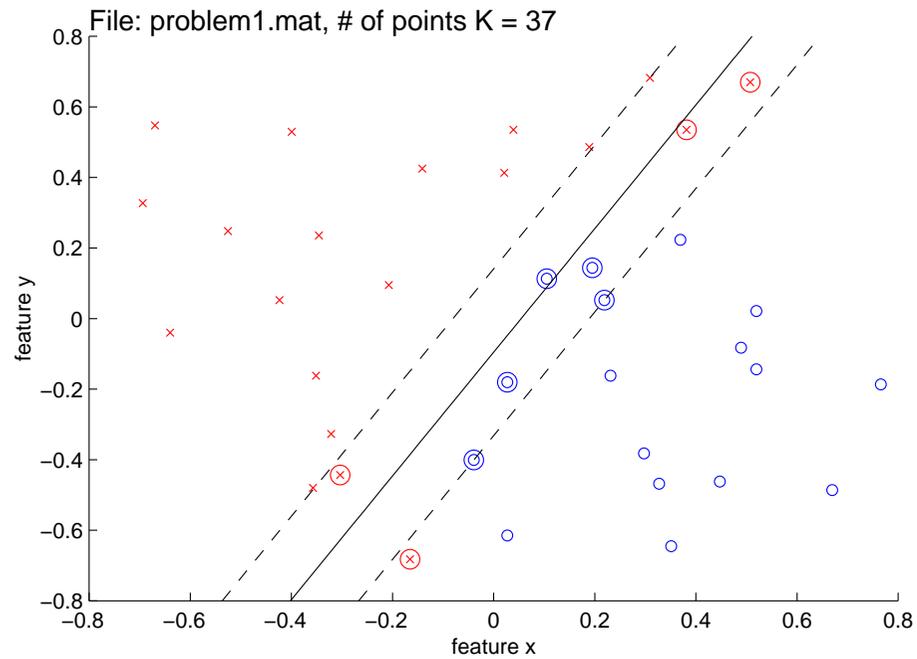
Exemple : $\chi_1 = \{(1, 1), (-1, -1)\}$ et $\chi_2 = \{(1, -1), (-1, 1)\}$.

$$\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^6$$

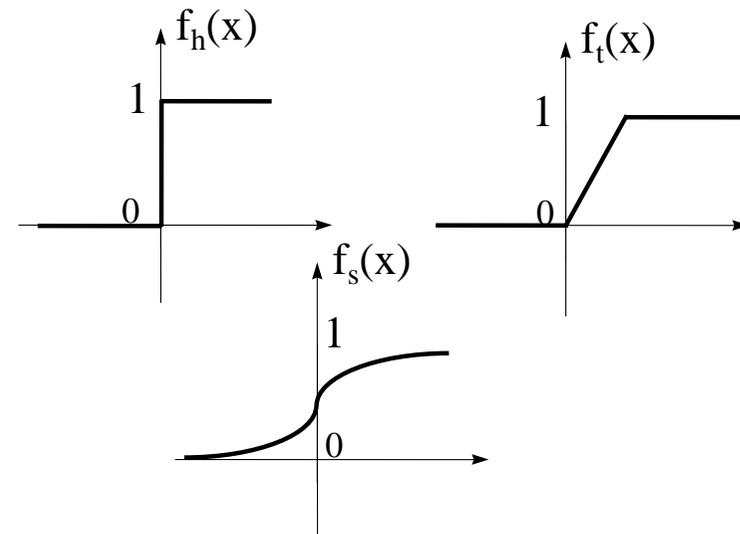
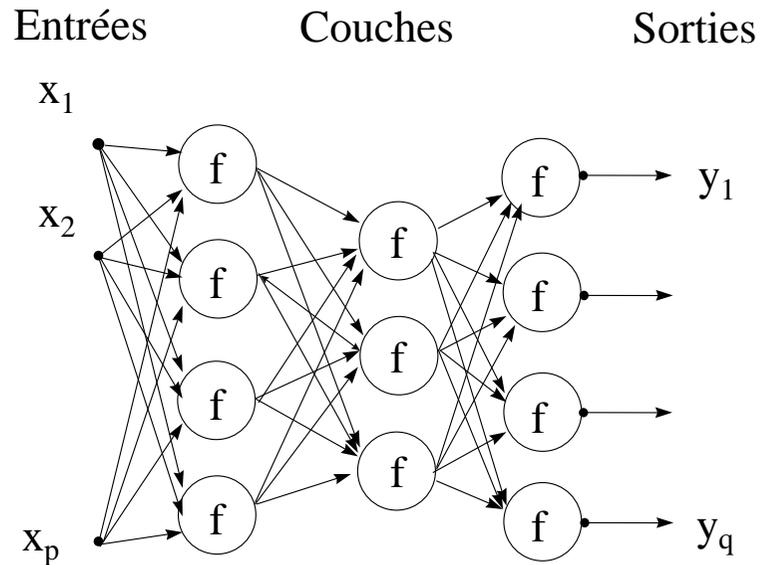
$$(x_1, x_2)^T \mapsto (\sqrt{2}x_1, \sqrt{2}x_1x_2, 1, \sqrt{2}x_2, x_1^2, x_2^2)^T$$



Exemple



Réseaux Neuronaux

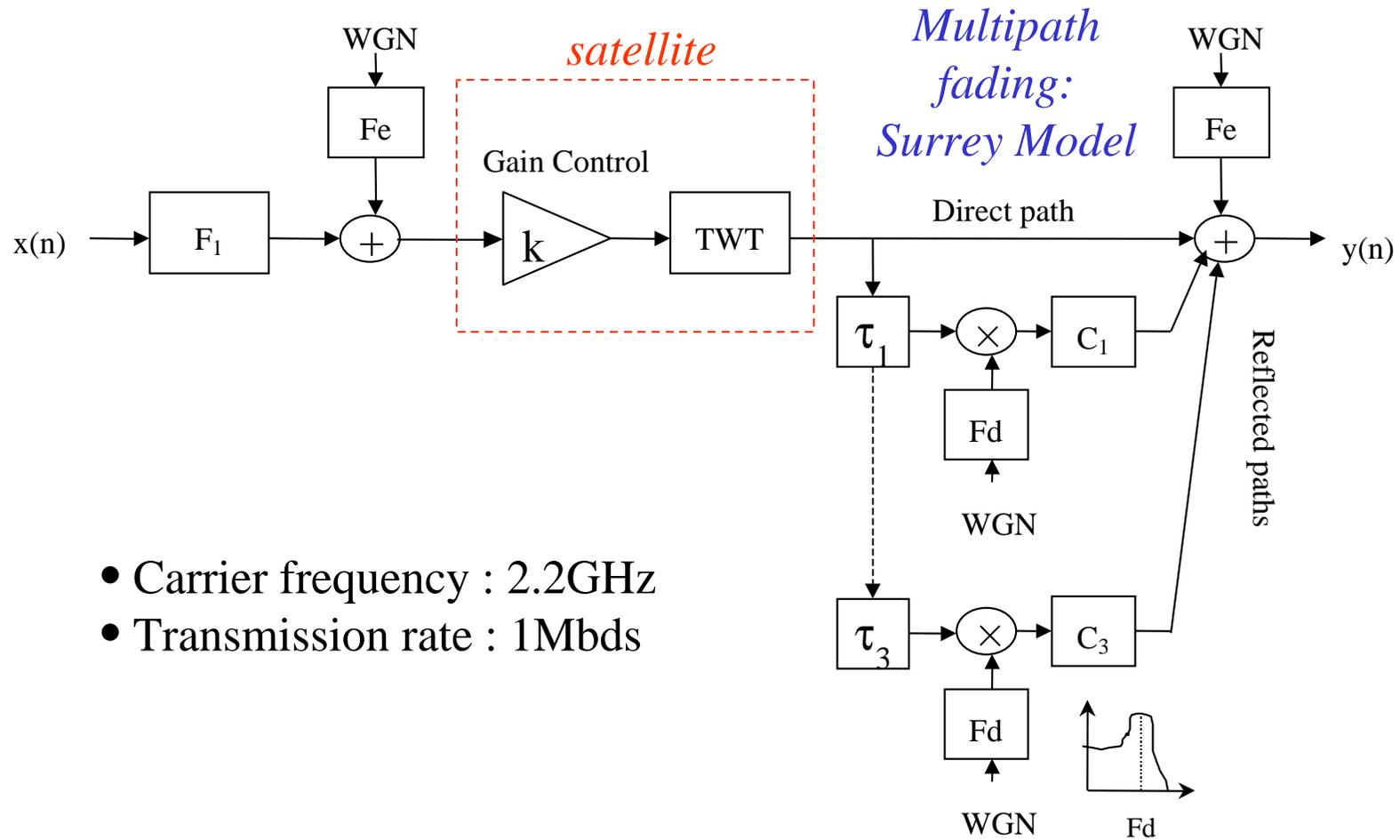


Relation entrée-sortie : $y_{i,l+1} = f \left(\sum_{j=1}^p w_{ij} y_{j,l} \right)$

Règle d'apprentissage : algorithme de rétropropagation du gradient

Application : égalisation de canaux non-linéaires

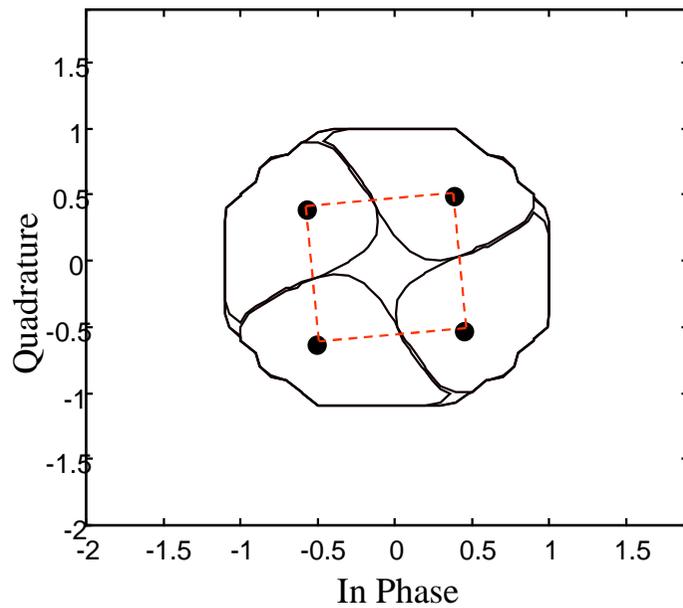
S-UMTS Channel Model



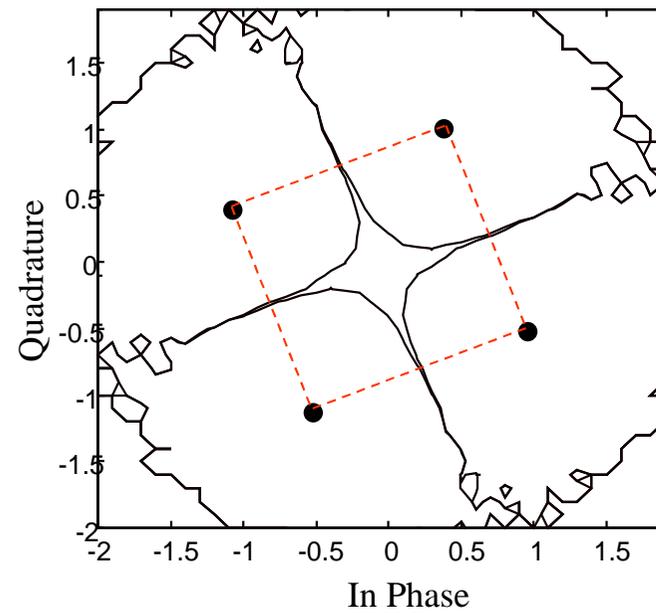
- Carrier frequency : 2.2GHz
- Transmission rate : 1Mbds

Consequences of the non-linearity

Distortion of a 4-QAM signal



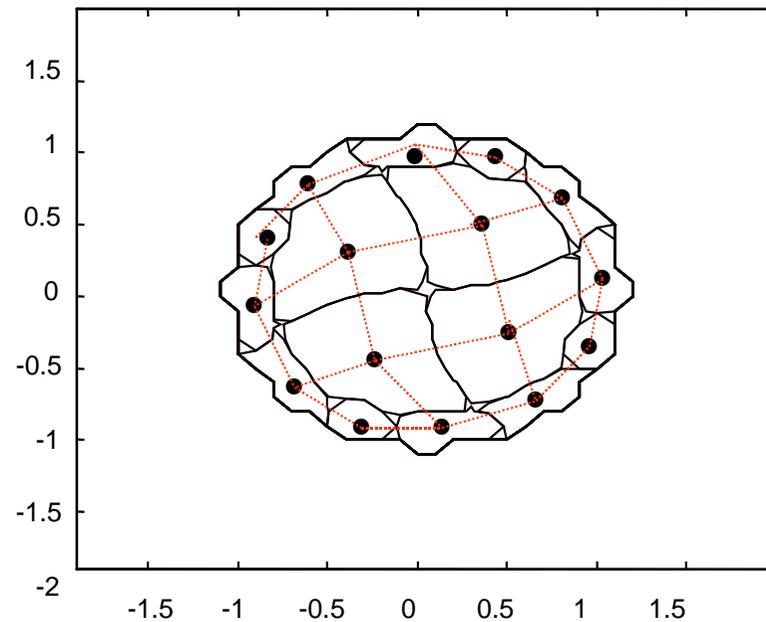
backoff 0dB
no downlink noise



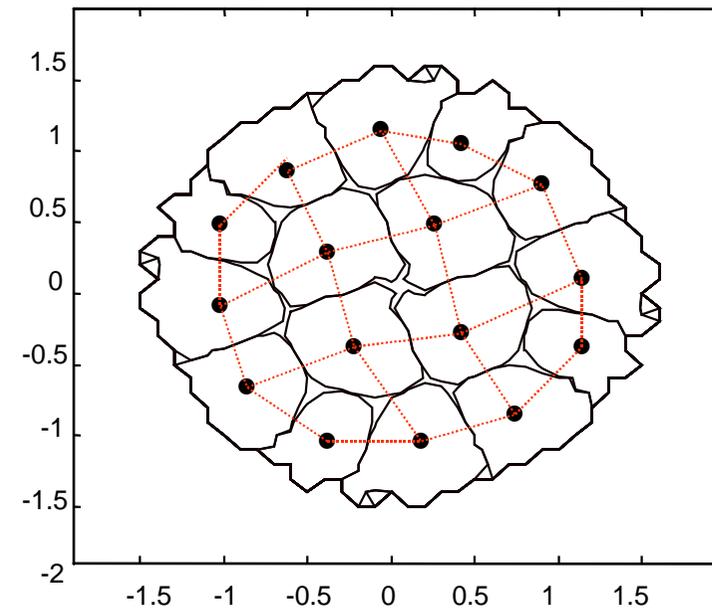
backoff 0dB
15dB downlink SNR

Consequences of the non-linearity

Distortion of a 16-QAM signal

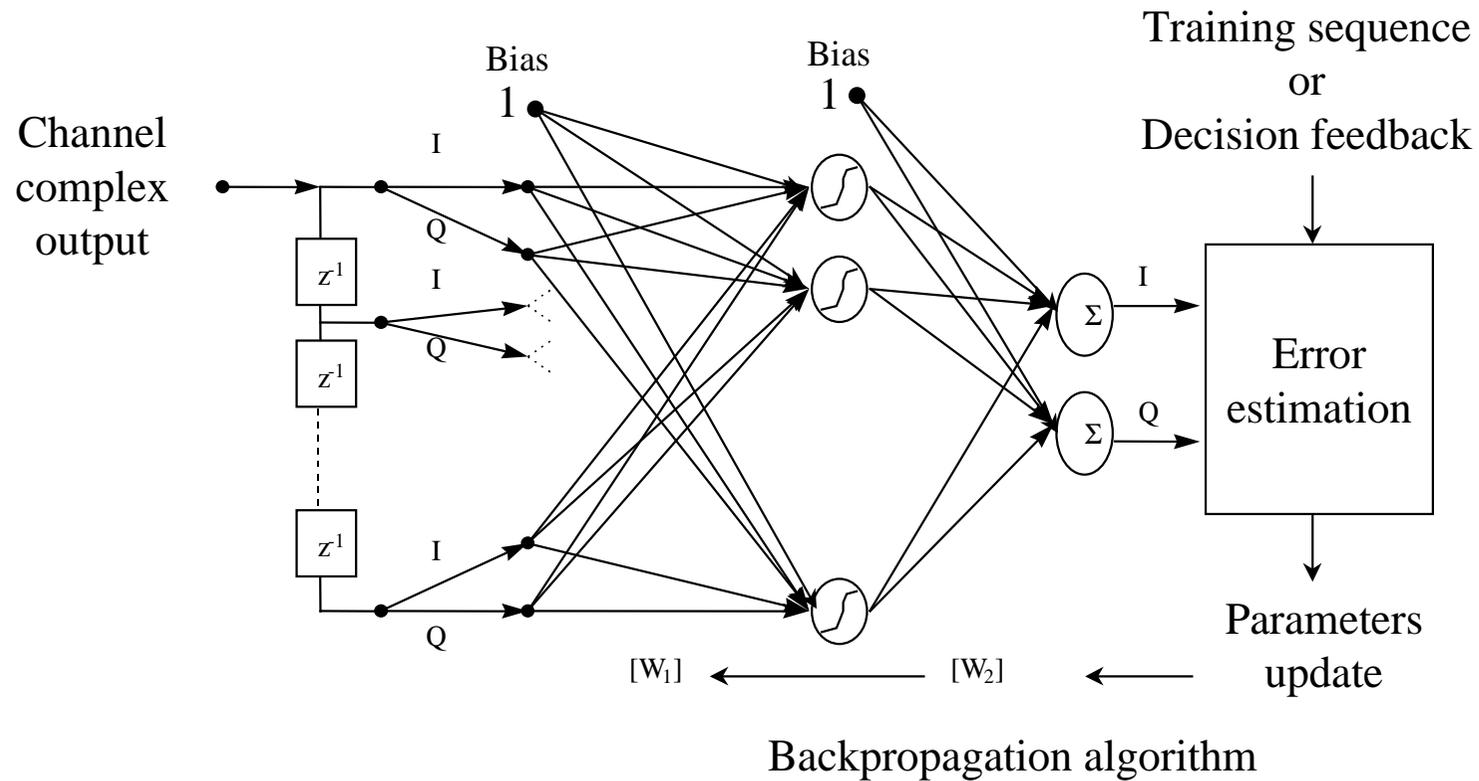


backoff -1dB
no downlink noise



backoff -1dB
15dB downlink noise

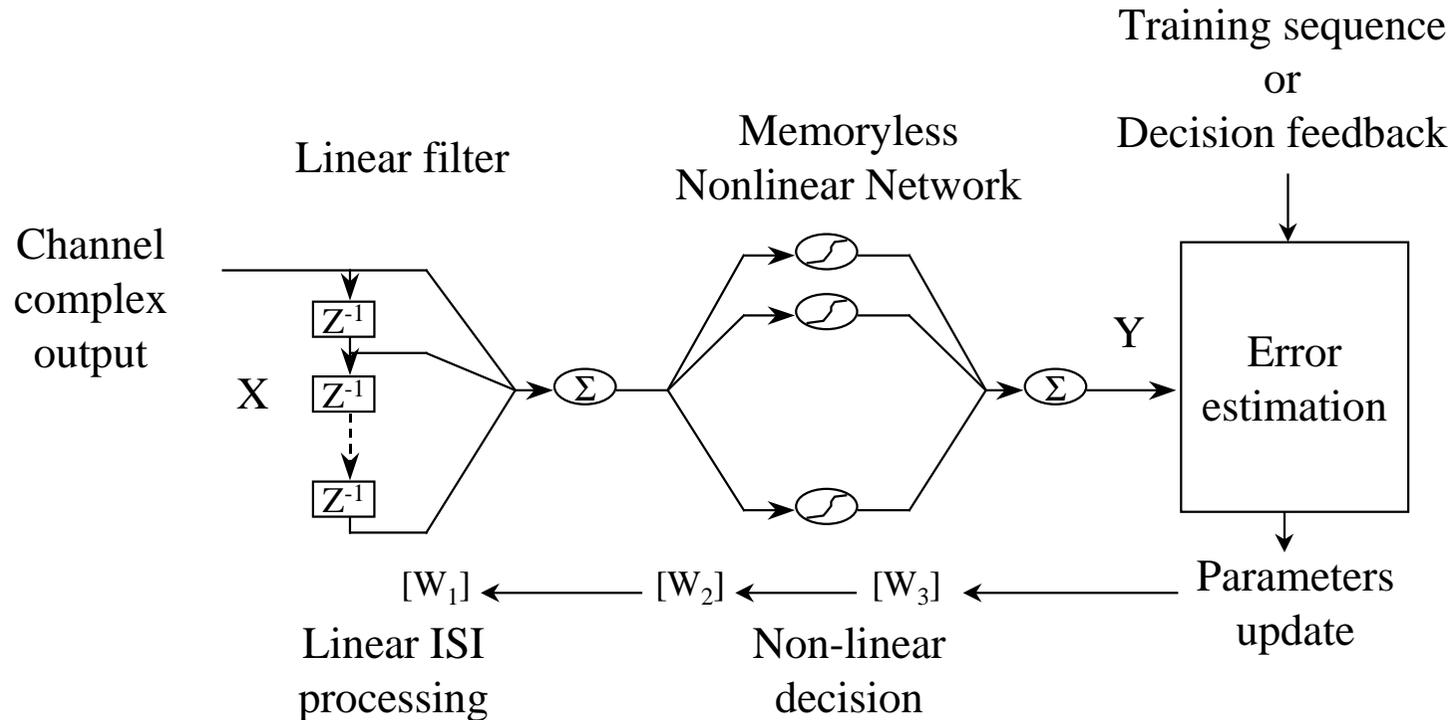
Neural Network Equalizers : Multilayer Perceptron (MLP)



- **Advantages** : universal approximation theorem, parallelism.
- **Drawback** : slow convergence.

Neural Network Equalizers :

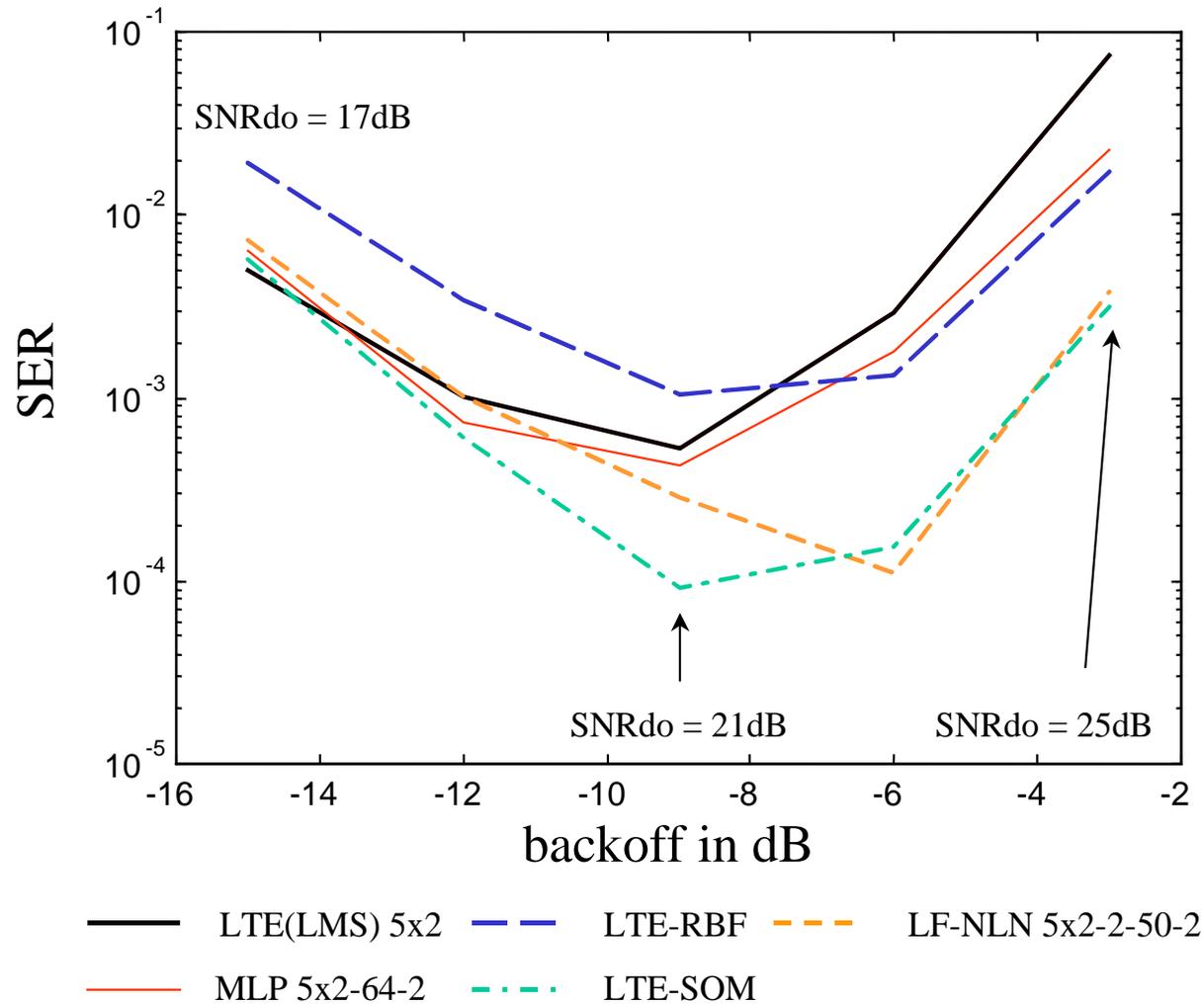
Linear Filter - NonLinear Network (LF-NLN)



- **Advantage** : More simple than the MLP.
- **Drawback** : Non optimal.

NN equalization of 16-QAM signals, S-UMTS

Backoff influence (16-QAM)



Plan de l'exposé

1. Introduction
2. Prétraitement
3. Méthodes Statistiques
4. Fonctions discriminantes
5. Classification non supervisée

Classification non supervisée

Base d'apprentissage : $\mathcal{B} = \{x_1, \dots, x_n\}$ où x_1, \dots, x_n sont n vecteurs de \mathbb{R}^p

- **Méthodes d'optimisation**

Apprentissage d'un mélange de lois

Erreur quadratique minimale (ISODATA)

Algorithmes de coalescence floue

- **Classification hiérarchique** Méthode des distances

Méthode des moments d'ordre 2

Hiérarchie ascendante ou descendante

Motivations

- Apprentissage supervisé coûteux

- Cas où les formes évoluent au cours du temps

- Donne des informations sur la structure des données (détection d'outliers)

Méthodes d'optimisation

Solution optimale en K classes

$$P(N, K) = \frac{1}{K!} \sum_{i=0}^K i^N (-1)^{K-i} C_K^i \quad K < N$$

⇒ recherche d'une partition intéressante dans un sous-ensemble approprié

Partition d'erreur quadratique minimale

- Erreur quadratique d'une partition : $E^2 = \sum_{i=1}^K E_i^2$ avec $E_i^2 = \sum_{k=1}^{N_i} d^2(x_k, g_i)$
- Propriété : erreur quadratique de \mathcal{B} = erreur intraclasse + erreur interclasse

$$\sum_{i=1}^K \sum_{k=1}^{N_i} d^2(x_k, g) = E^2 + \sum_{i=1}^K N_i d^2(g_i, g)$$

Algorithme ISODATA

Recherche d'une partition pour un minimum local de E^2

1) Choix initial du nombre de classes

Choix des barycentres de ces classes.

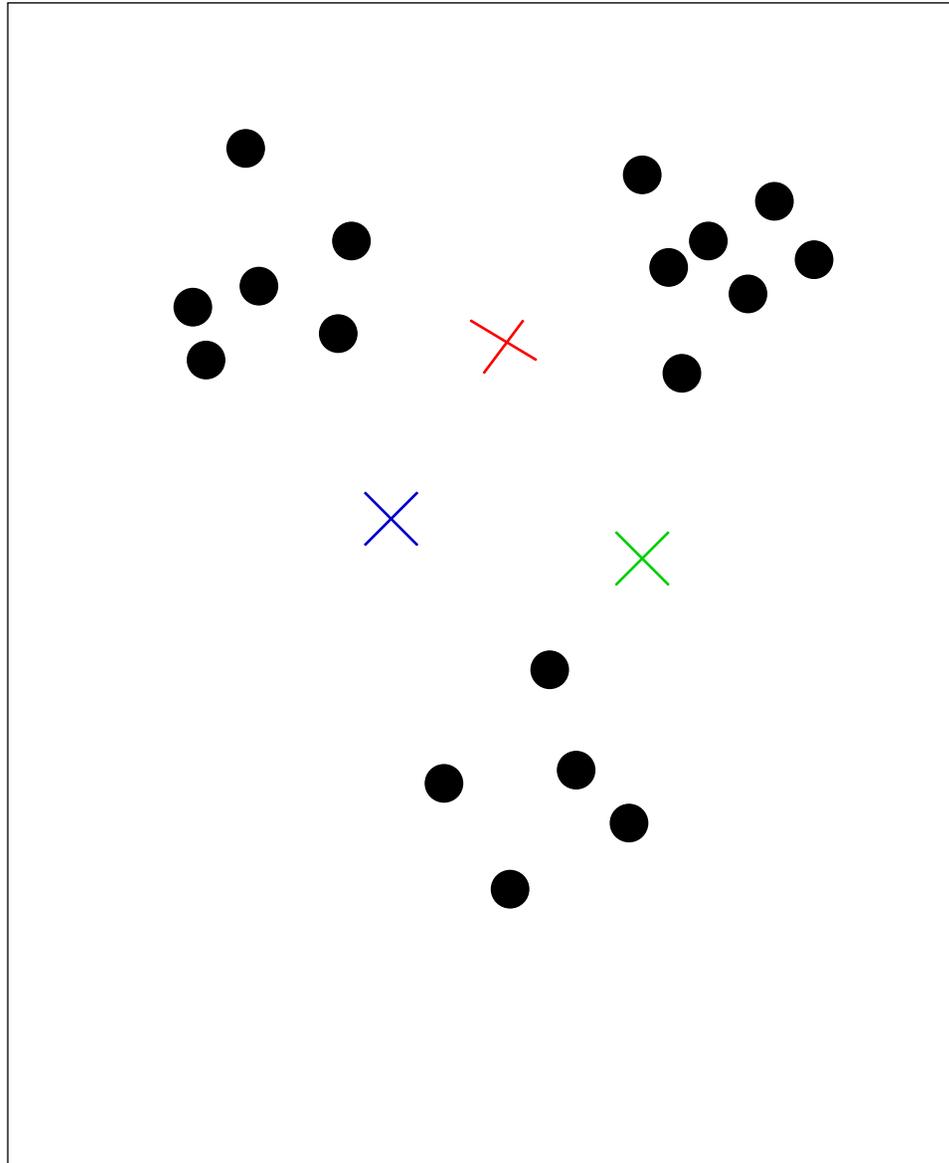
2) Affectation d'un vecteur x_i à la classe ω_j telle que

$$d(x_i, g_j) = \inf_k d(x_i, g_k)$$

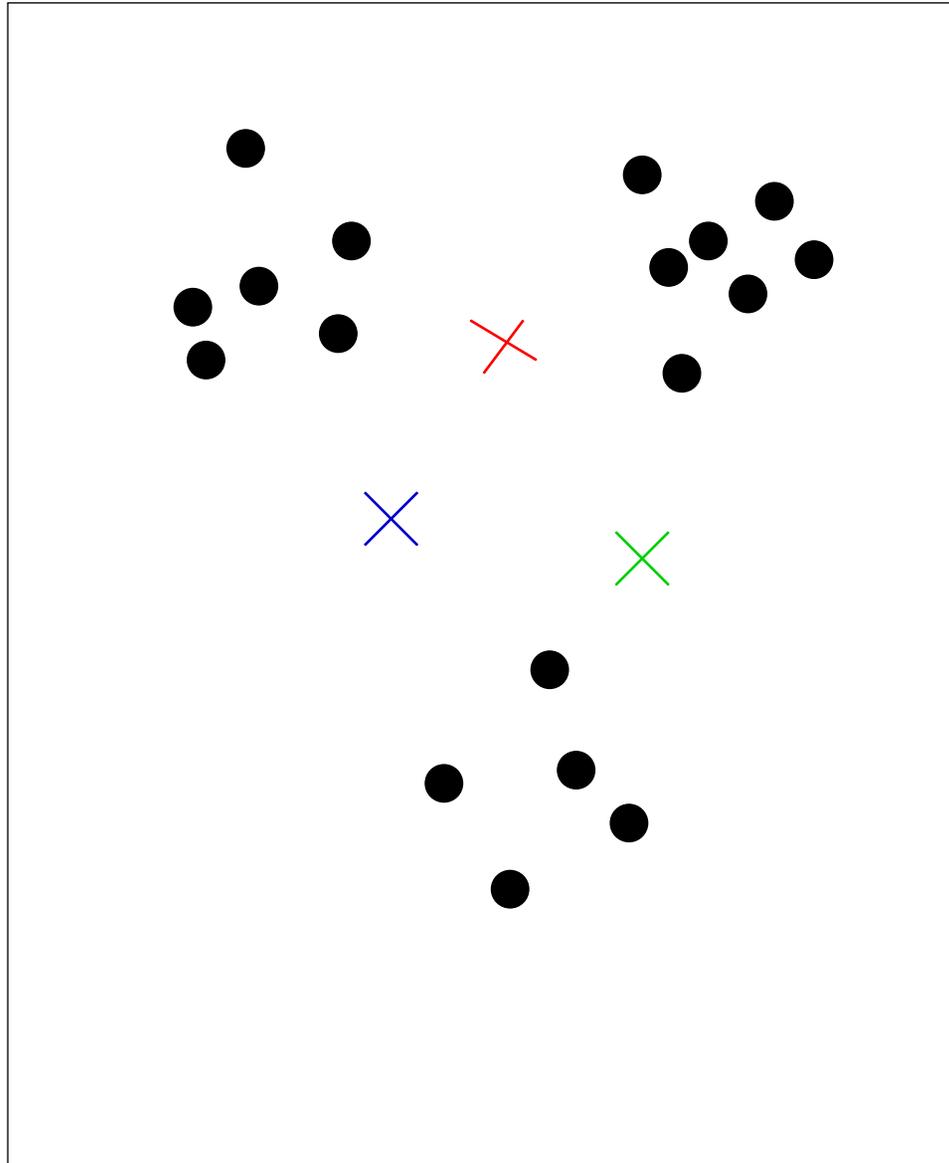
3) Calcul du centre de gravité g_k^* de chaque nouvelle classe ainsi formée ω_k^*

4) Répétition des étapes 2) et 3) jusqu'à convergence

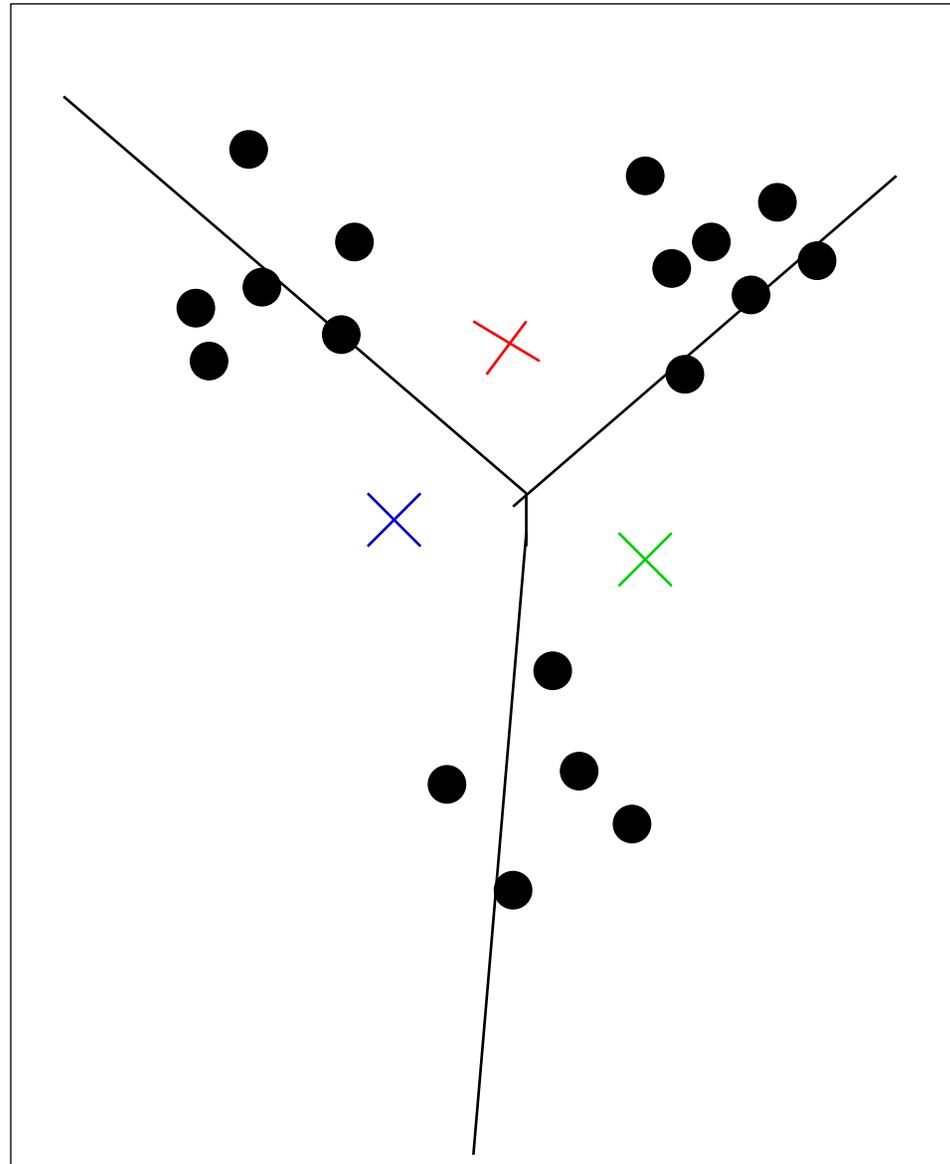
K-means



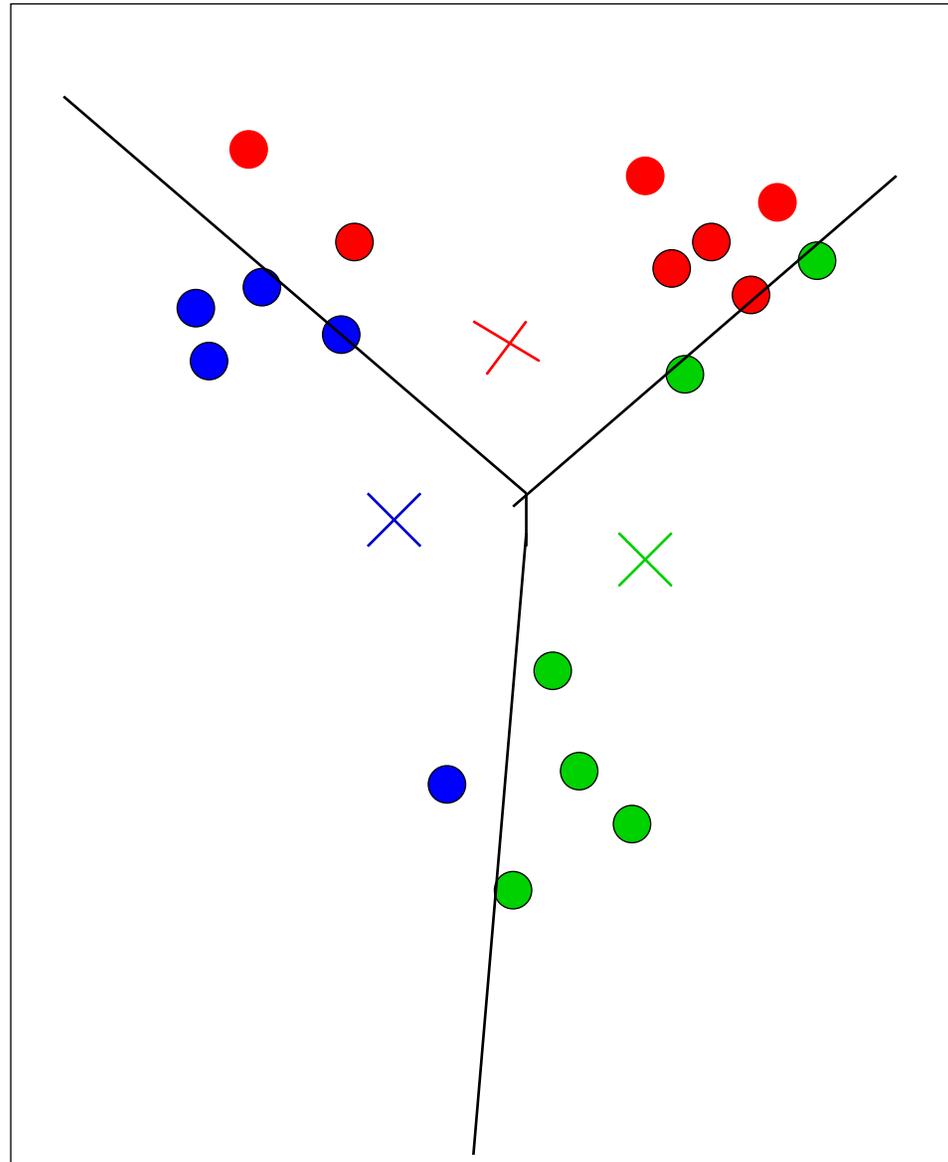
K-means



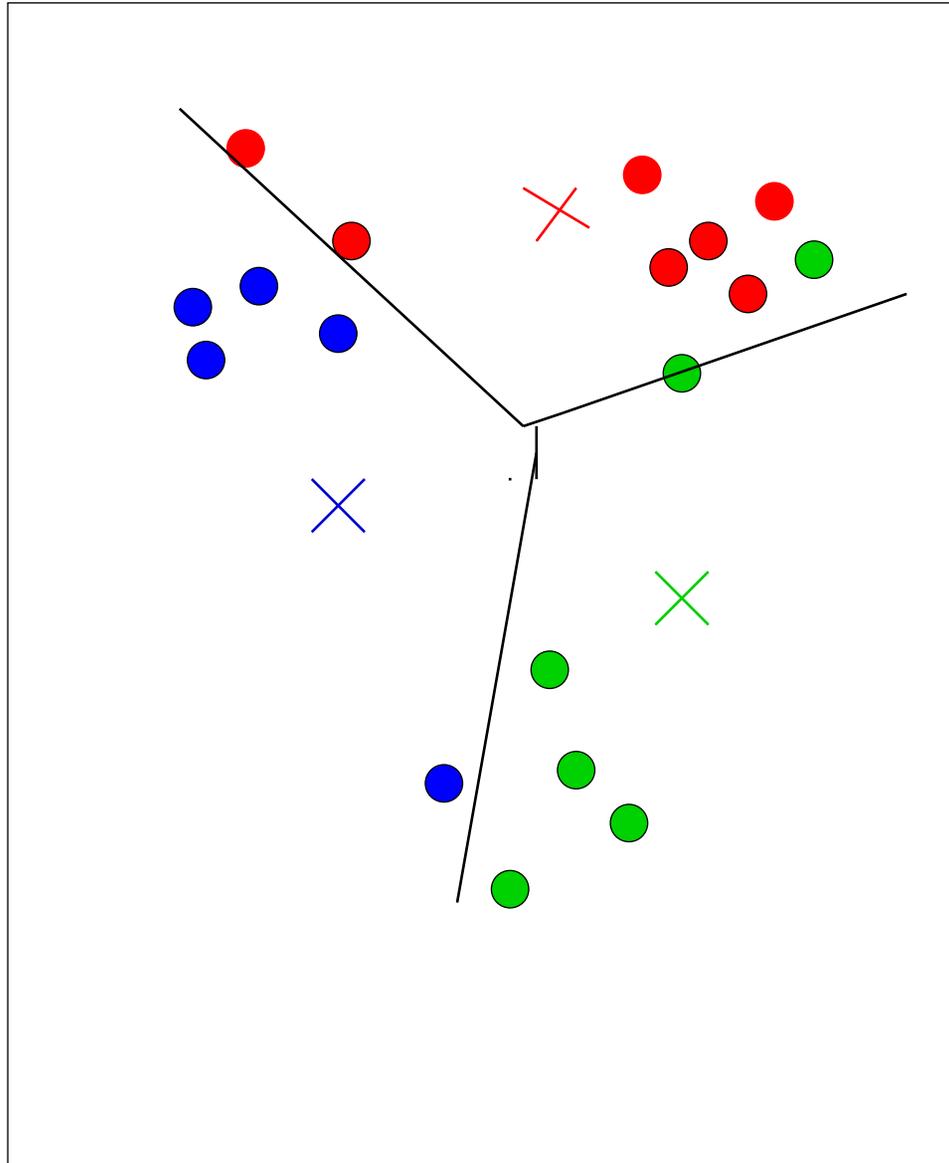
K-means



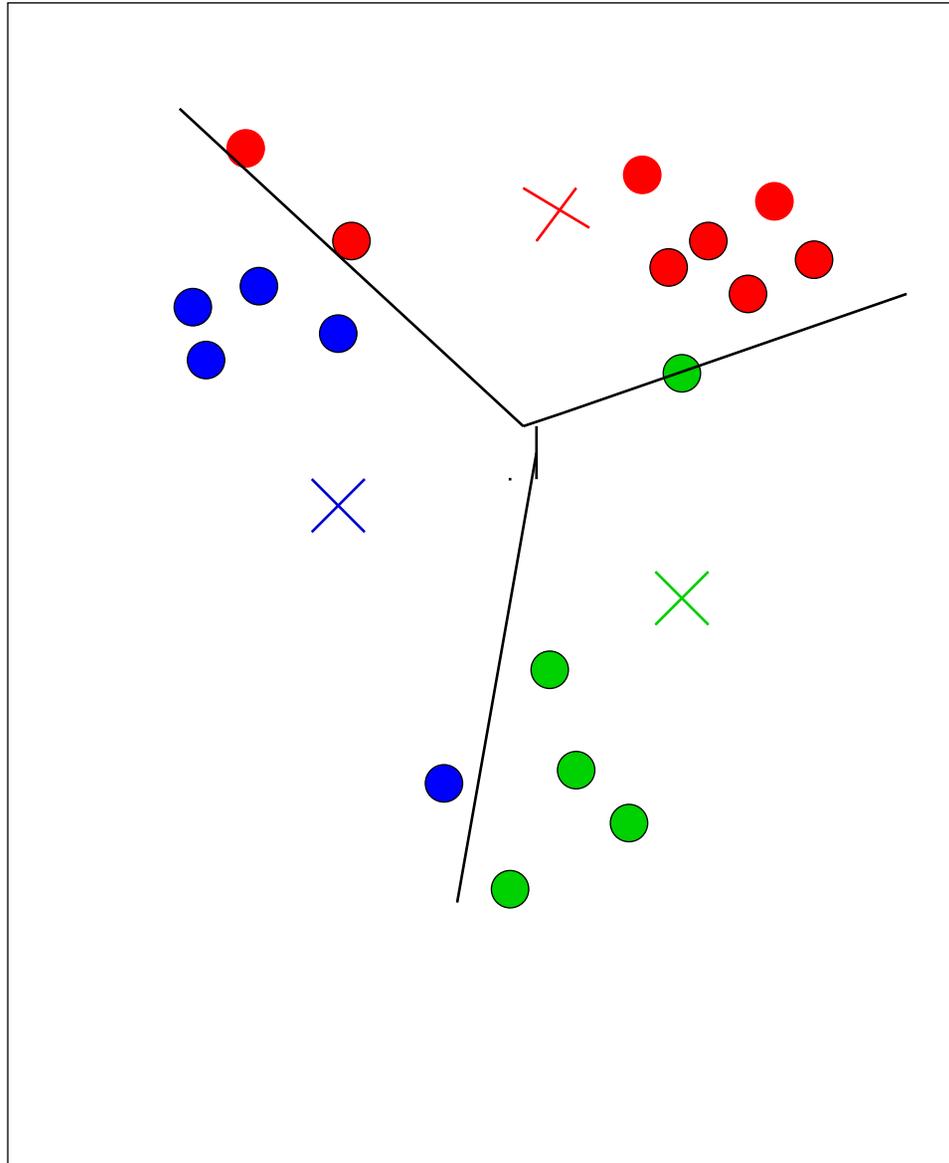
K-means



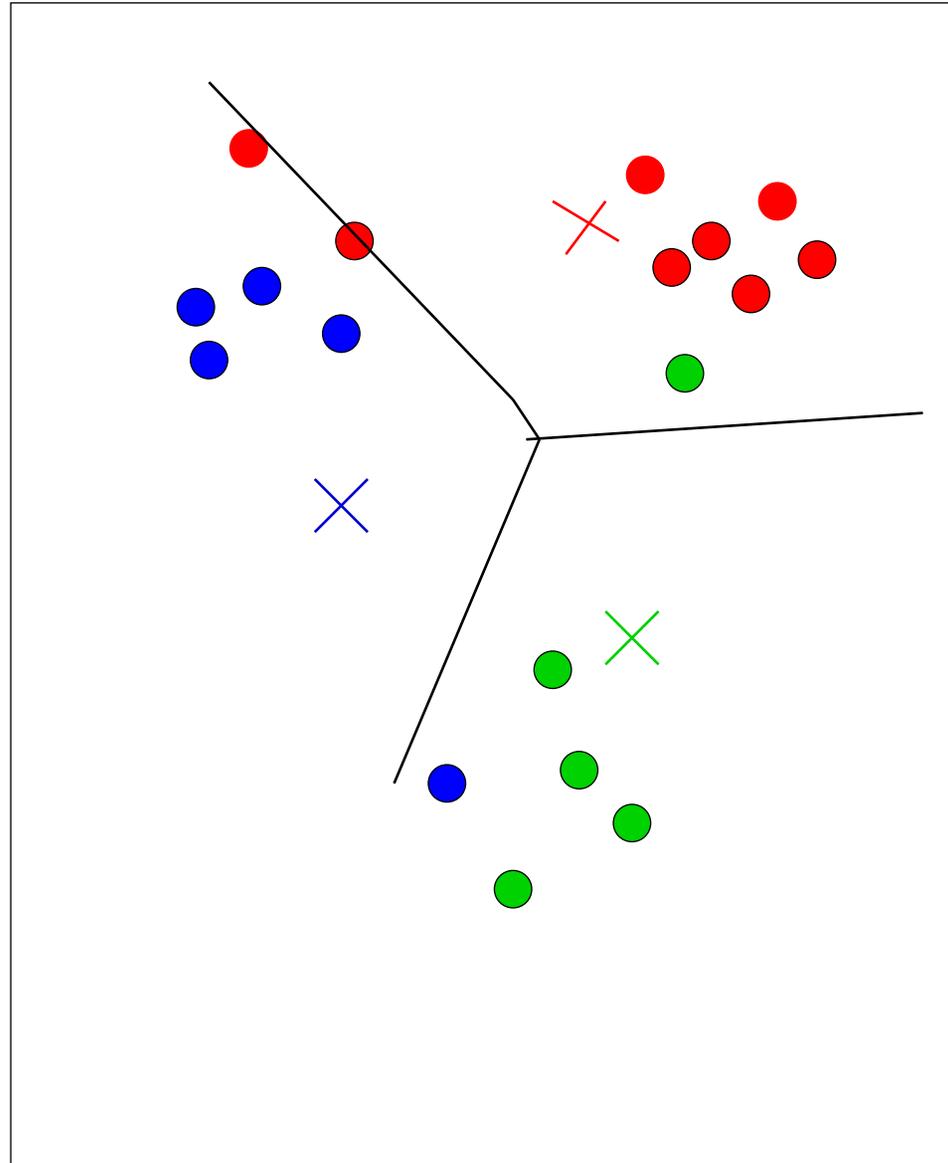
K-means



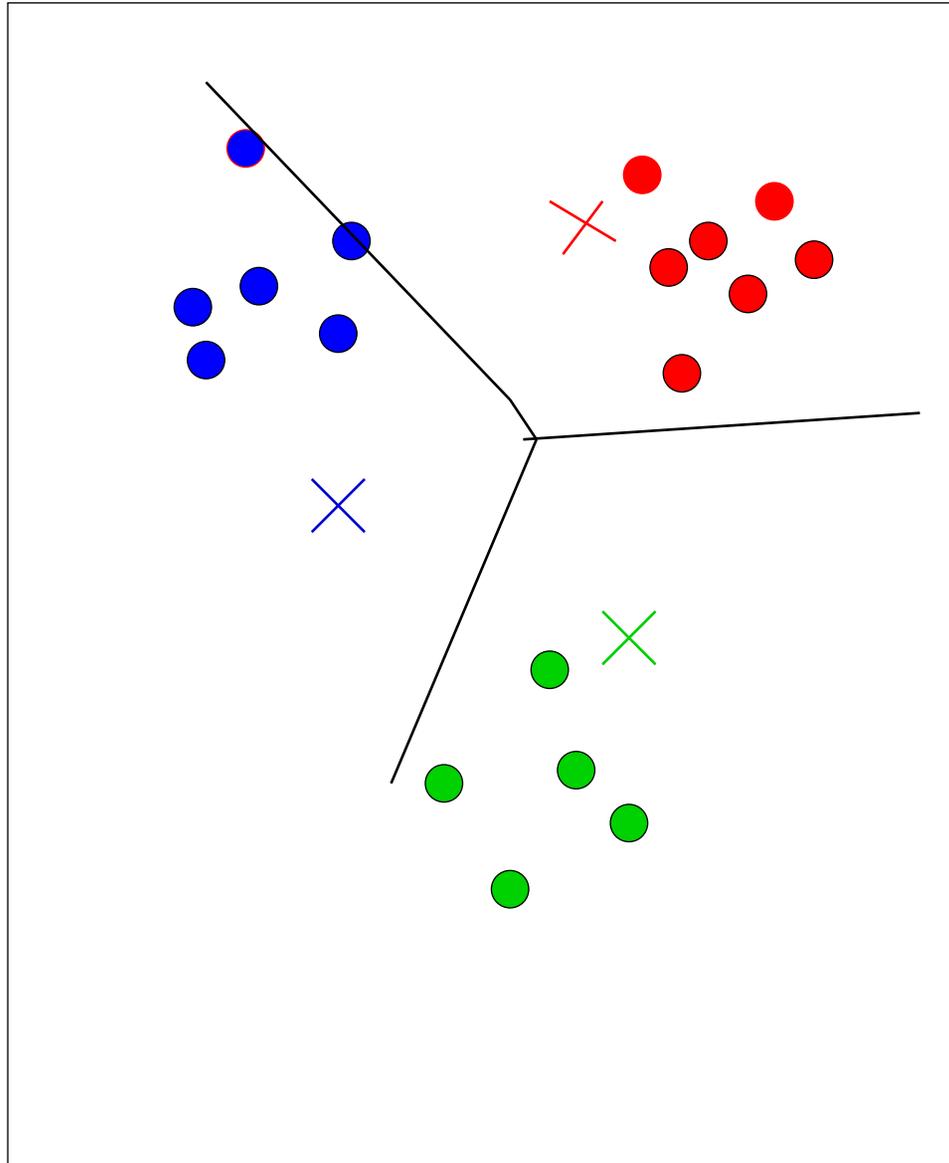
K-means



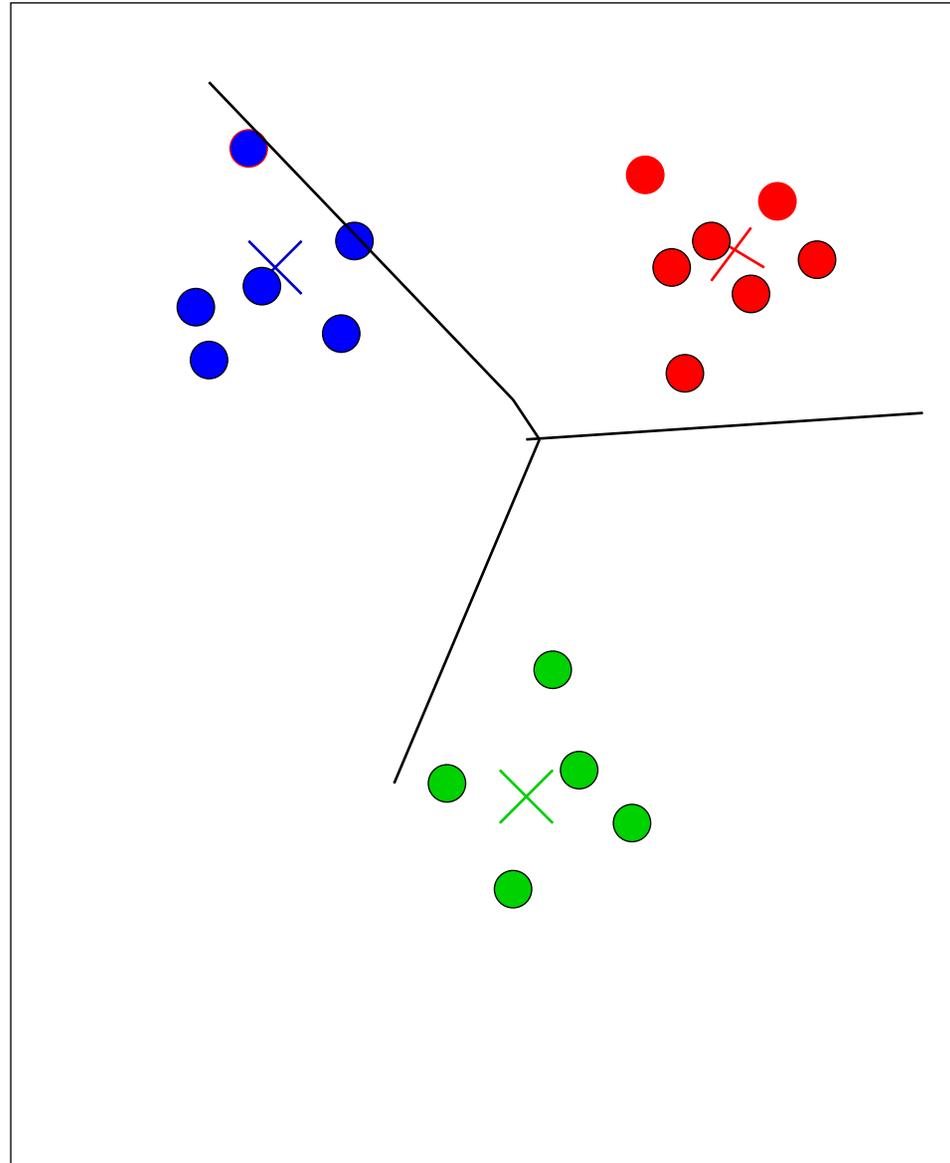
K-means



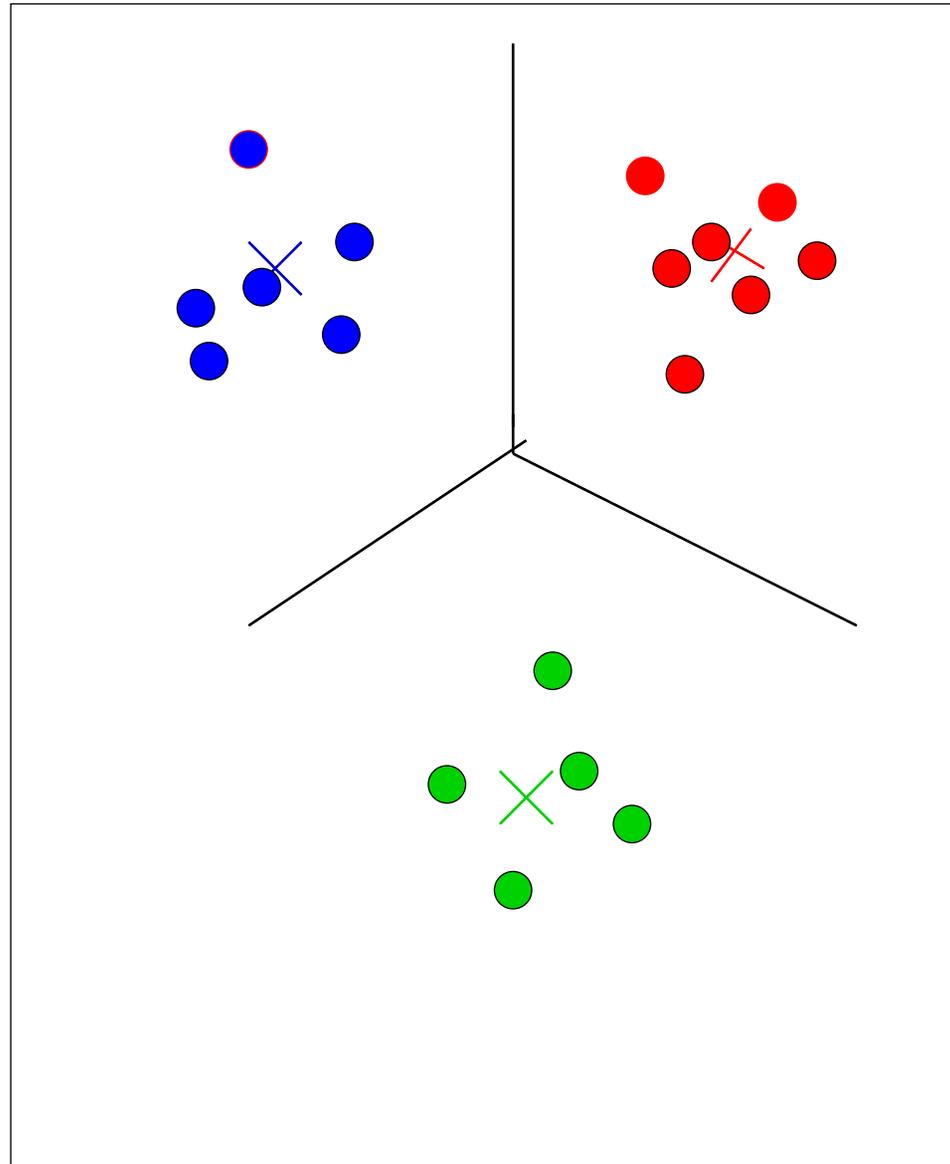
K-means



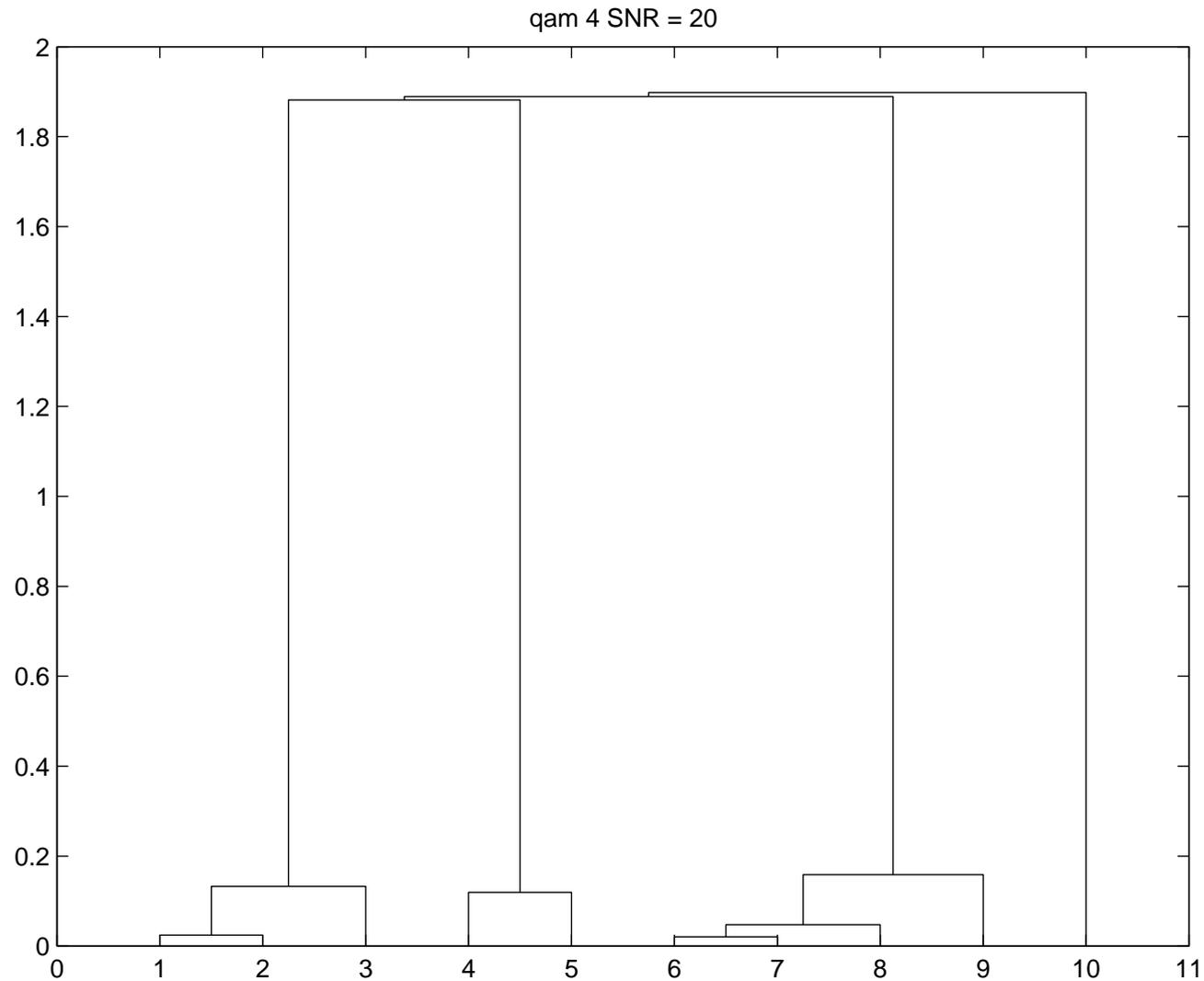
K-means



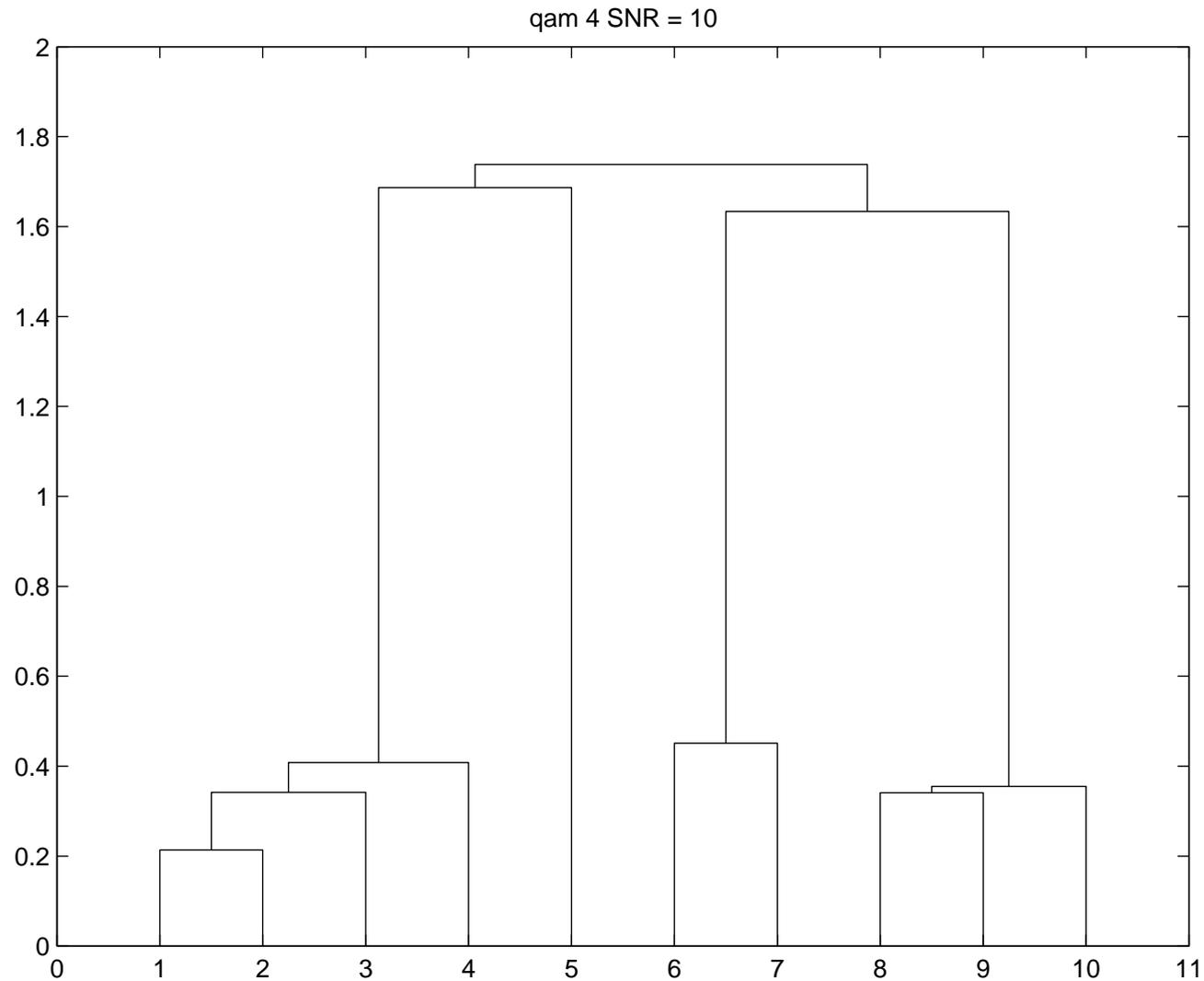
K-means



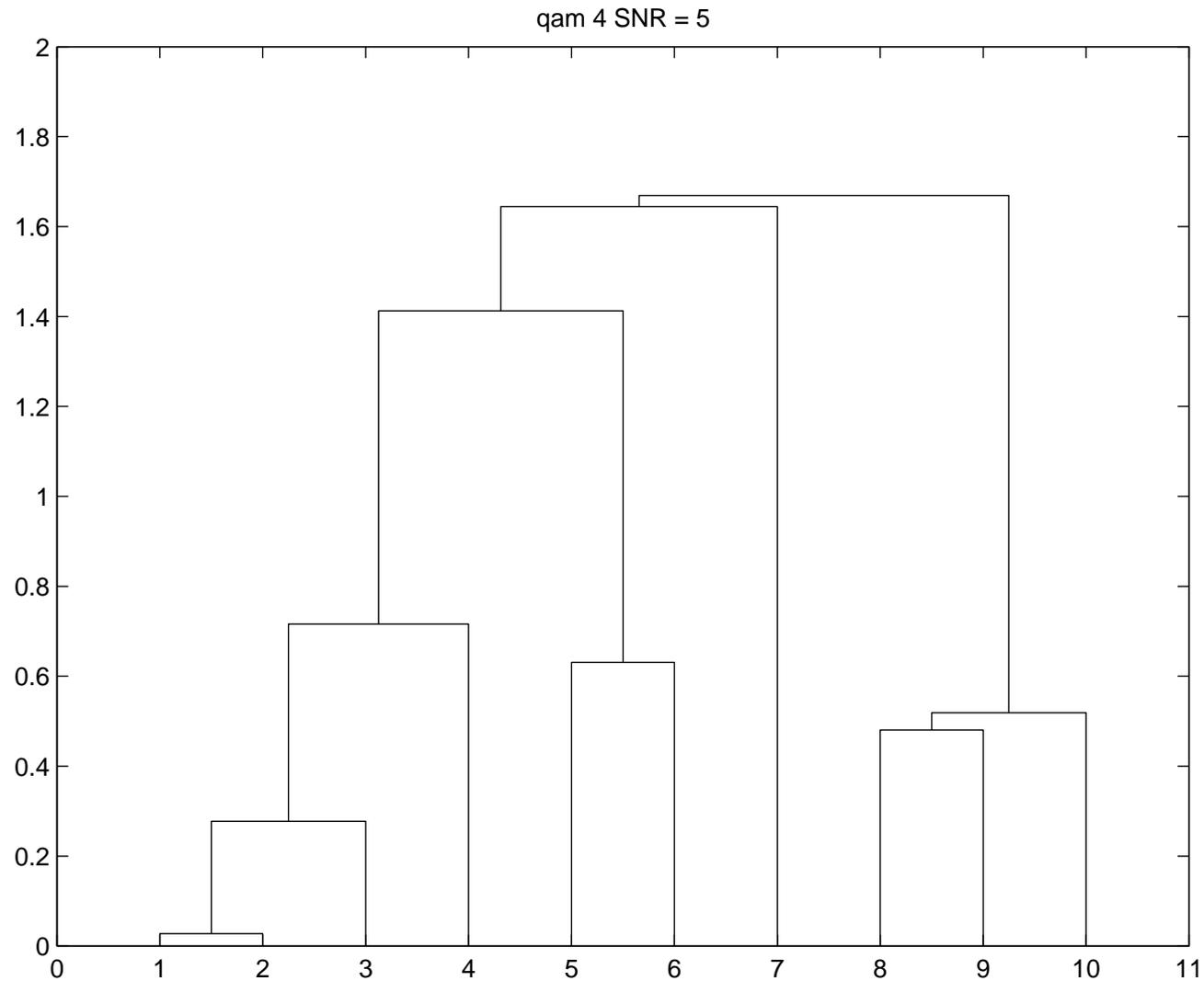
Classification hiérarchique ascendante



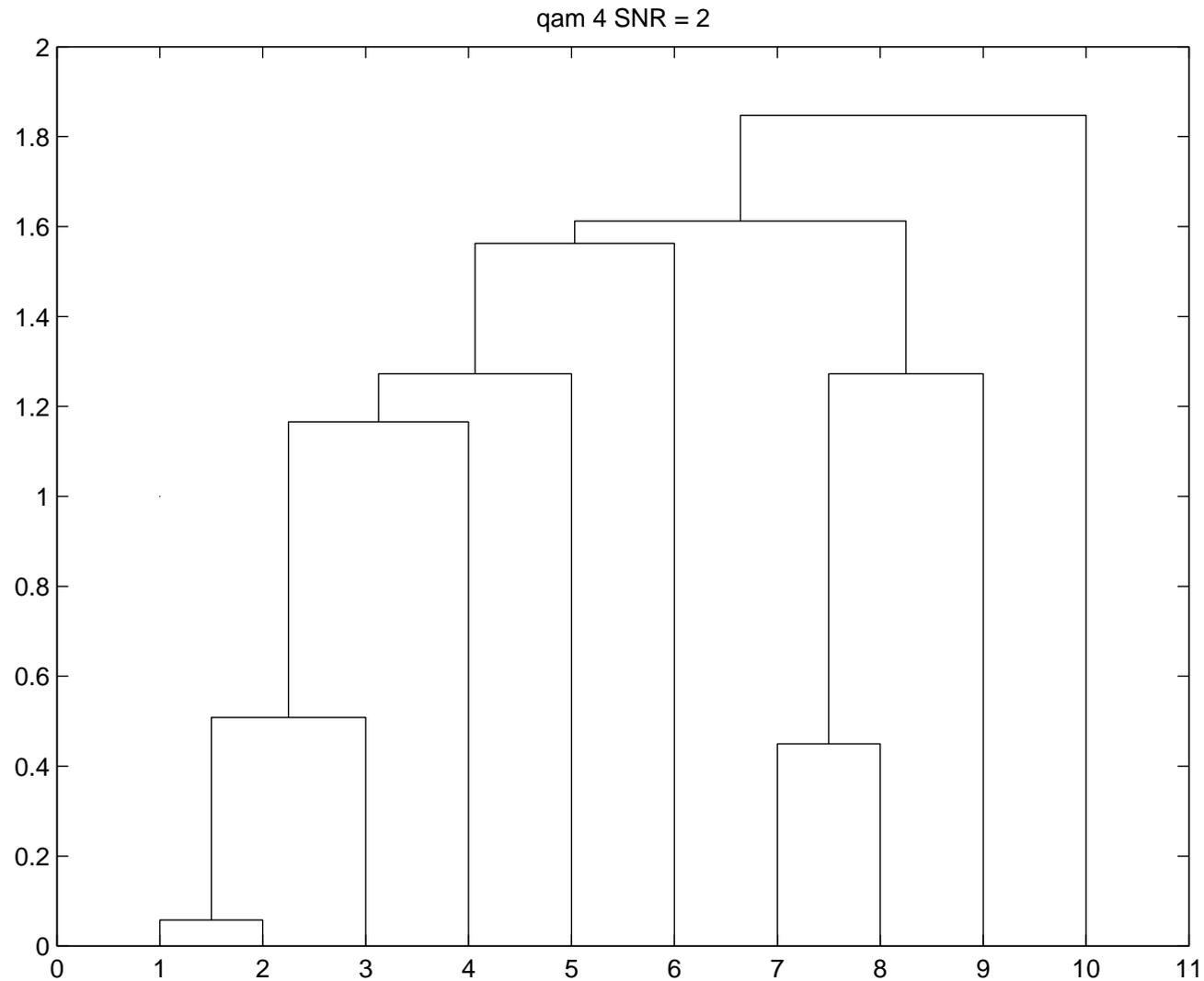
Classification hiérarchique ascendante



Classification hiérarchique ascendante



Classification hiérarchique ascendante



Fin

Merci pour votre attention !