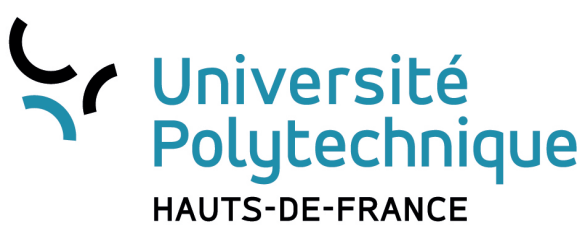


Apprentissage Statistique Automatique I

Apprentissage non supervisé

Andrés F. López-Lopera
Université Polytechnique Hauts-de-France (UPHF)

Thèmes

1. Introduction

2. *k*-means
   - Algorithme de Lloyd
   - Propriétés théoriques

3. Modèles de mélanges gaussiens
   - Mélange de gaussiens
   - Estimation des paramètres

Introduction

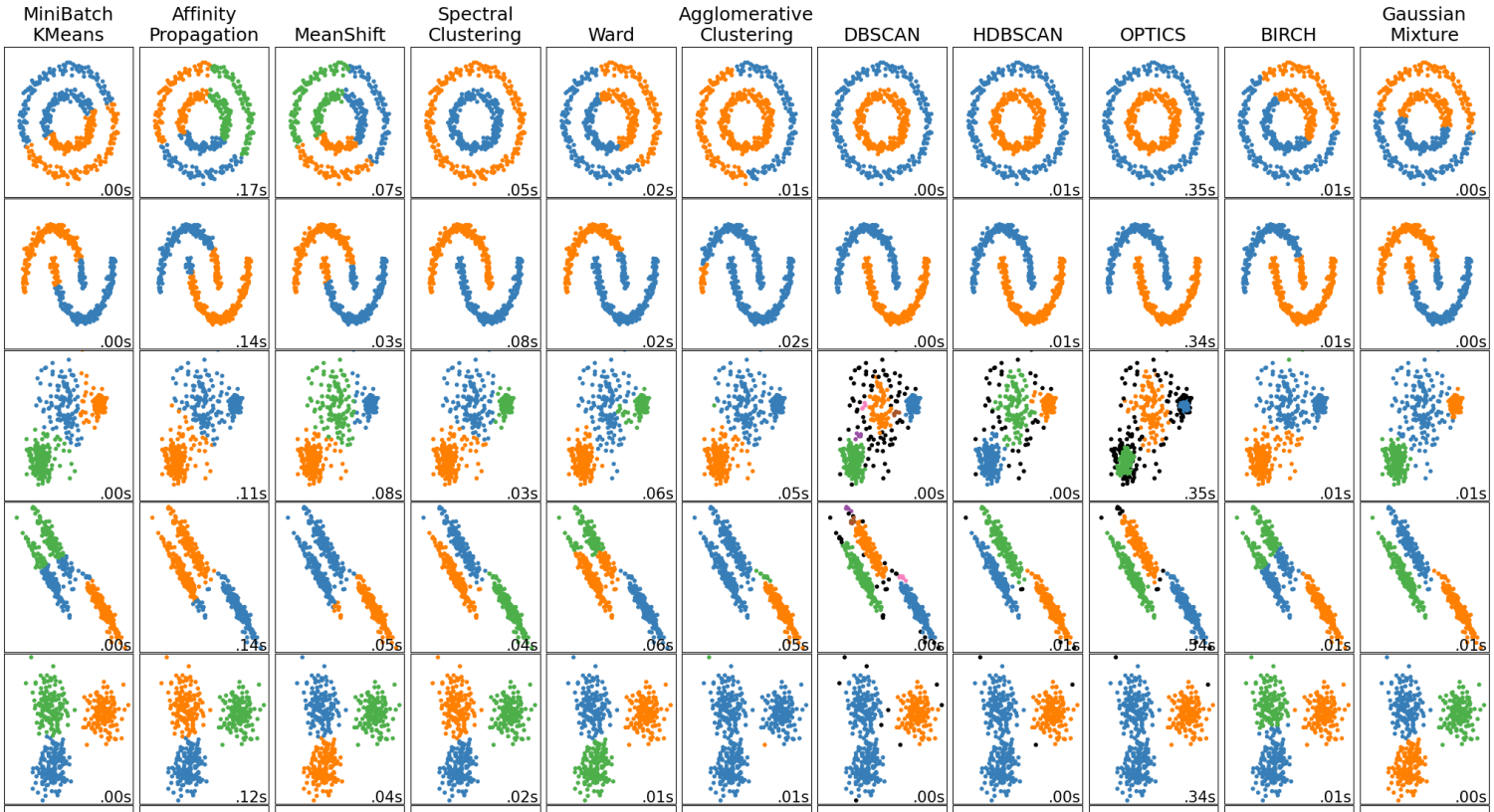
Introduction

Objectifs

- Structurer les données
- On cherche à regrouper les observations “proches” en classes

Vocabulaire

- Partitionner les données (*clustering*)
- Une méthode non-supervisé (sans étiquettes, *i.e.*, sans *y*)



Exemples d'applications

Gestion - Marketing

- Données : infos client, produits, ...
- But : segmenter la clientèle, définir des profils

Traitement Naturel du Langage (*NLP*)

- Données : texte, email, ...
- But : grouper automatiquement les textes proches

Sociologie

- Données : attributs d'un individu, e.g., revenus, sexe, ...
- But : former des catégories de population

Analyse génomique

- Données : gênes
- But : former des groupes homogènes de gênes

Notion de proximité

Questions

- Comment mesurer la proximité de deux observations ?
- Comment mesurer la proximité de deux classes ?

Ingrédients

- Fonction de dissimilarité : plus la mesure est faible, plus les objets sont similaires ($\approx$ à une distance)
- Fonction de similarité : plus la mesure est grande, plus les objets sont similaires

Distances usuelles entre deux observations

· Soient $x_1, x_2 \in \mathcal{X}$ deux observations

- Distance Euclidienne

$$d^2(x_1, x_2) = \sum_{i=1}^{d} (x_{1,i} - x_{2,i})^2$$

- Distance de Manhattan

$$d(x_1, x_2) = \sum_{i=1}^{d} |x_{1,i} - x_{2,i}|$$

- Distance de Minkowski

$$d(x_1, x_2) = \left(\sum_{i=1}^{d} |x_{1,i} - x_{2,i}|^p \right)^{\frac{1}{p}}$$

- Distance de Mahalanobis (pour une matrice symétrique W)

$$d^2(x_1, x_2) = \sum_{i=1}^{d} \sum_{j=1}^{d} W_{i,j}(x_{1,i} - x_{2,i})(x_{1,j} - x_{2,j})$$

Distances usuelles entre deux observations

- Distance de Hamming (pour le cas des variables discrètes)

$$d(x_1, x_2) = \sum_{i=1}^{d} \mathbb{1}_{x_{1,i} \neq x_{2,i}}$$

Exemple. Pour

$$x_1 = [0, 1, 2, 1, 2, 1, 0]^\top,$$
$$x_2 = [1, 0, 2, 1, 0, 1, 0]^\top,$$

on obtient

$$d(x_1, x_2) = 3$$

Distances entre deux classes $\mathcal{C}_0$ et $\mathcal{C}_1$

- Plus proche voisin

$$d(\mathcal{C}_0, \mathcal{C}_1) = \inf\{\text{dist}(x, y) : x \in \mathcal{C}_0, y \in \mathcal{C}_1\}$$

- Diamètre maximum

$$d(\mathcal{C}_0, \mathcal{C}_1) = \sup\{\text{dist}(x, y) : x \in \mathcal{C}_0, y \in \mathcal{C}_1\}$$

- Diamètre moyenne

$$d(\mathcal{C}_0, \mathcal{C}_1) = \frac{1}{\#\mathcal{C}_0 \#\mathcal{C}_1} \sum_{x \in \mathcal{C}_0, y \in \mathcal{C}_1} \text{dist}(x, y)$$

- Distance des barycentres

$$d(\mathcal{C}_0, \mathcal{C}_1) = \text{dist}(\mu_0, \mu_1)$$

- Distance de Ward

$$d(\mathcal{C}_0, \mathcal{C}_1) = \left(\frac{\#\mathcal{C}_0 \#\mathcal{C}_1}{\#\mathcal{C}_0 + \#\mathcal{C}_1} \right)^{\frac{1}{2}} \text{dist}(\mu_0, \mu_1)$$

k-means

k-means

· Supposons qu'on dispose d'un échantillon, supposé i.i.d., de taille n : $x_1, \ldots, x_n$ à valeurs dans $\mathbb{R}^d$

· L'algorithme k-means se base principalement sur deux étapes

- Définir $K \leq n$ groupes où chaque groupe k est représenté par un centroïde $\mu_k \in \mathbb{R}^d$
- Affecter chaque donnée au centroïde le plus proche

$$\widehat{y}_i = \min_{k \in [\![0, K-1]\!]} d(x_i, \mu_k)$$

· Pour déterminer les K centroïdes $\mu_0, \ldots, \mu_{K-1}$, il est possible de minimiser un critère de distorsion :

$$\mathcal{E}_n(\mu_0, \ldots, \mu_{K-1}) = \frac{1}{n} \sum_{i=1}^{n} \min_{k \in [\![0, K-1]\!]} d^2(x_i, \mu_k)$$

Algorithme de Lloyd : optimisation alternée

Algorithm Algorithme de Lloyd pour minimiser $\mathcal{E}_n(\mu_0, \ldots, \mu_{K-1})$

Données d'entré : $x_1, \ldots, x_n$, $K \leq n$

1: Initialiser les K centroïdes $\mu_0, \ldots, \mu_{K-1}$ (*e.g.*, au hasard, ou via k-means++ [Arthur and Vassilvitskii, 2007])

2: **while** convergence **do**

3: Affecter chaque observation au centre le plus proche :

$$x_i \in \mathcal{C}_{k^\star}, \quad \text{où} \quad k^\star = \underset{k \in [\![0, K-1]\!]}{\operatorname{argmin}} \; d(x_i, \mu_k) \tag{1}$$

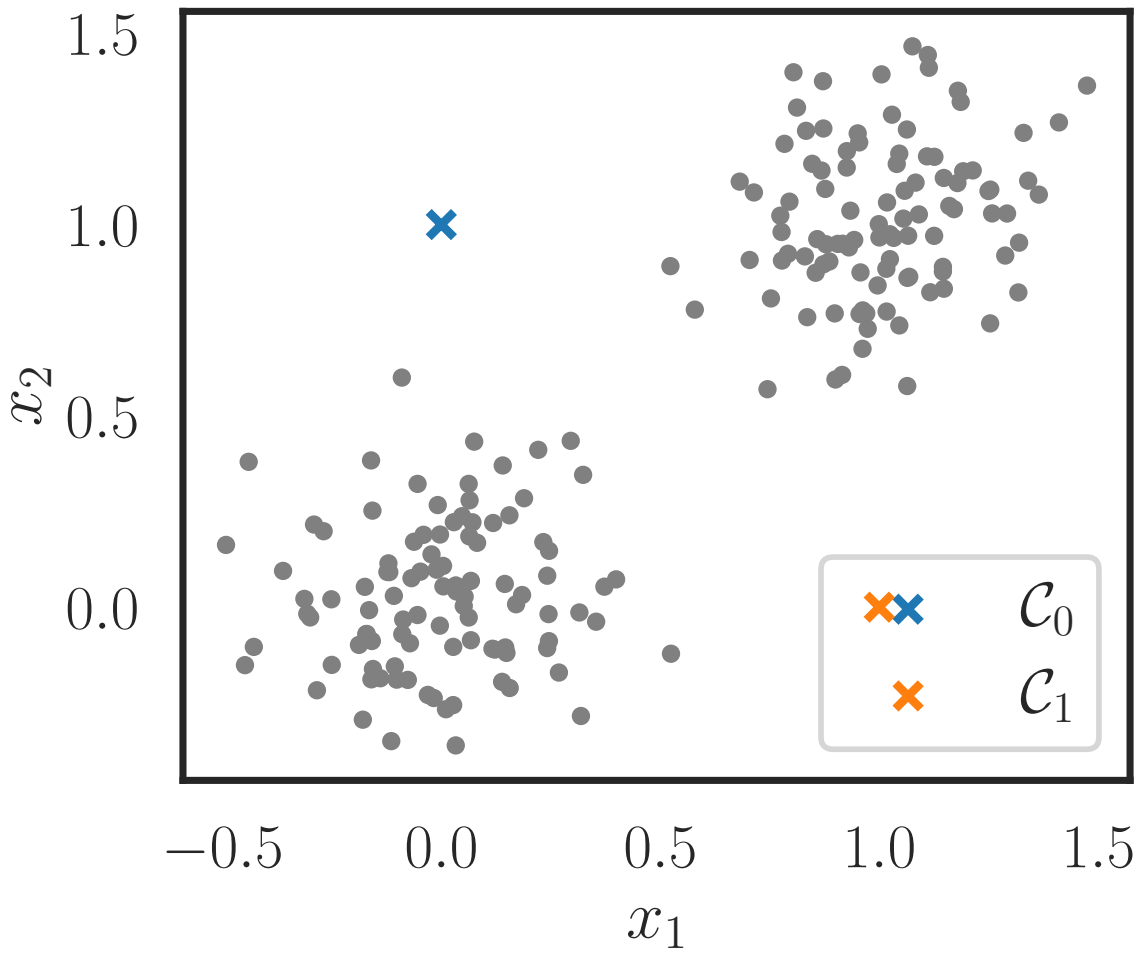
4: Mettre à jour les centroïdes, en calculant la moyenne empirique des observations dans chaque classe :

$$\mu_k = \frac{1}{\#\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} x_i \tag{2}$$

Rem :

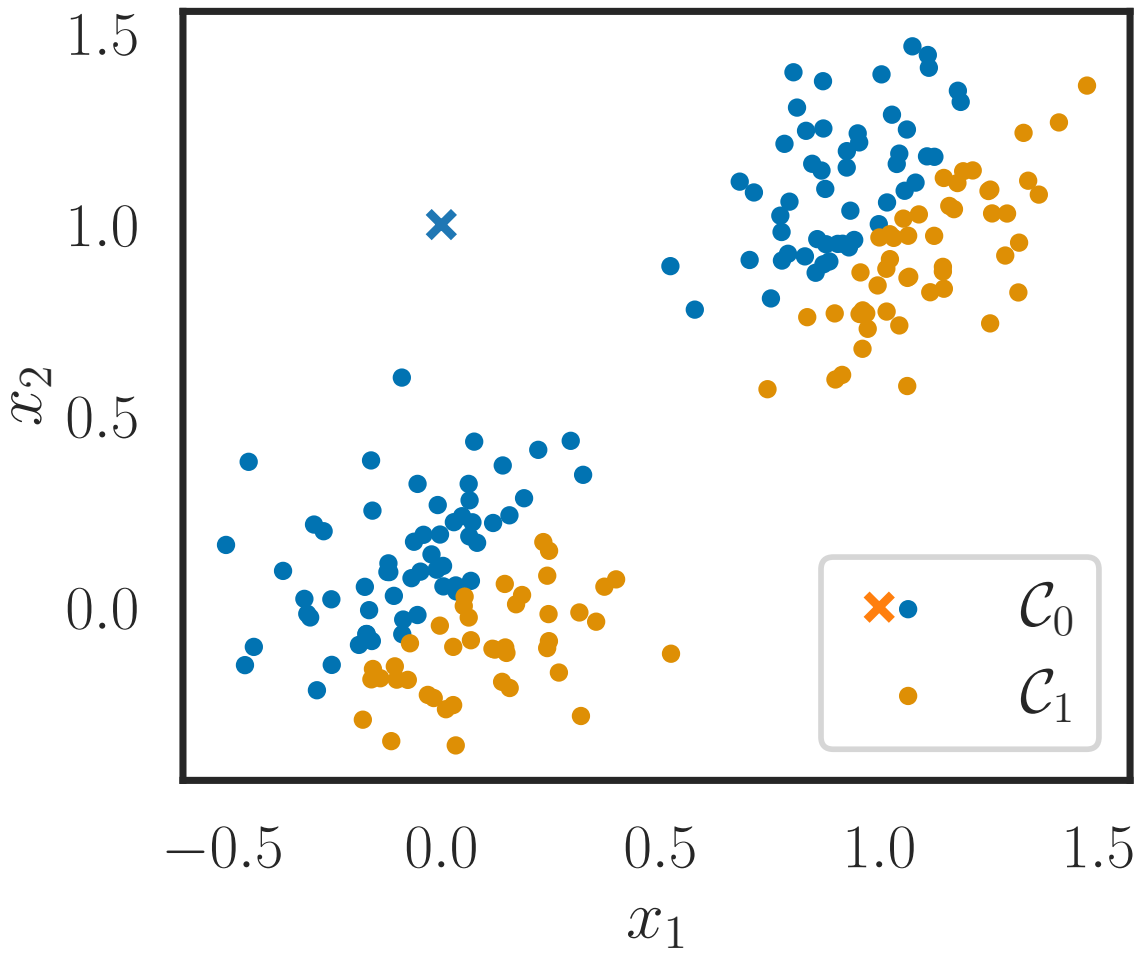
- Convergence vers un minimum local seulement
- Solution possible : adopter une procédure "multi-initialisation" (*multi-start*) et choisir le meilleur minimum local

Exemple avec $d = 2$ et $K = 2$



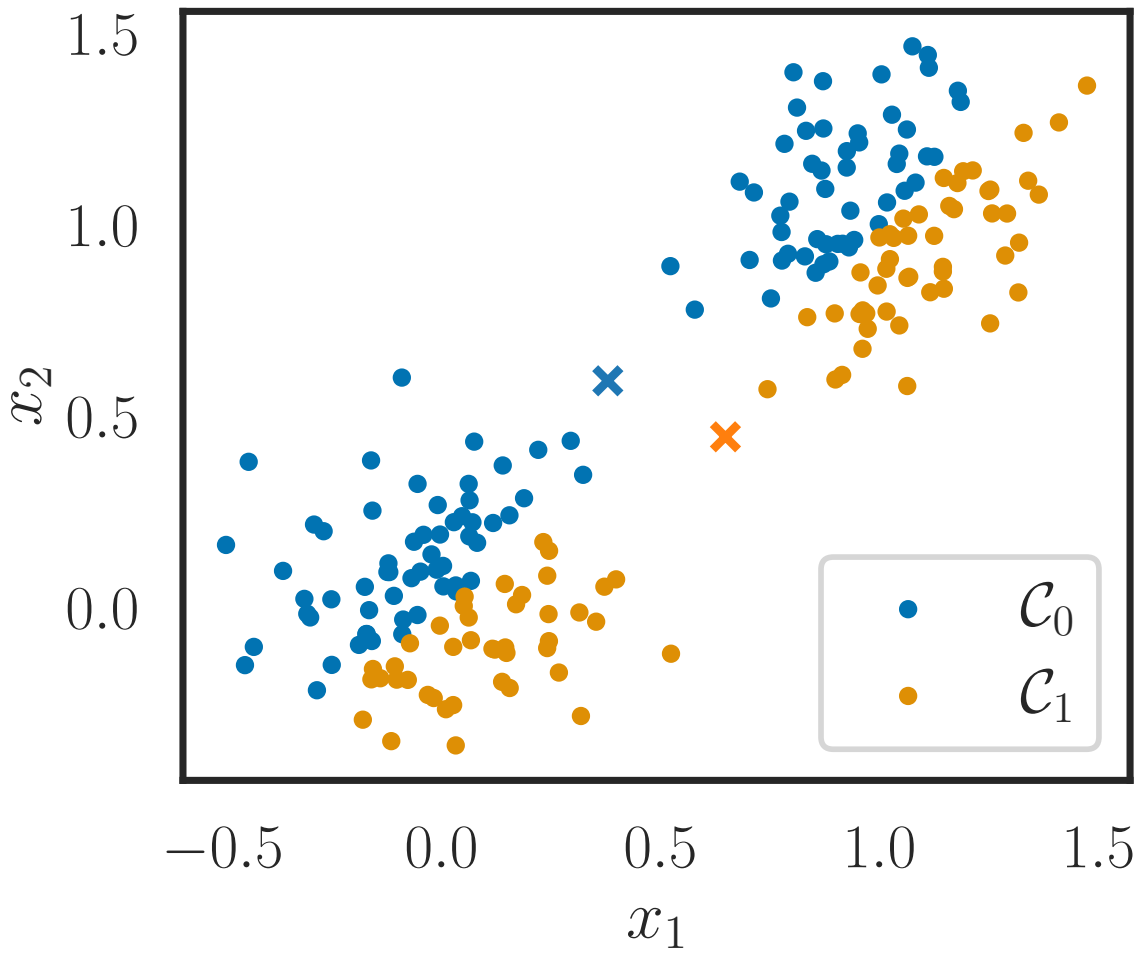
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



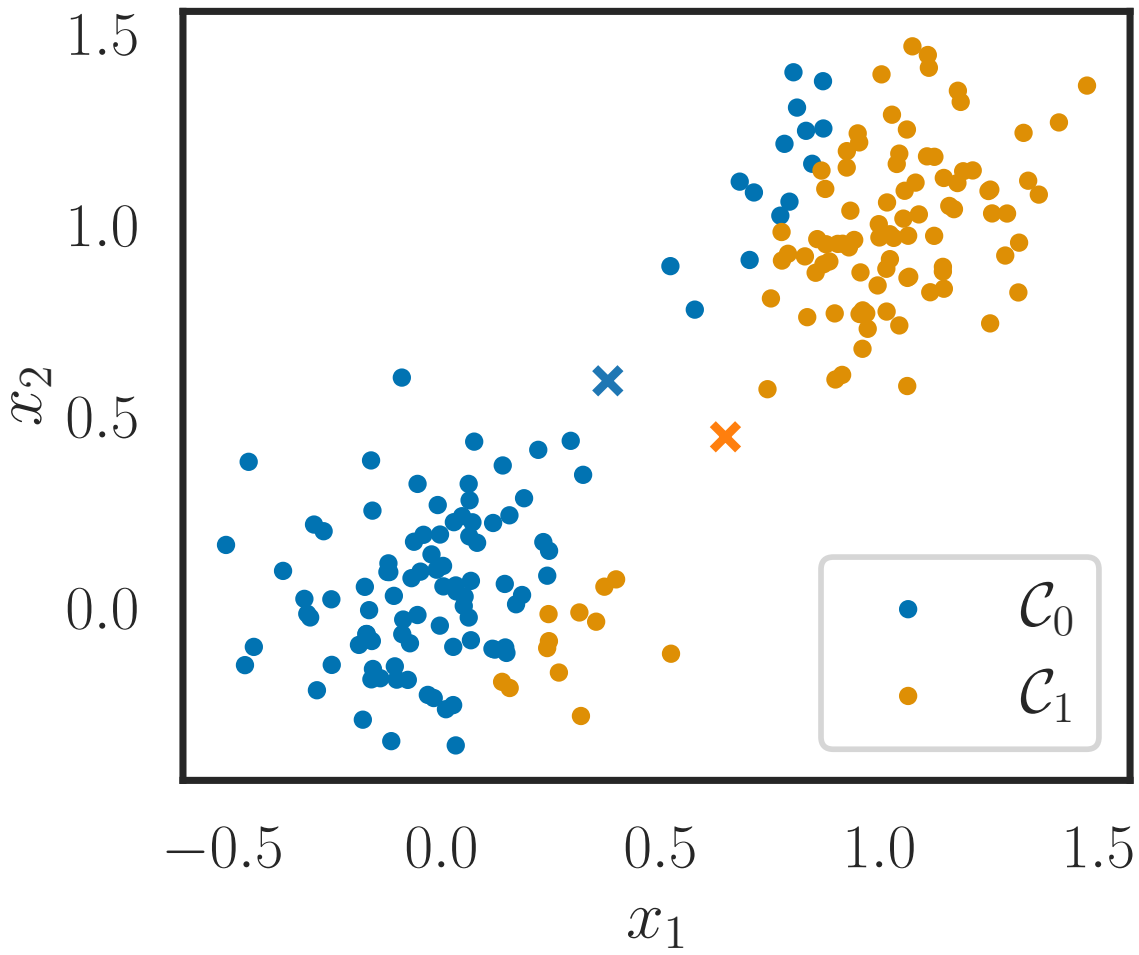
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



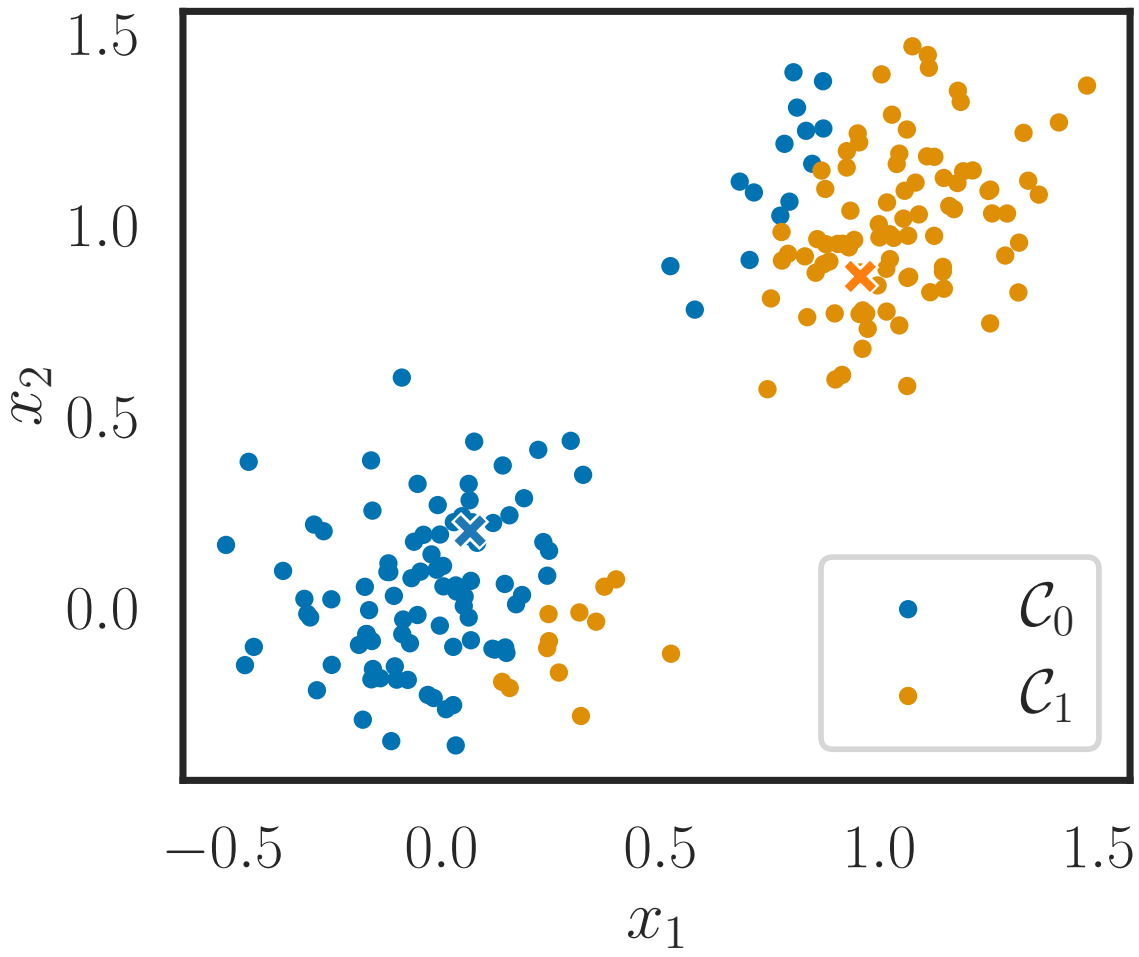
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



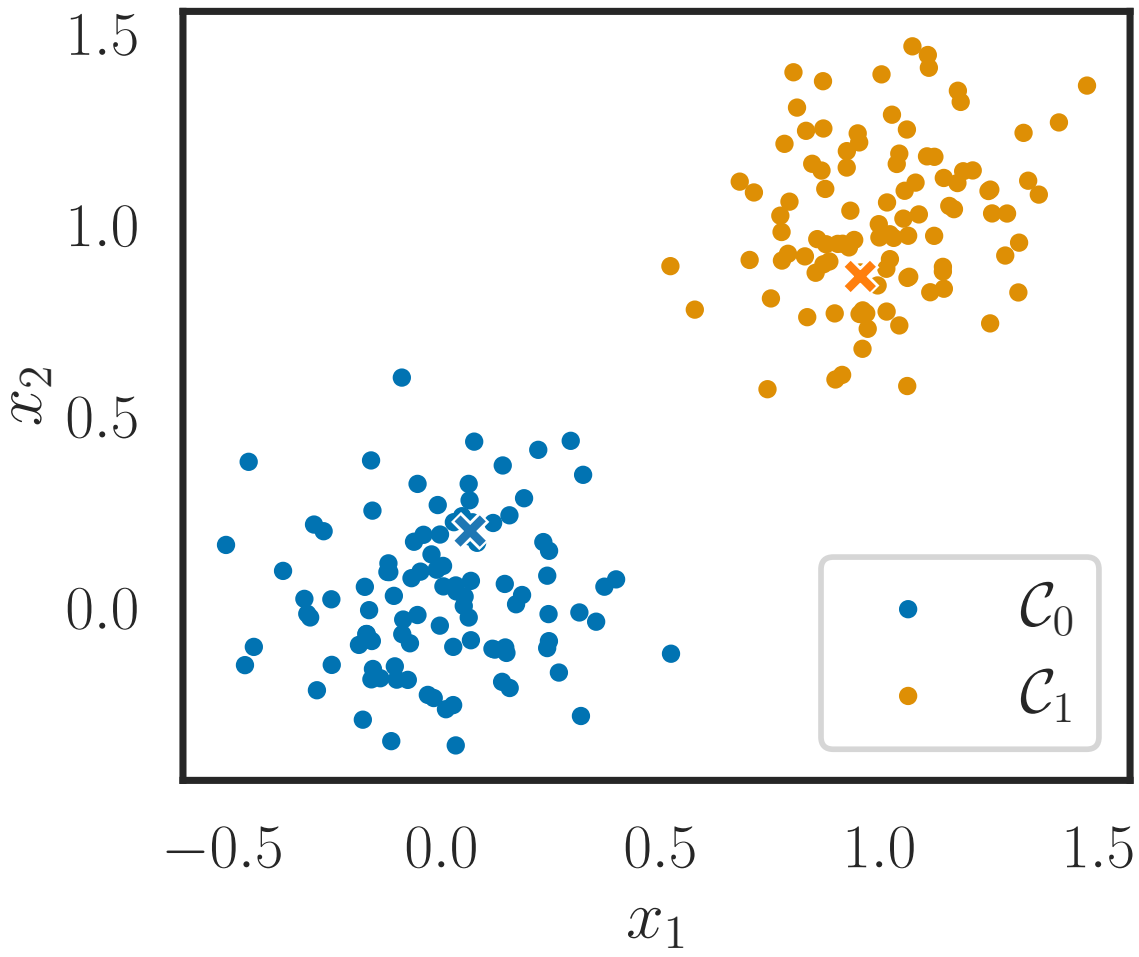
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



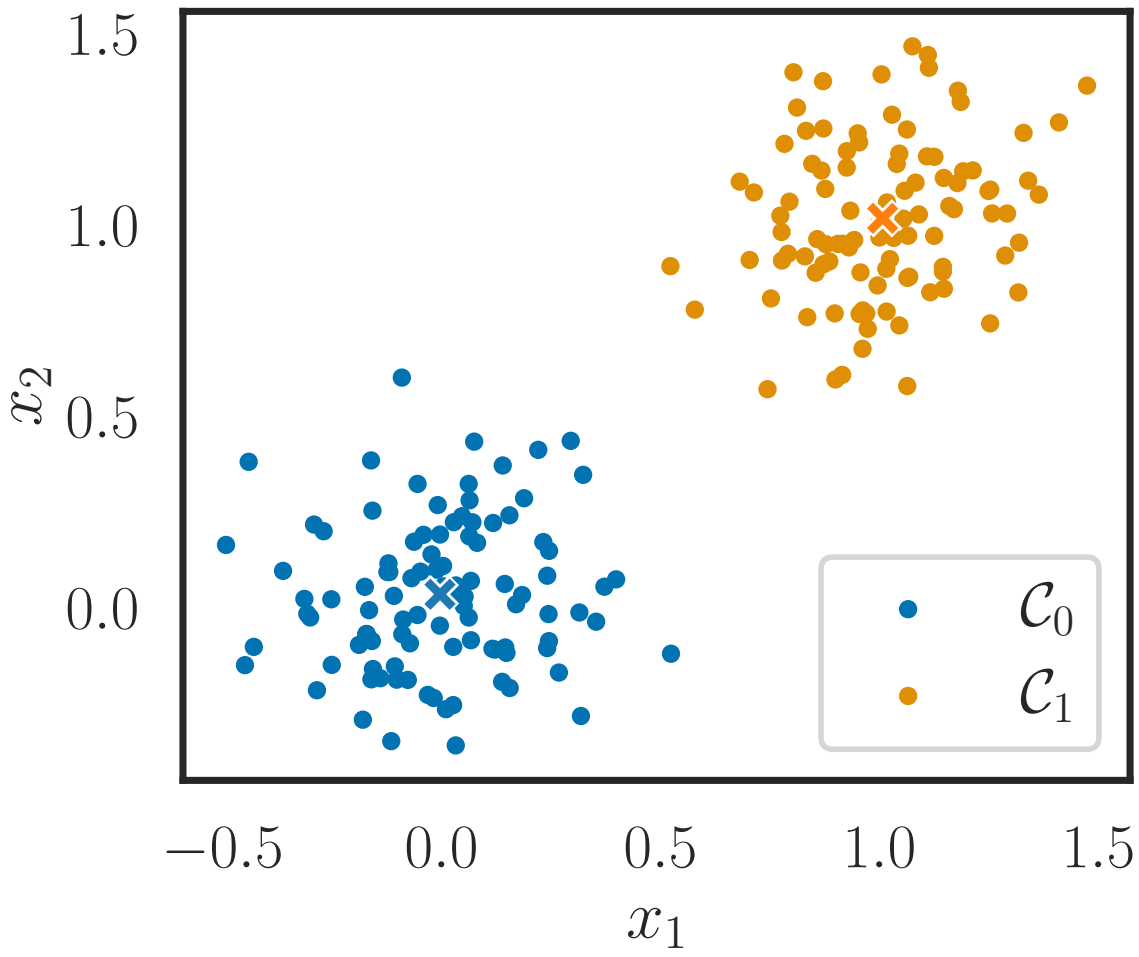
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



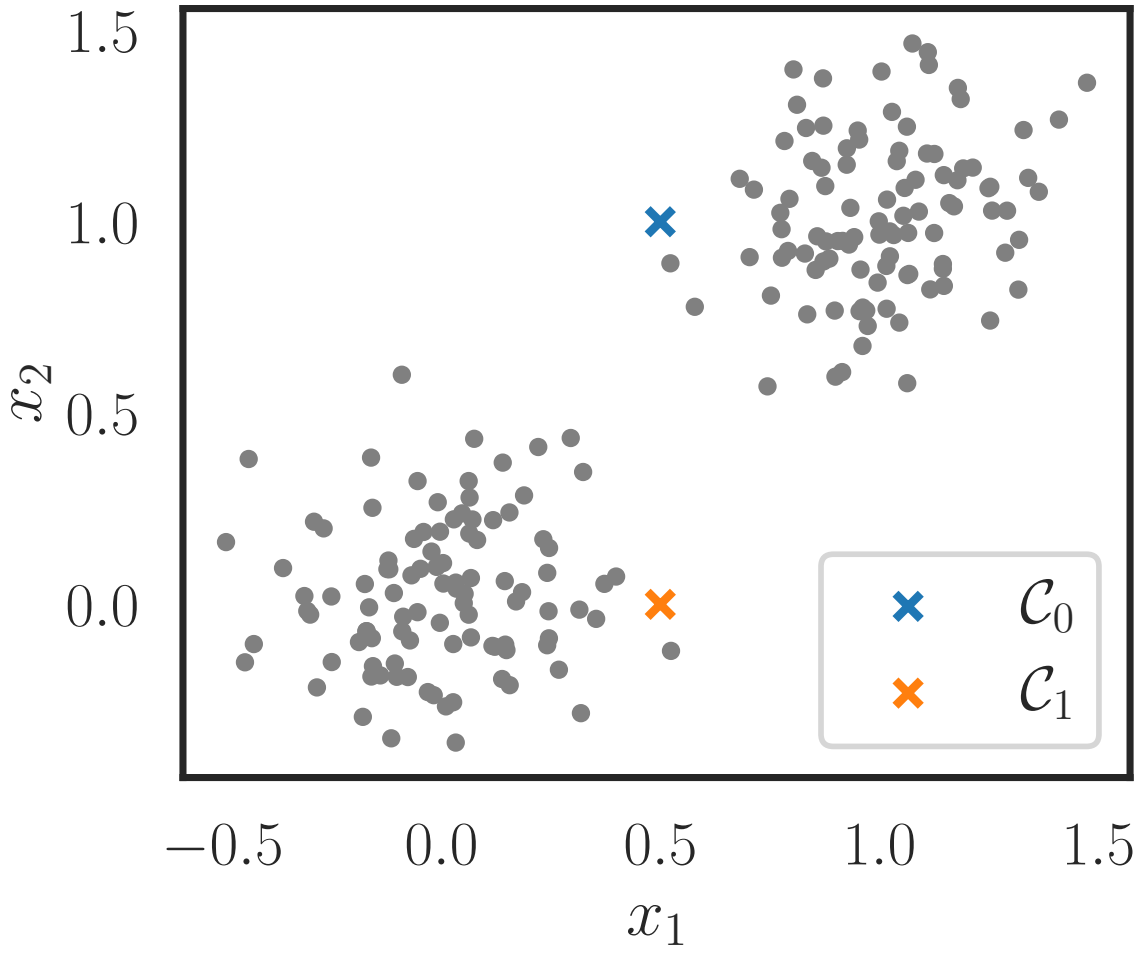
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



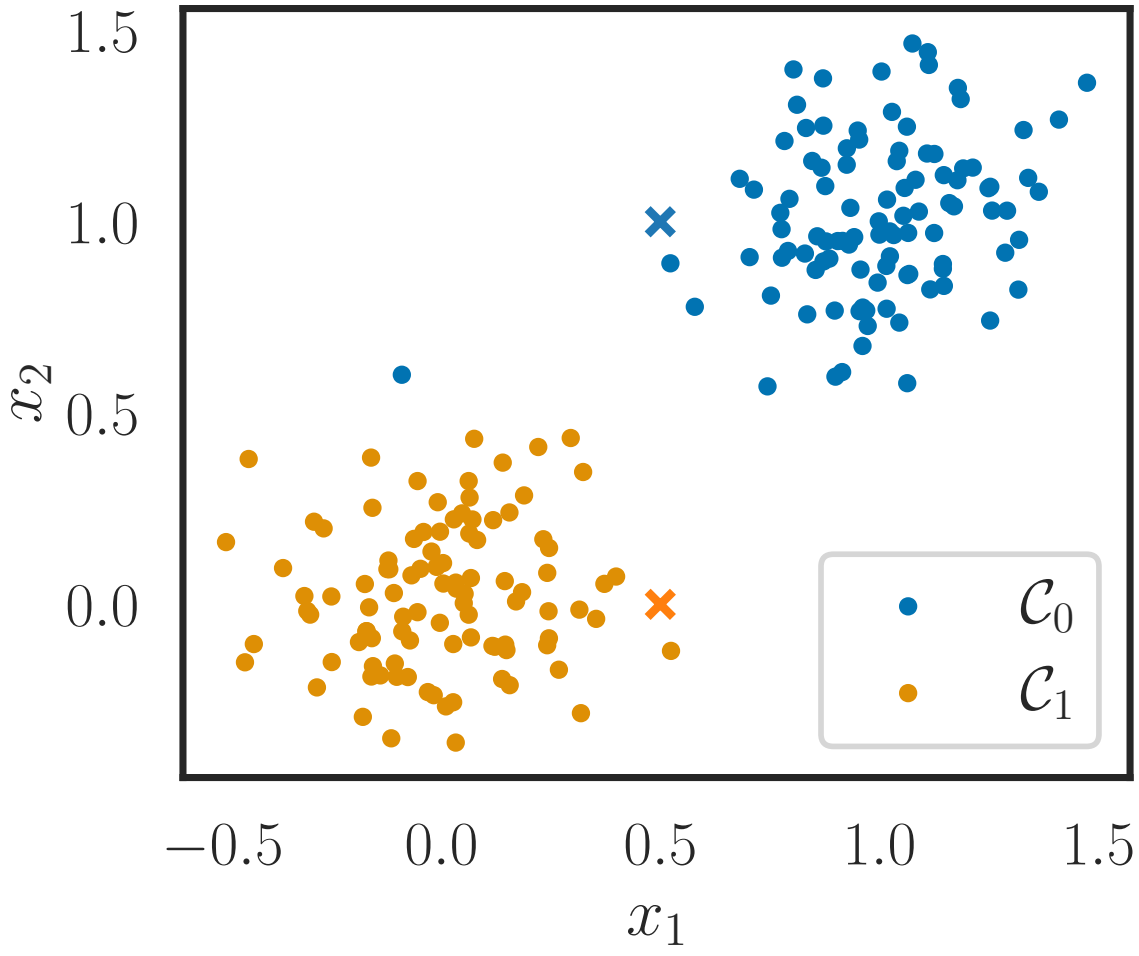
· Bonne performance pour des données “séparable linéairement”

Exemple avec $d = 2$ et $K = 2$



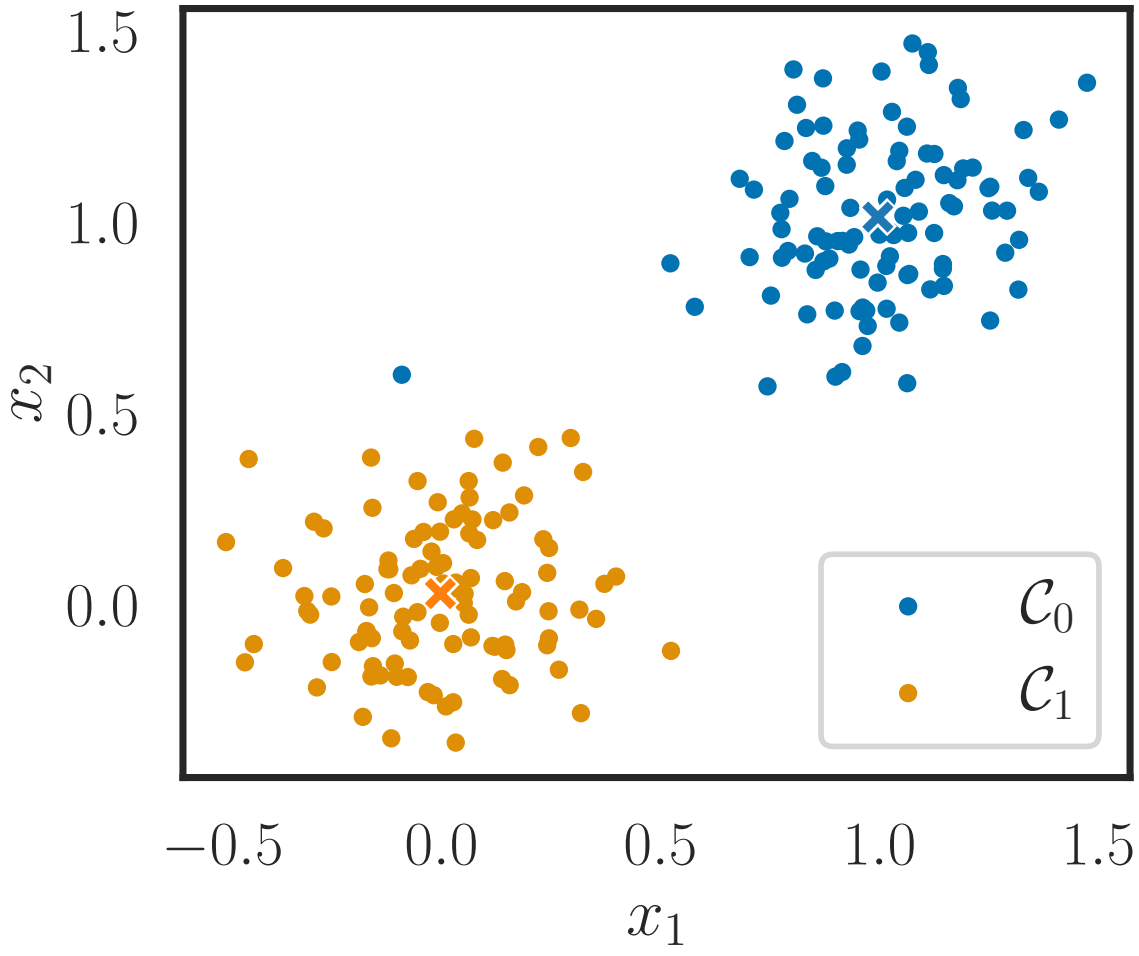
- Problème d'identification...

Exemple avec $d = 2$ et $K = 2$



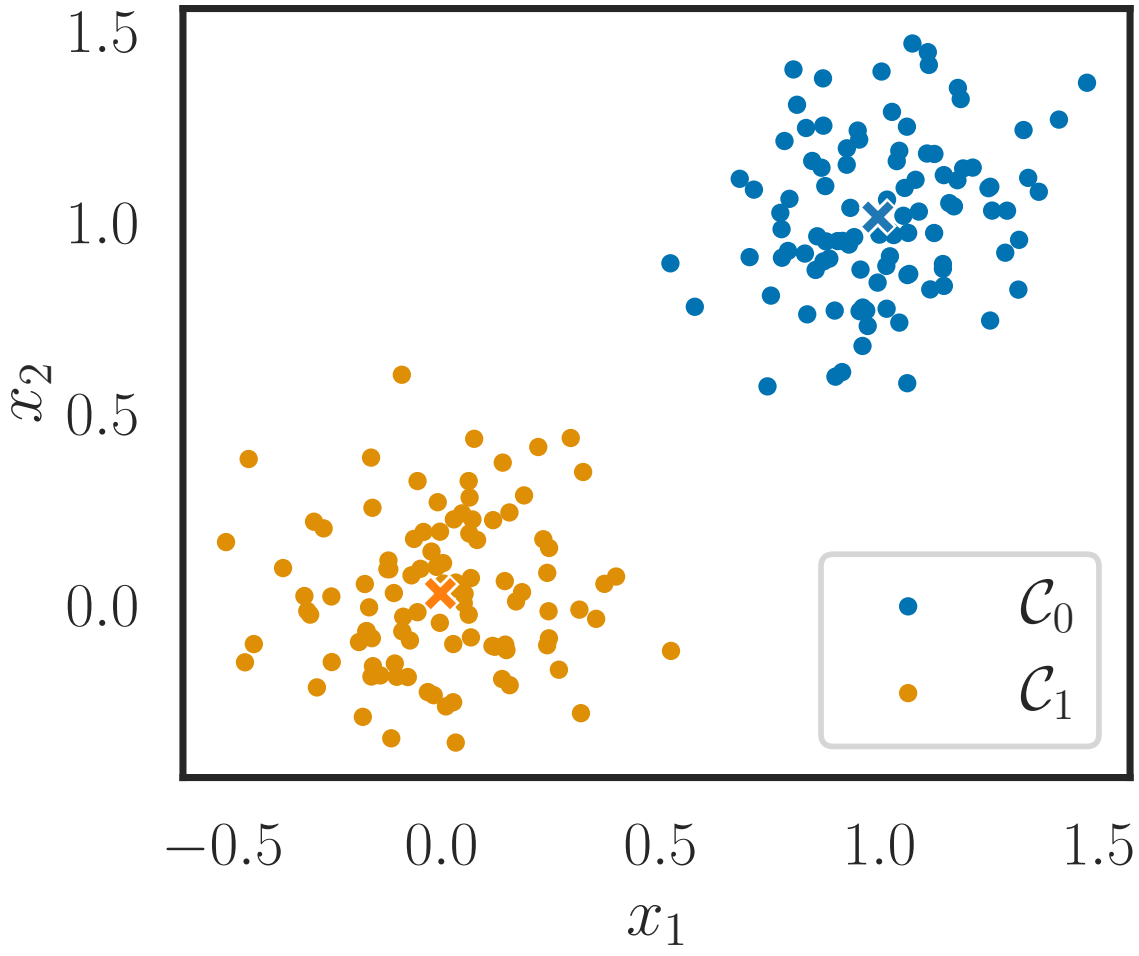
- Problème d'identification...

Exemple avec $d = 2$ et $K = 2$



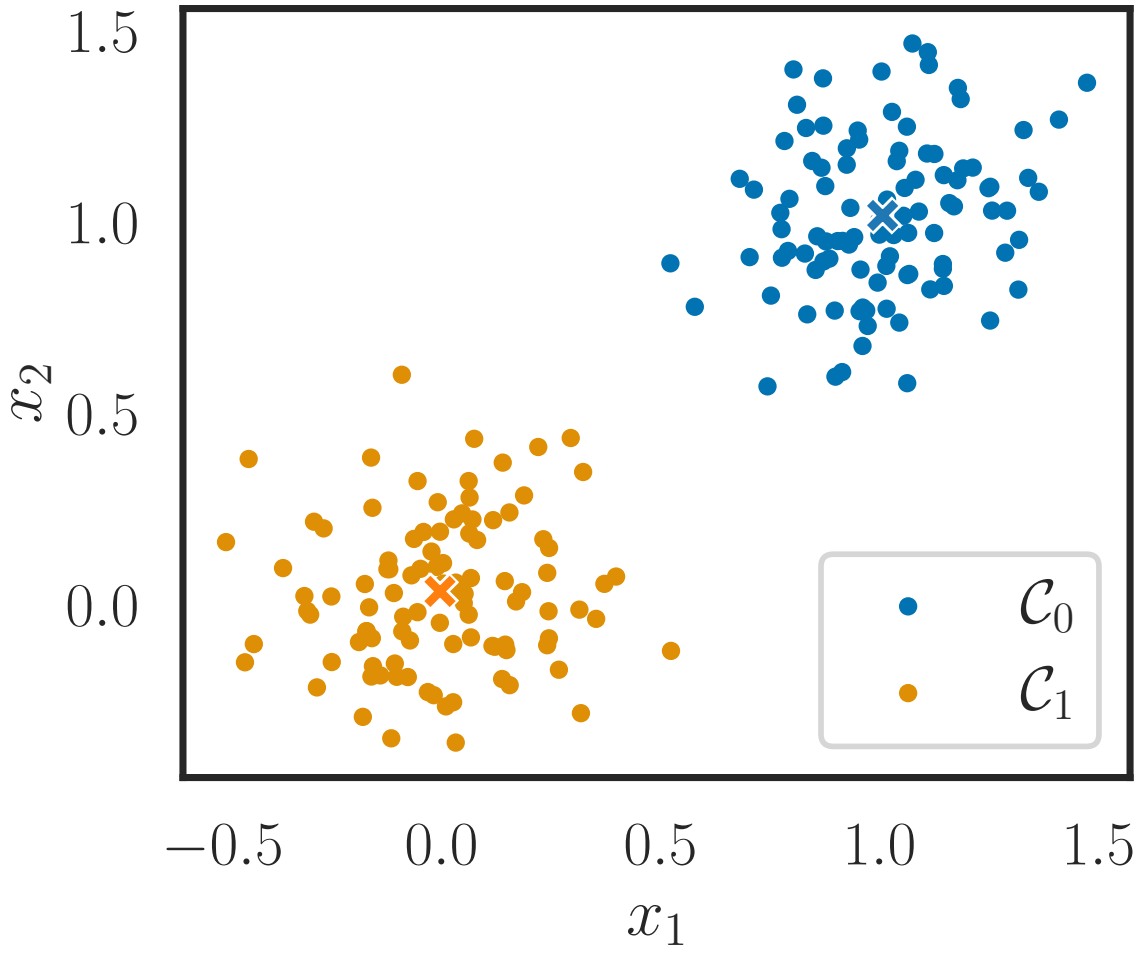
- Problème d'identification...

Exemple avec $d = 2$ et $K = 2$



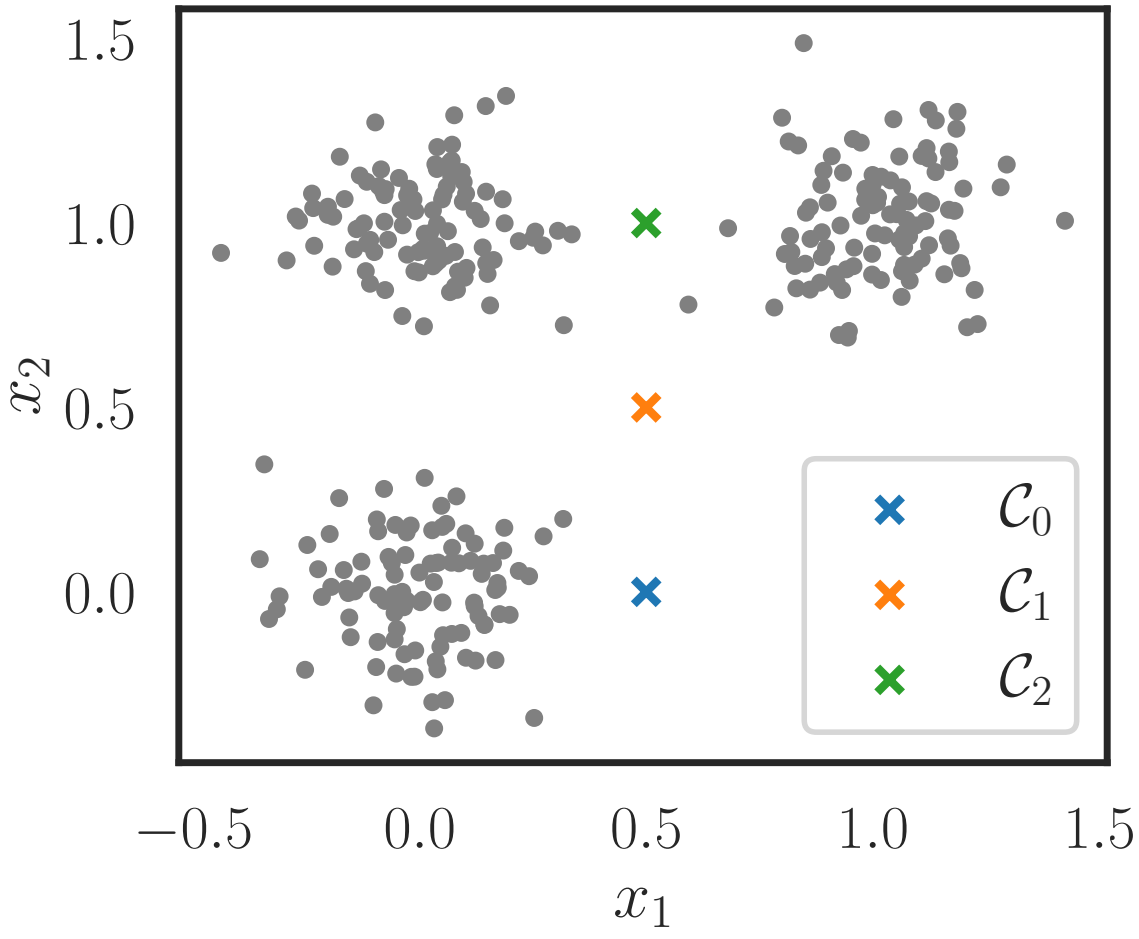
- Problème d'identification...

Exemple avec $d = 2$ et $K = 2$



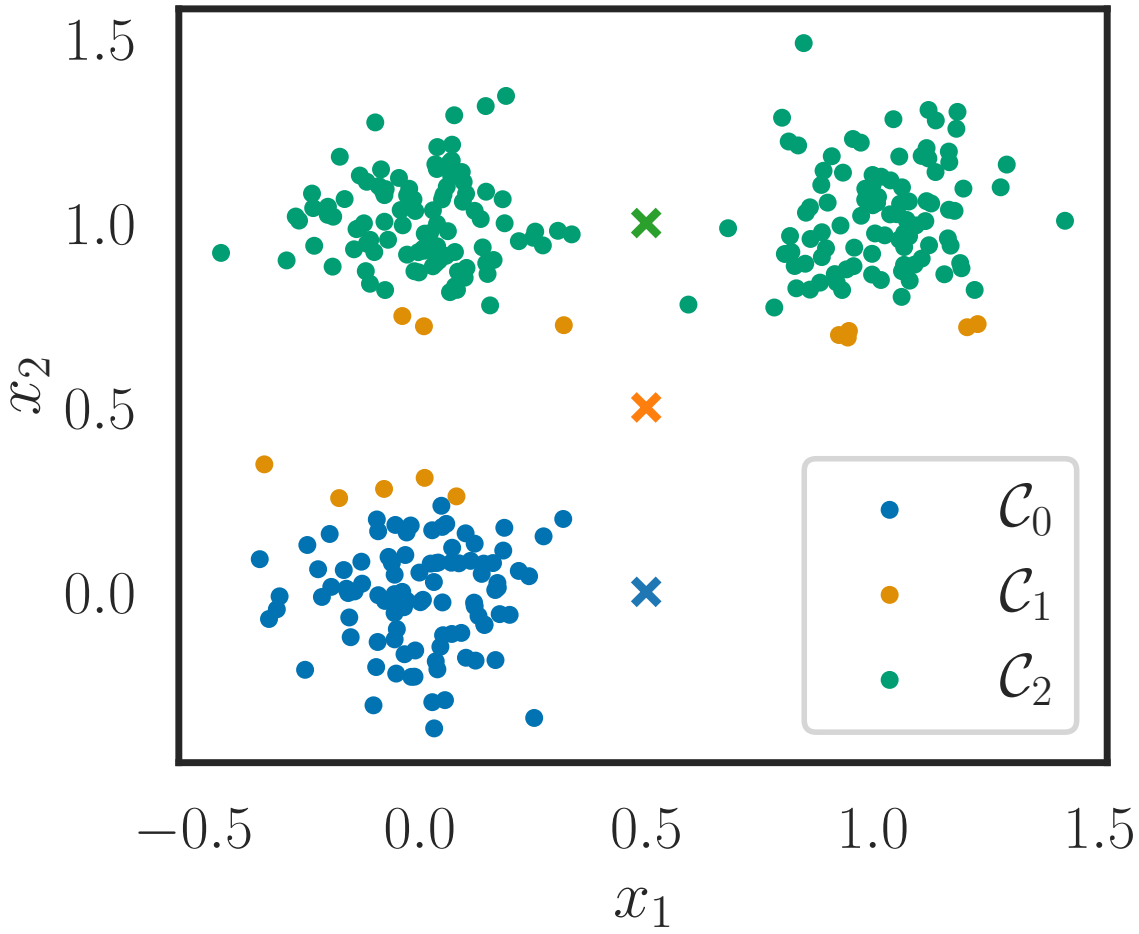
- Problème d'identification...

Exemple avec $d = 2$ et $K = 2$



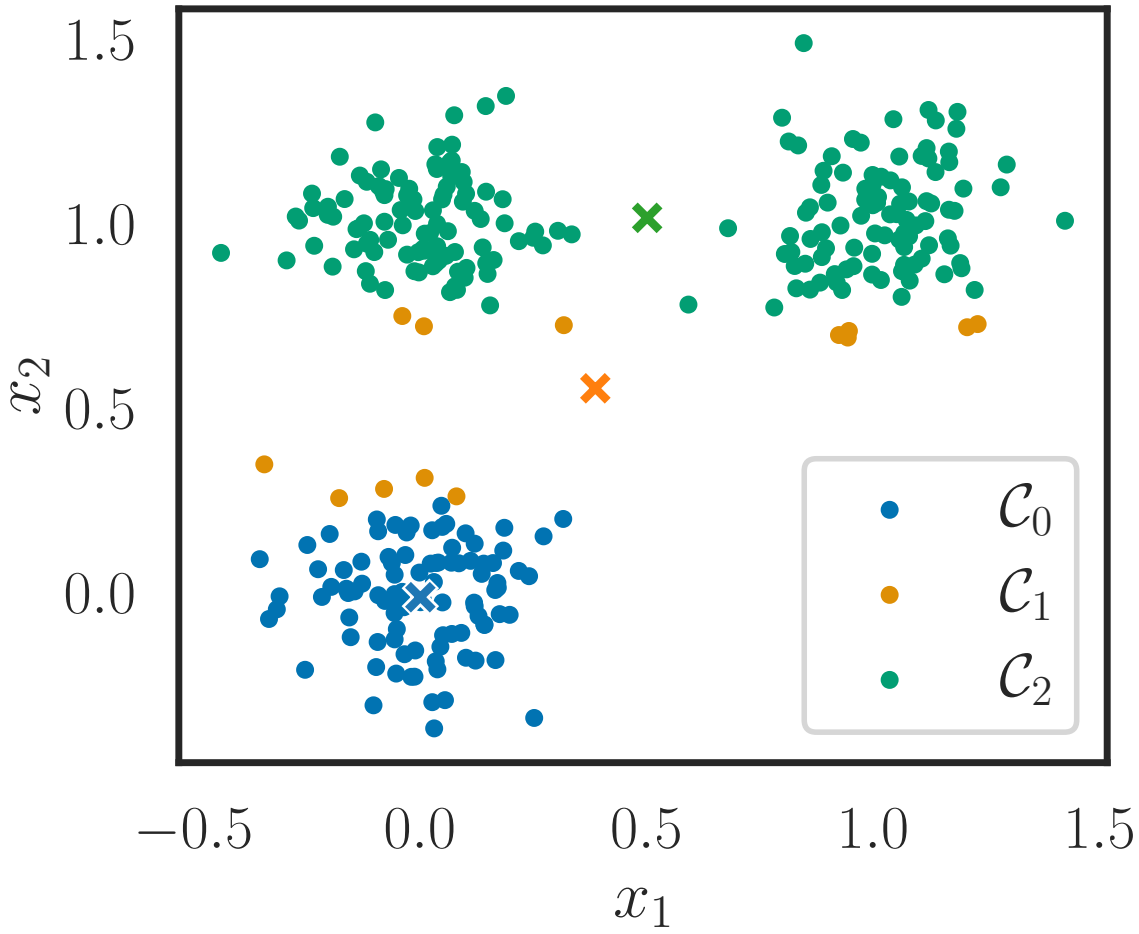
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



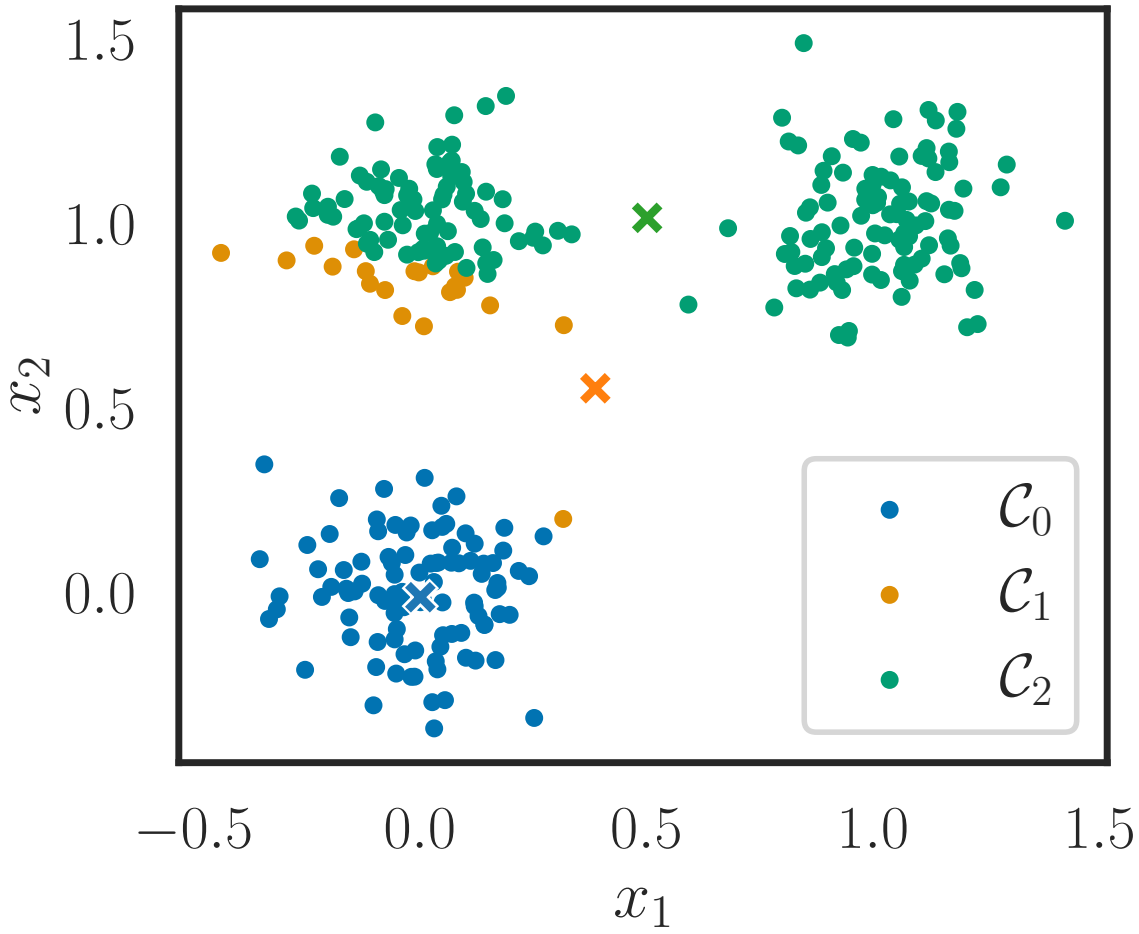
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



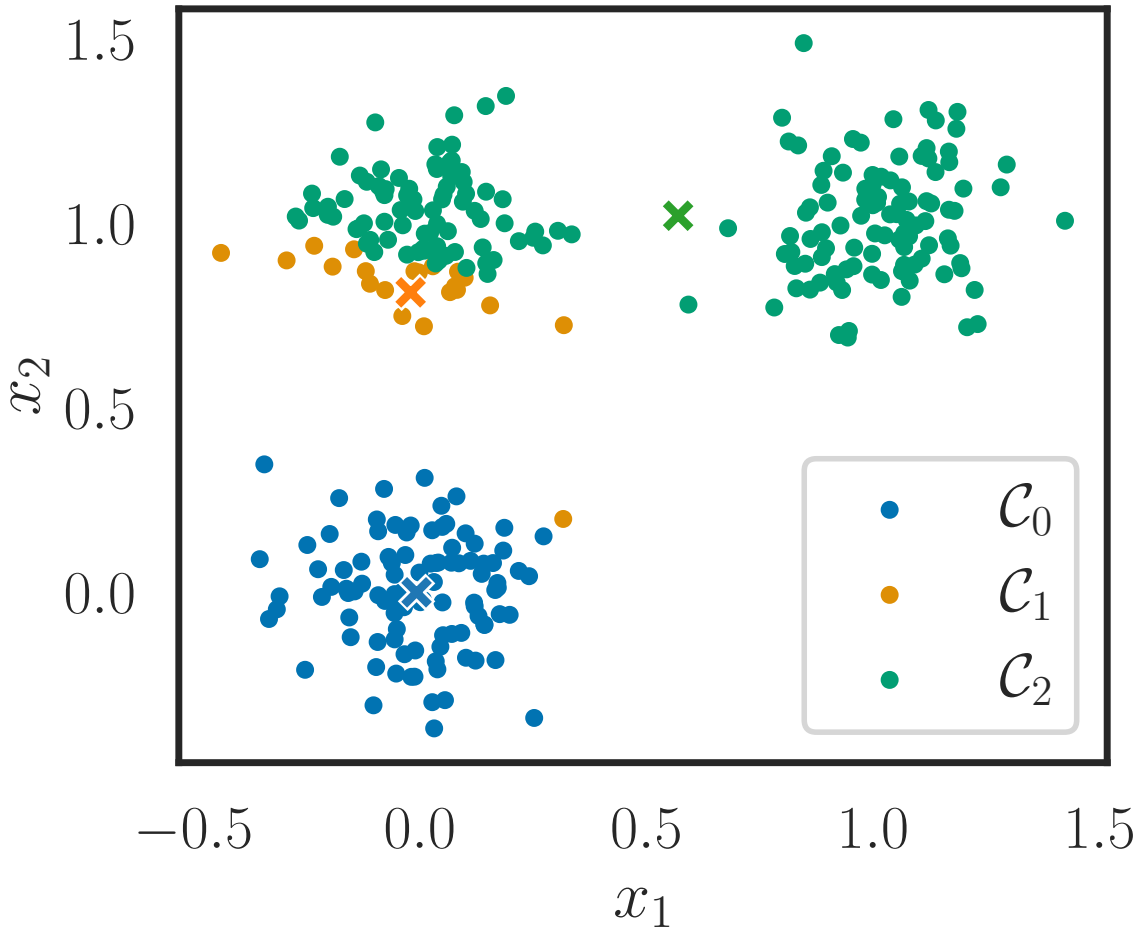
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



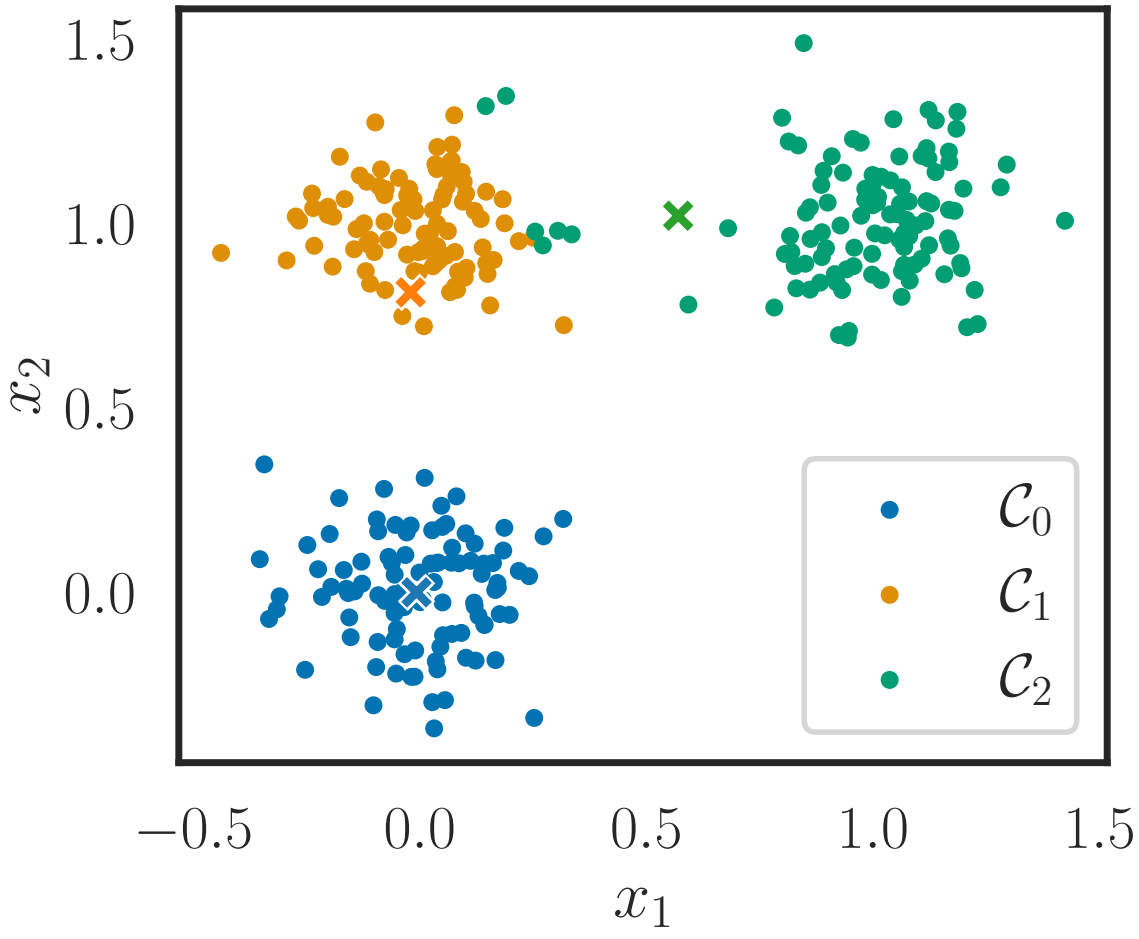
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



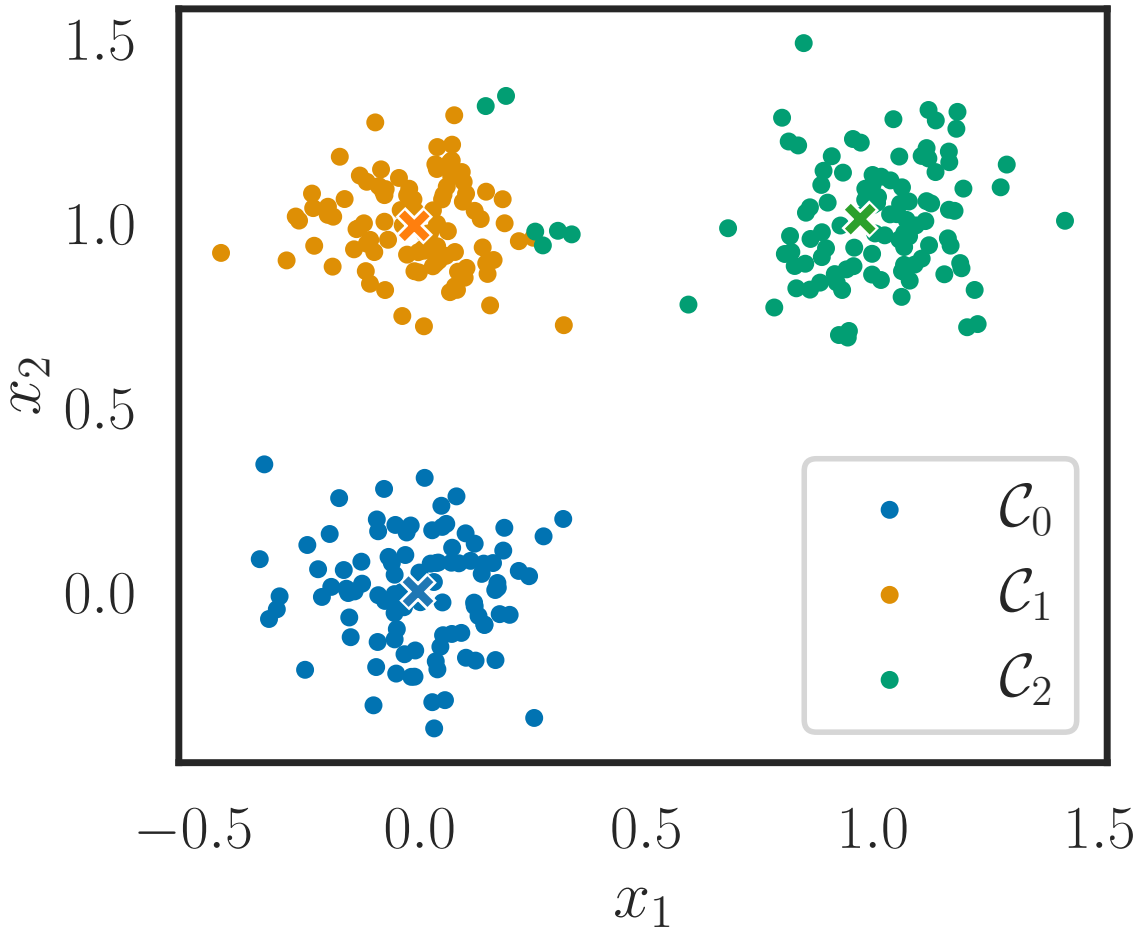
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



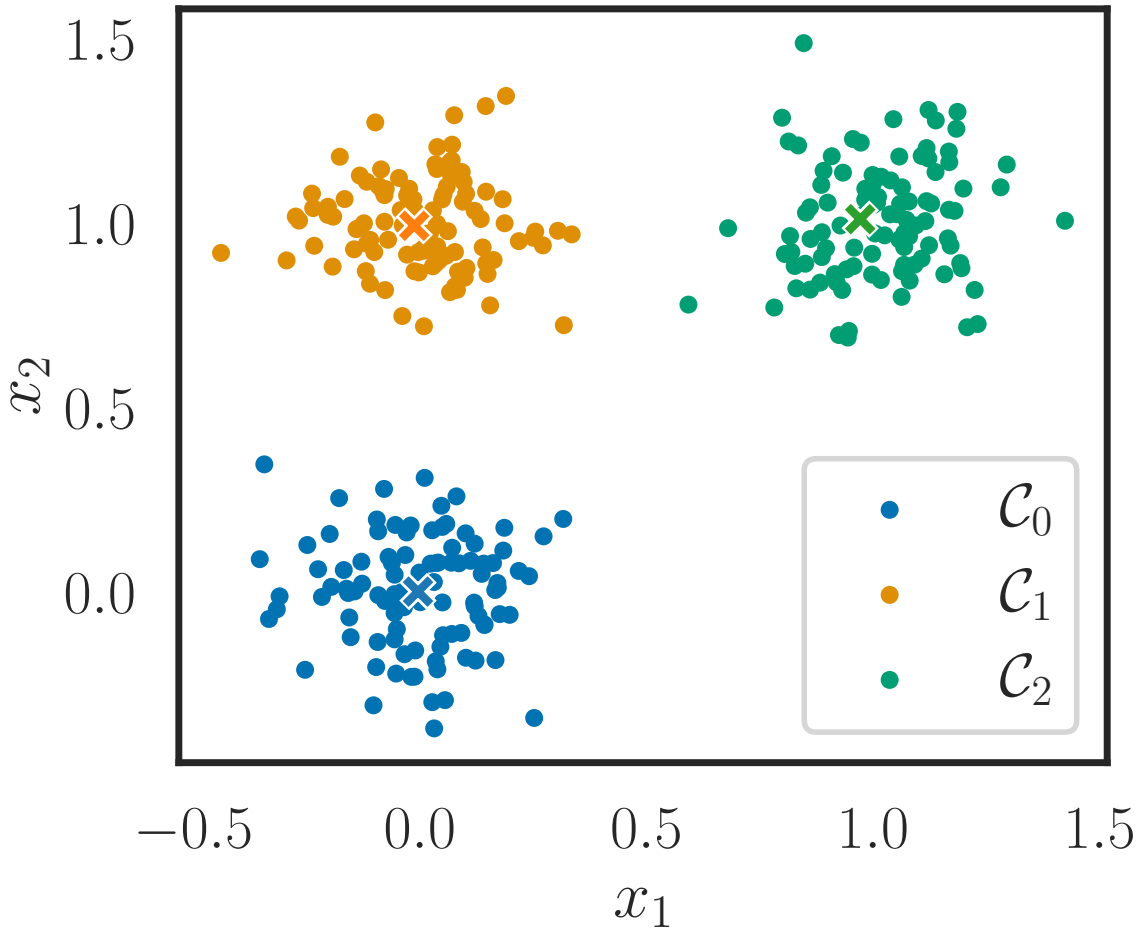
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



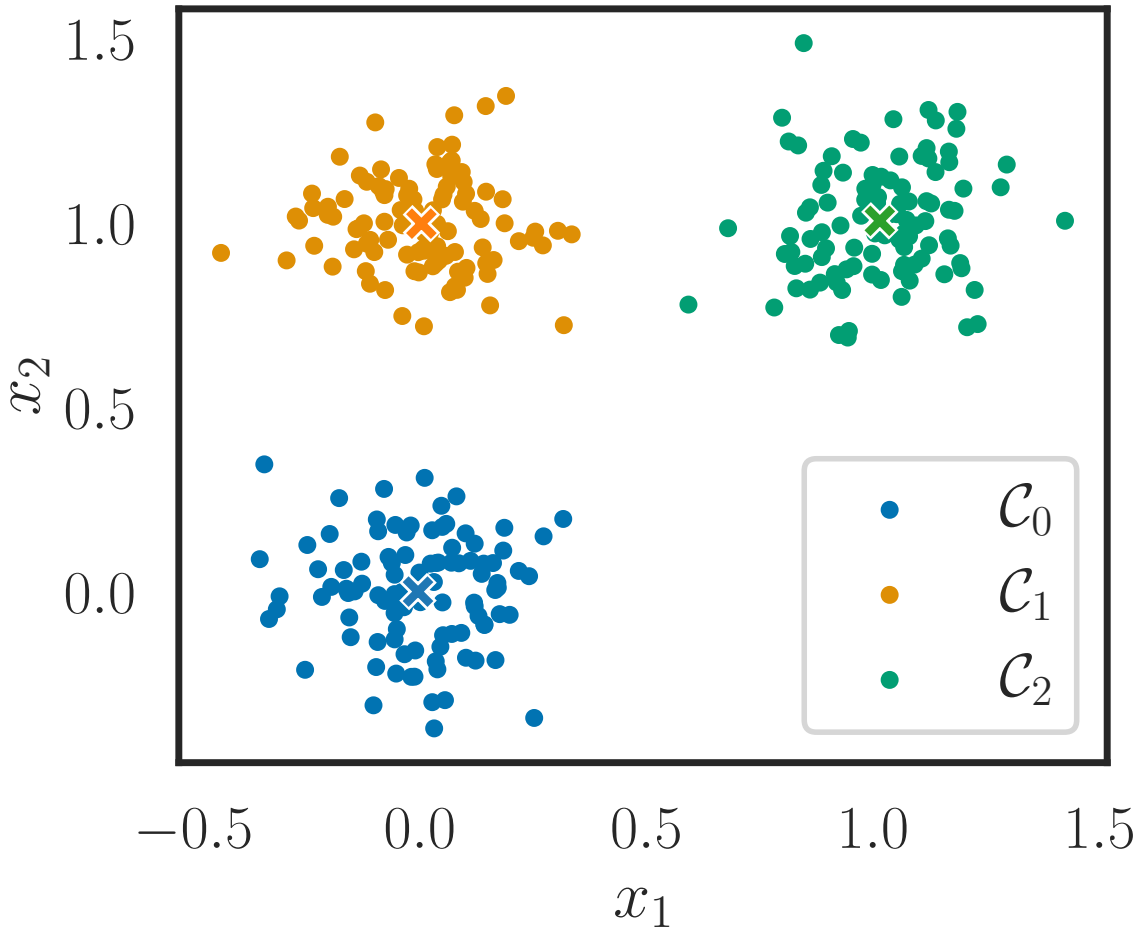
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$



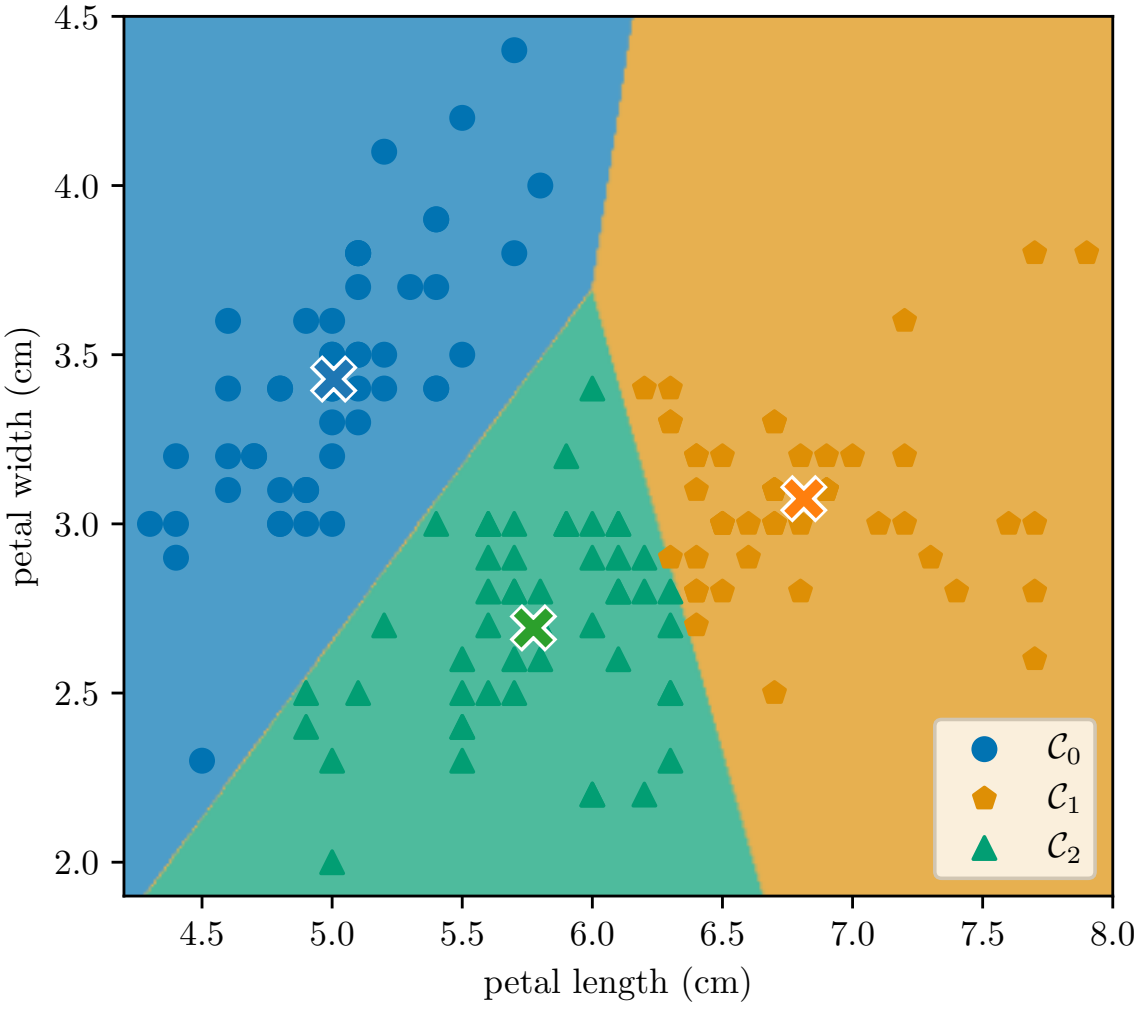
- Facile de généraliser pour les problèmes multi-classes

Exemple avec $d = 2$ et $K = 2$

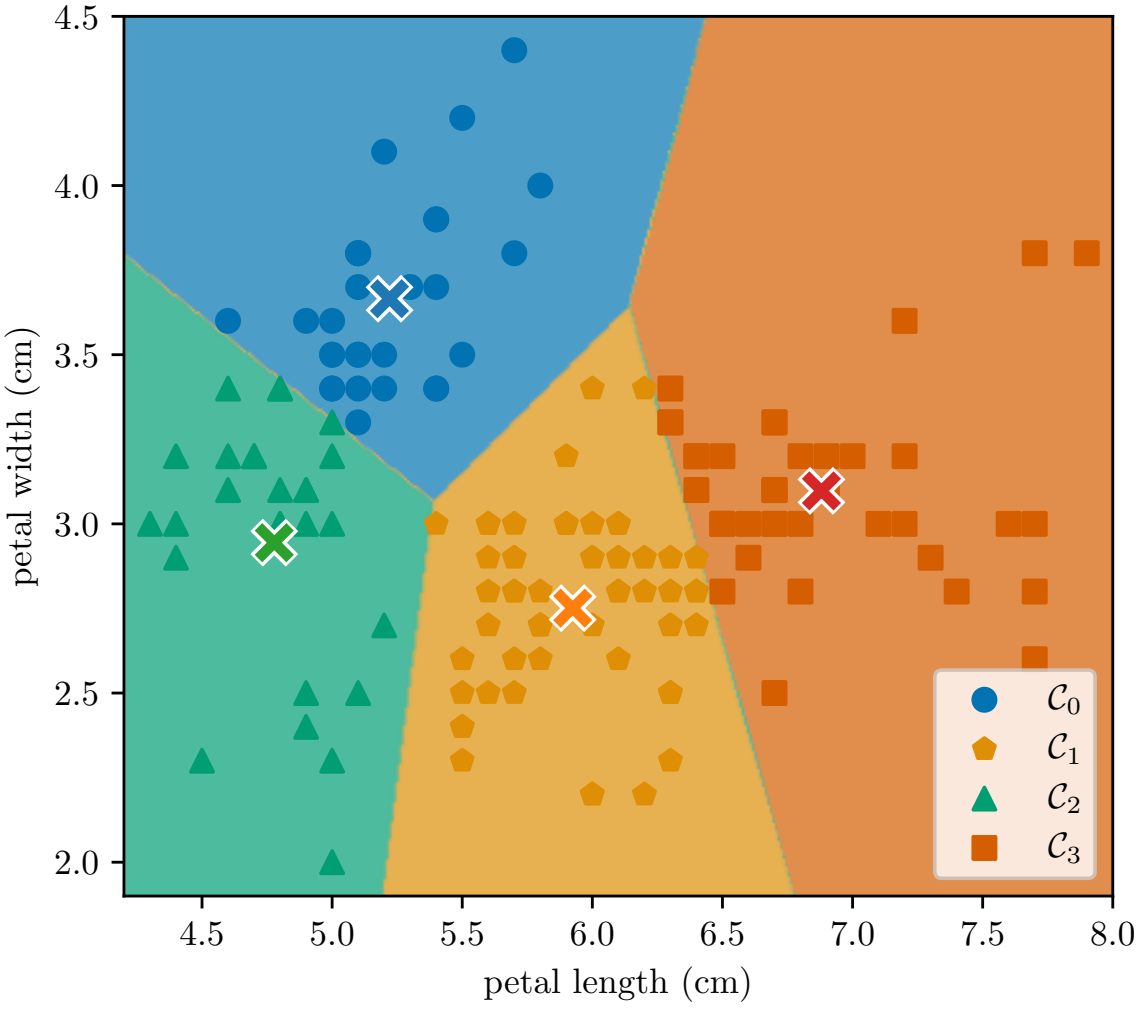


· Facile de généraliser pour les problèmes multi-classes

Cas de test - Iris (sklearn)



Cas de test - Iris (sklearn)



Géométrie des classes

· Les K centres $\mu_0, \ldots, \mu_{K-1}$ induisent une partition de $\mathbb{R}^d$ appelé la **partition de Voronoi** $V_0, \ldots, V_{K-1}$, où :

- $V_k = \left\{ x \in \mathbb{R}^d : \|x - \mu_k\| \leq \min_{\ell \neq k} \|x - \mu_\ell\| \right\}$
- $V_0 \sqcup \cdots \sqcup V_{K-1} = \mathbb{R}^d$

· Les V_k sont appelées **cellule de Voronoi**

· x_i est affecté à la k-ème classe si $\|x - \mu_k\| \leq \min_{\ell \neq k} \|x - \mu_\ell\|$, et dans ce cas, il appartient à la cellule V_k

Rem : Les cellules de Voronoi sont convexes

Convergence

· Si la loi P est connue, alors il est possible de définir K centroïdes optimaux $\mu_0^\star, \ldots, \mu_{K-1}^\star$ tels que

$$\mathcal{E}(\mu_0^\star, \ldots, \mu_{K-1}^\star) = \inf_{\mu_0^\star, \ldots, \mu_{K-1}^\star} \mathcal{E}(\mu_0, \ldots, \mu_{K-1}),$$

où

$$\mathcal{E}(\mu_0, \ldots, \mu_{K-1}) = \mathbb{E}\left(\min_{k \in [\![0, K-1]\!]} \|X - \mu_k\|^2 \right)$$

Théorème

Supposons que $\mathcal{E}$ admette un minimum unique en $(\mu_0^\star, \ldots, \mu_{K-1}^\star)$ (à une permutation d'indice près). Notons $(\widehat{\mu}_{0,n}, \ldots, \widehat{\mu}_{K-1,n})$ un choix de centroïdes minimisant $\mathcal{E}_n$. Alors, pour tout $k \in [\![0, K-1]\!]$, et à une permutation des indices près, on a

$$\widehat{\mu}_{k,n} \xrightarrow{p.s.} \mu_k^\star$$

Exemple en 2D avec une loi uniforme

· Considérons la loi uniforme sur l'intervalle $[0;1]$, *i.e.*, , $X \sim \mathcal{U}(0;1)$, et $K = 2$ classes, de centroïdes a et b

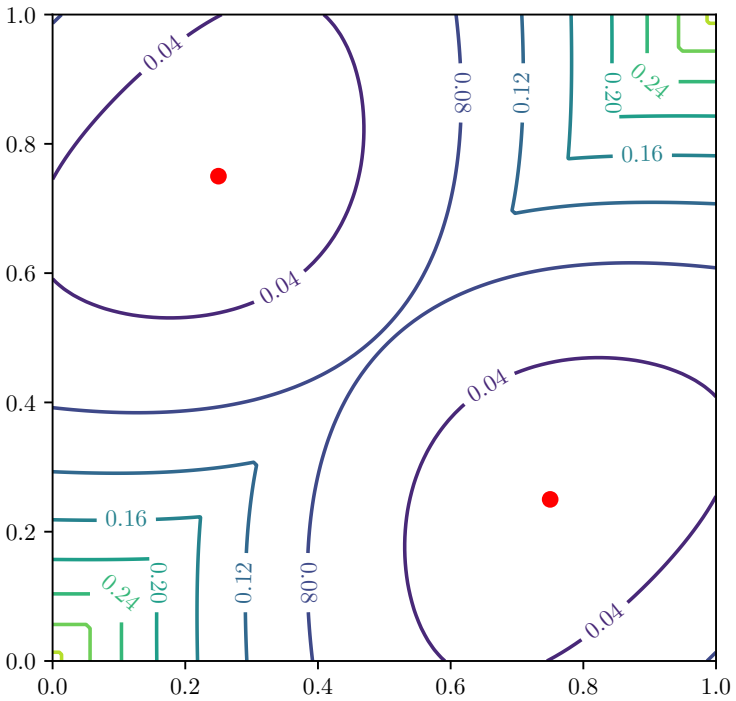
· En supposant $a \leq b$, on a **[exercice]**

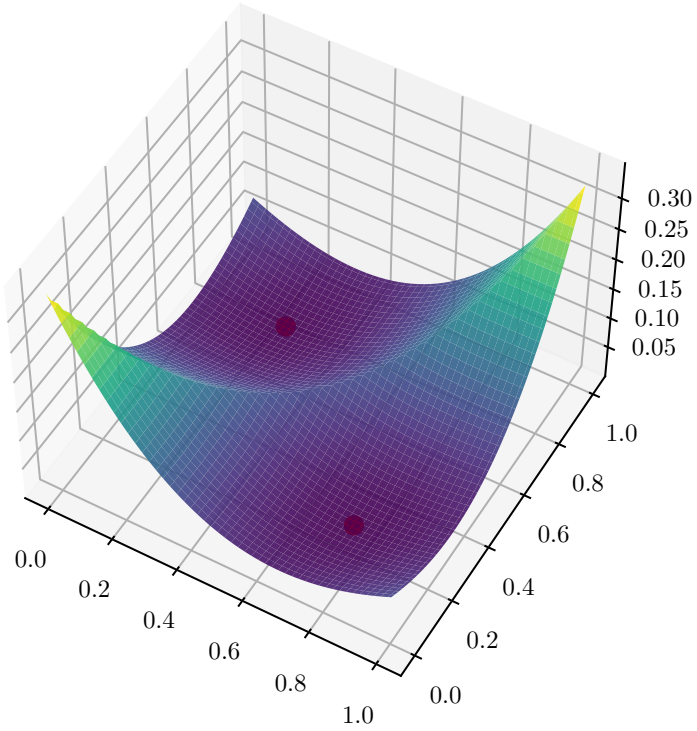
$$\mathcal{E}(a,b) = \frac{1}{3}a^3 + \frac{1}{3}(1-b)^3 + \frac{1}{12}(b-a)^3,$$

où $\mathcal{E}$ admet un minimum unique en

$$(a^\star, b^\star) = \left(\frac{1}{4}, \frac{3}{4}\right)$$

Exemple en 2D avec une loi uniforme





Modèles de mélanges gaussiens

Mélange de gaussiens

· Un mélange de lois gaussiennes est une loi dont la densité s'écrit :

$$f(x) = \sum_{m=1}^{M} \pi_m \phi(x; \mu_m, \Sigma_m), \tag{3}$$

où $x \in \mathbb{R}^d$ et

- $\pi_1, \ldots, \pi_M$ son les coefficients ("poids") du mélange :

$$\sum_{m=1}^{M} \pi_m = 1, \quad \pi_m \geq 0 \; \forall \; m \in [\![1, M]\!]$$

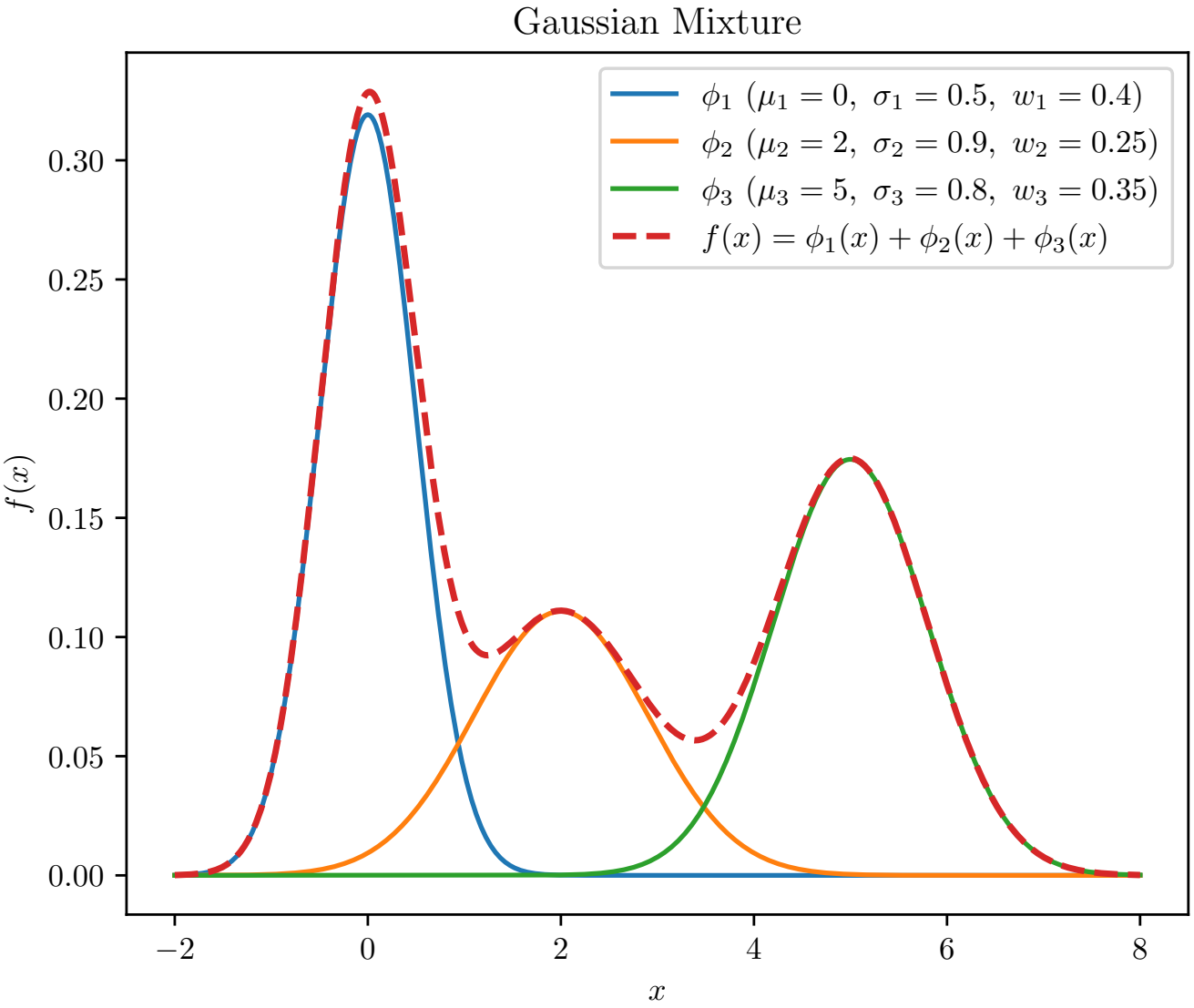
- $\phi(\cdot; \mu_m, \Sigma_m)$ est la densité de la loi gaussienne, de moyenne μ_m et de matrice de covariance Σ_m

$$\phi(x; \mu_m, \Sigma_m) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_m|}} \exp\left(-\frac{1}{2}(x - \mu_m)^\top \Sigma_m^{-1}(x - \mu_m)\right) \tag{4}$$

· Méthode utile pour l'estimation de densités [Hastie et al., 2009]

Rem : autres familles de lois possibles (Cauchy, Laplace, t-student, ...)

Mélange de gaussiens



Modèles de mélanges gaussiens

· Pour le clustering, on s'intéresse à affecter l'observation x_i à un groupe $m \in [\![1, M]\!]$ selon le modèle de mélanges gaussiens

· La probabilité que x_i soit dans le groupe m s'écrit (posterieur) :

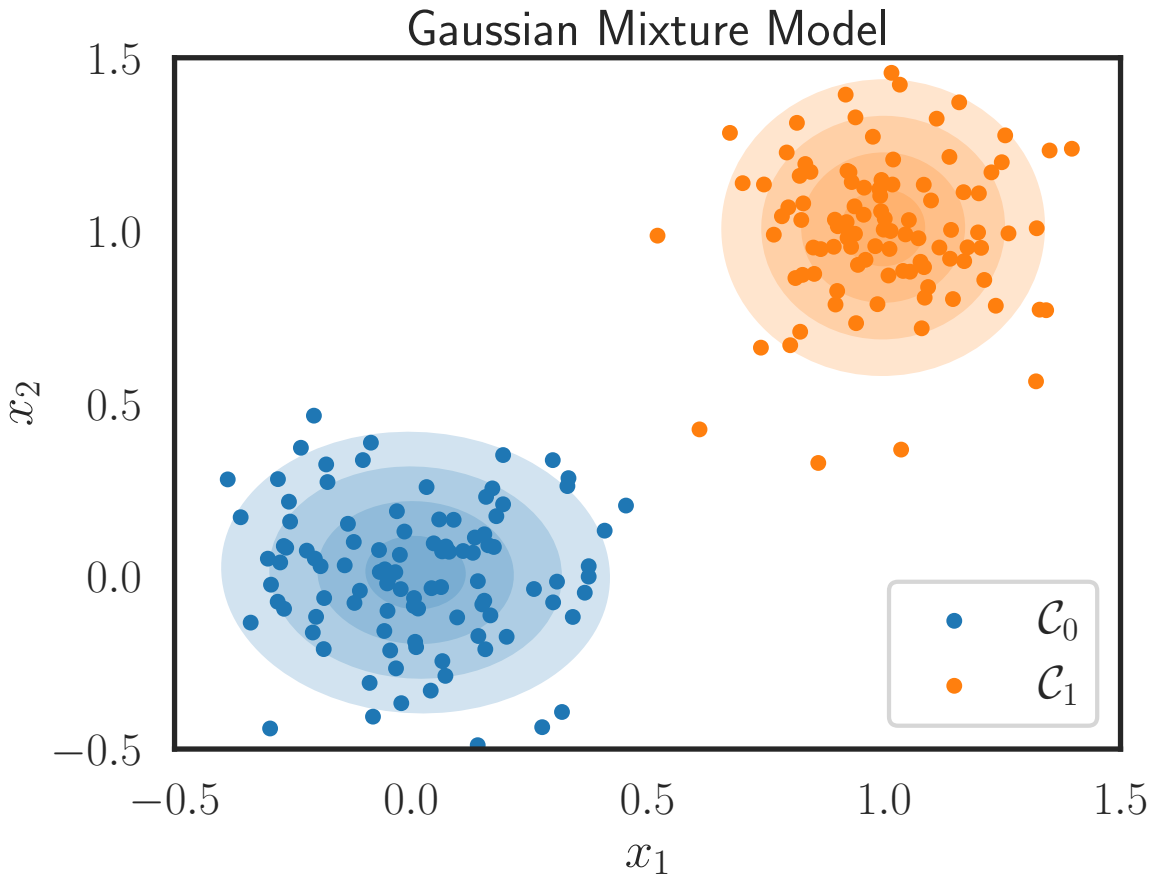
$$\begin{aligned}\mathbb{P}(y_i = m | X_i = x_i) &= \frac{\mathbb{P}(X_i = x_i | y_i = m)\mathbb{P}(y_i = m)}{\sum_{q=1}^{M} \mathbb{P}(X_i = x_i | y_i = q)\mathbb{P}(y_i = q)} \\ &= \frac{\pi_m \phi(x_i; \mu_m, \Sigma_m)}{\sum_{q=1}^{M} \pi_q \phi(x_i; \mu_q, \Sigma_q)} \end{aligned} \tag{5}$$

· Ainsi

$$\begin{aligned}\widehat{y}_i &= \underset{m \in [\![1,M]\!]}{\operatorname{argmax}}\, \mathbb{P}(y_i = m | X_i = x_i) \\ &= \underset{m \in [\![1,M]\!]}{\operatorname{argmax}}\, \pi_m \phi(x_i; \mu_m, \Sigma_m)\end{aligned}$$

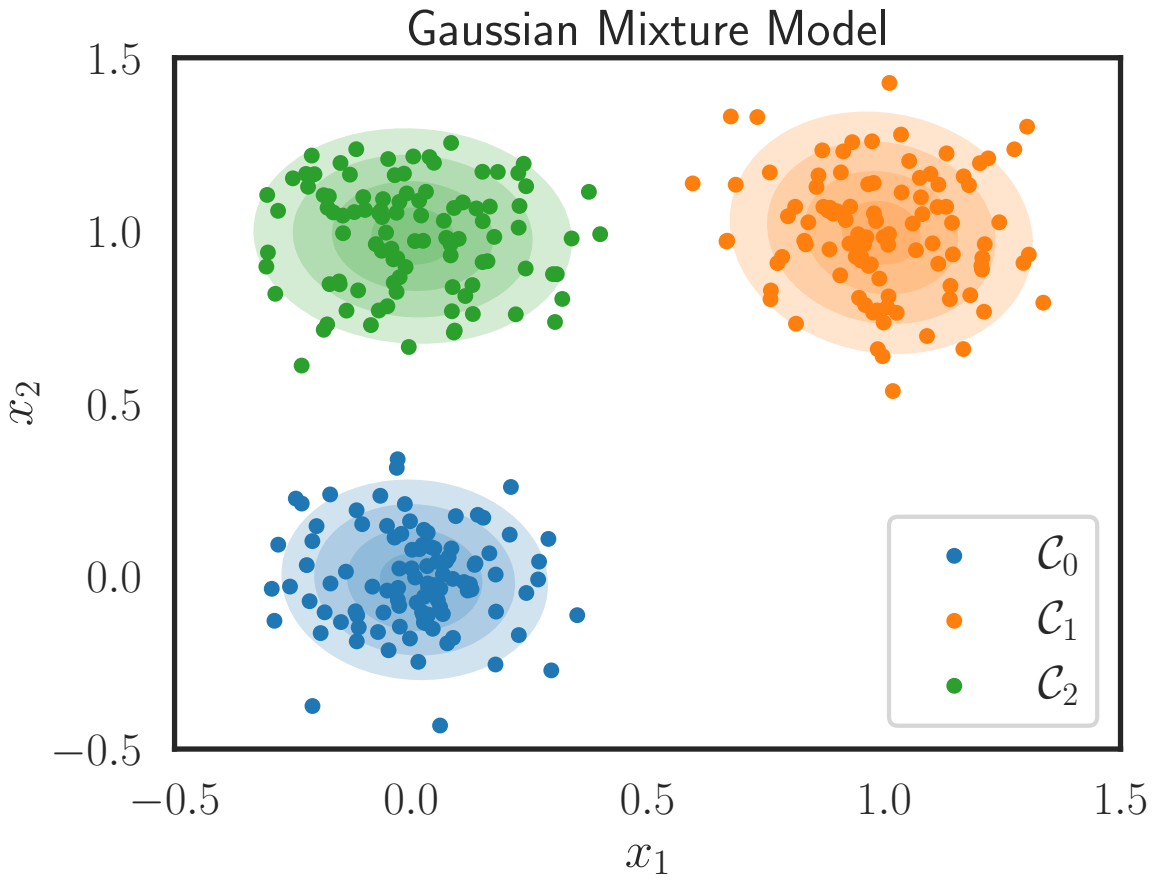
Rem : la variable aléatoire y_i (donnant le groupe auquel x_i appartient) est une variable caché, *i.e.*, non observé, que l'on souhaite reconstruire

Exemple avec $K = 2$



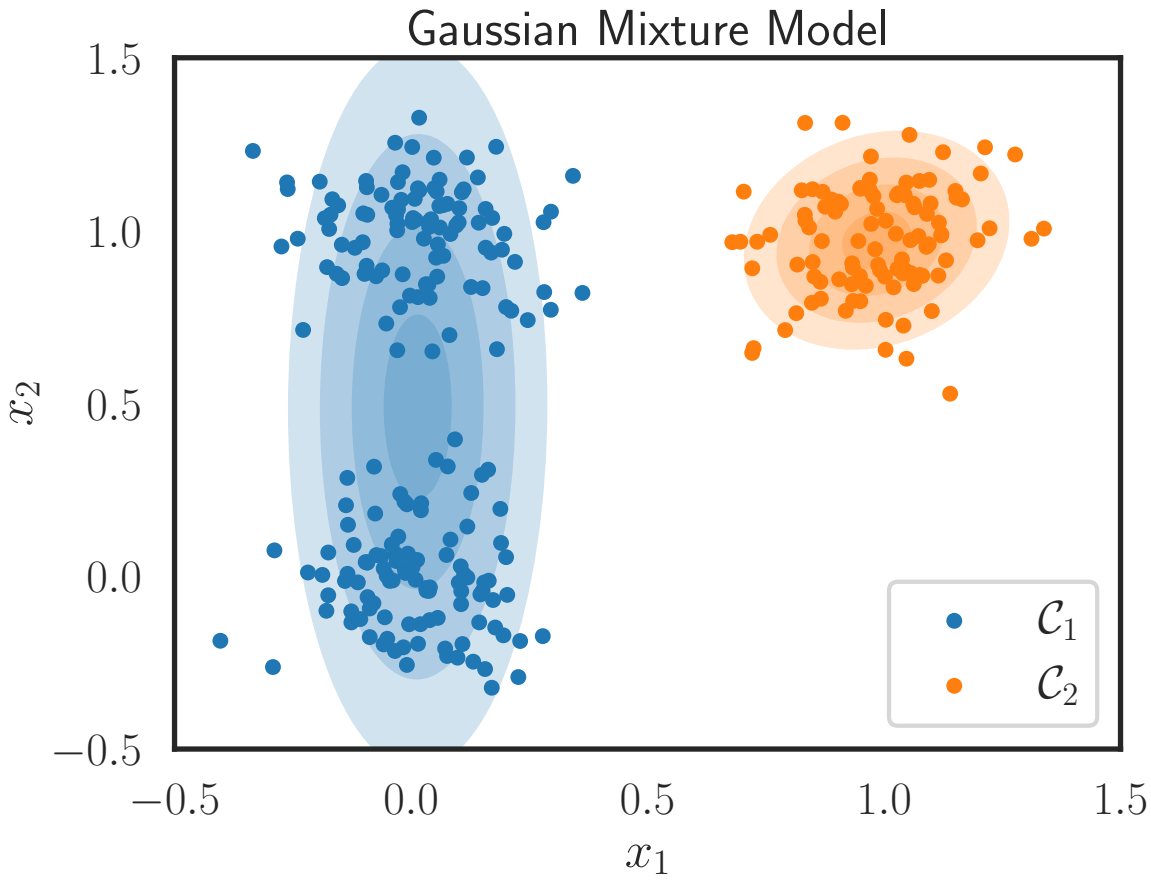
- Performant pour des données séparable linéairement

Exemple avec $K = 3$



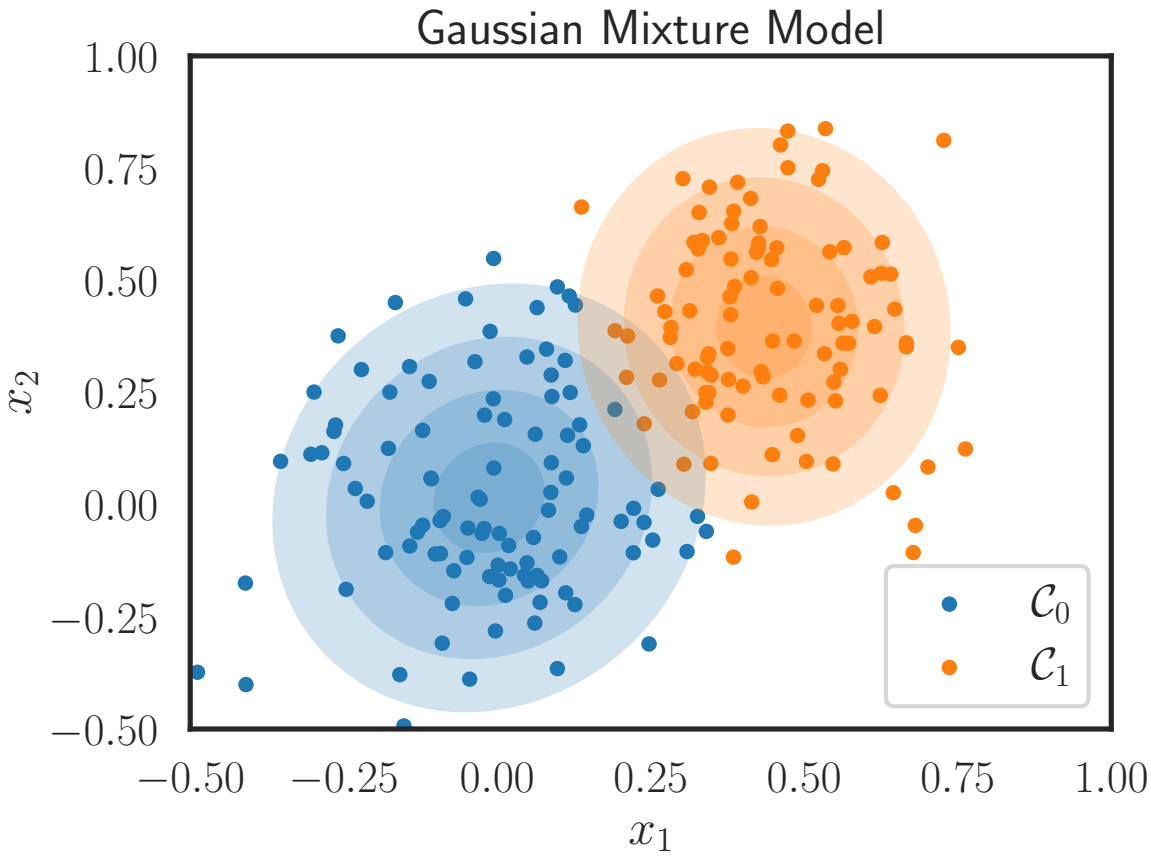
· Performant pour des données séparable linéairement

Exemple avec $K = 3$



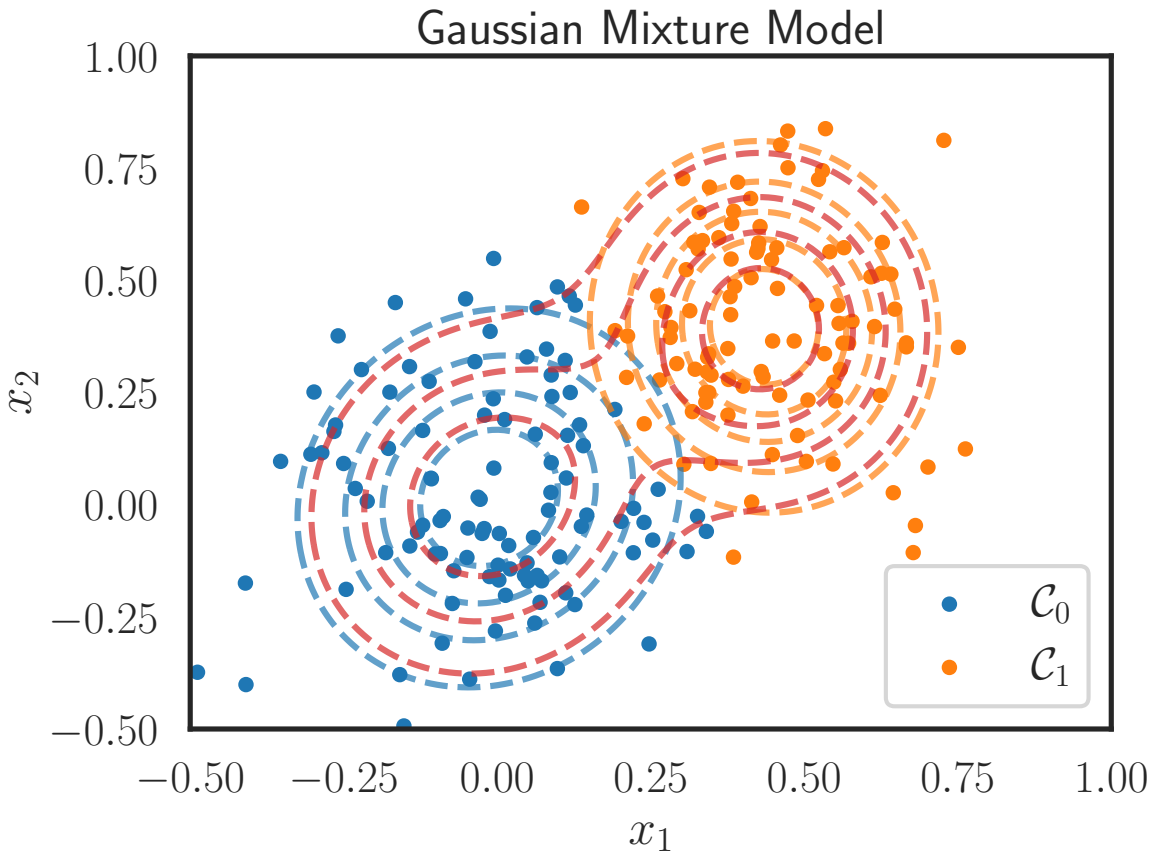
· Performant pour des données séparable linéairement. *M* doit être choisi !

Exemple avec $K = 3$



· Aussi performant pour des données non séparable linéairement

Exemple avec $K = 3$



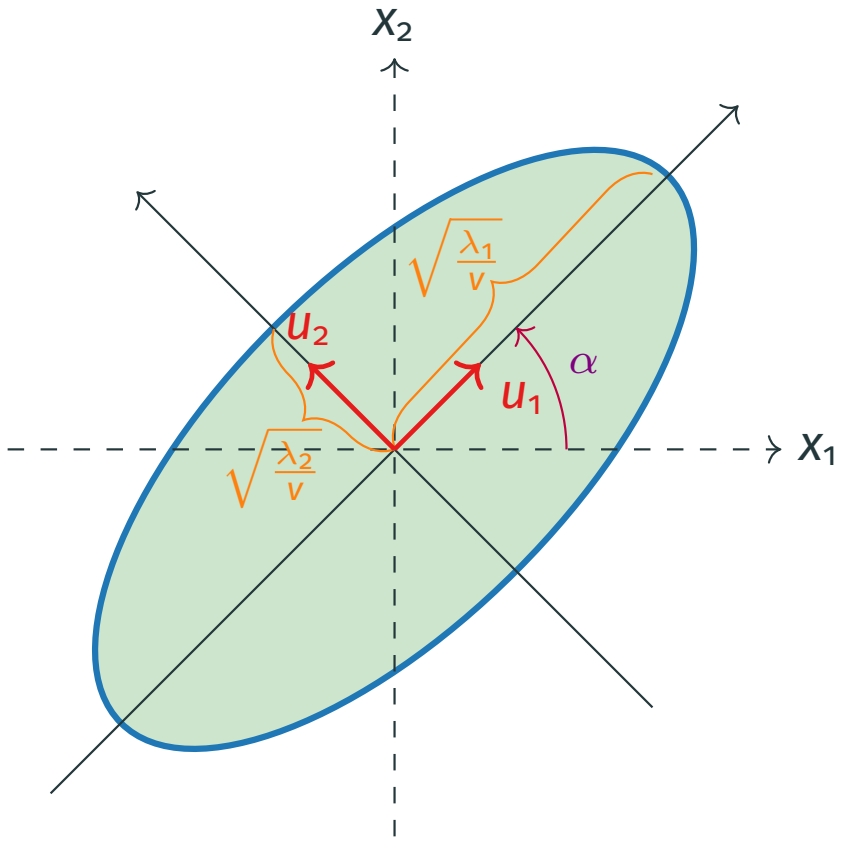
· Aussi performant pour des données non séparable linéairement

Influence de la matrice de covariance

· Une matrice de covariance (*i.e.*, **symétrique** et définie positive) est diagonalisable en base orthonormale :

$$\Sigma = vUDU^\top \tag{6}$$

- son déterminant $v = (\det(\Sigma_m))^{\frac{1}{d}}$ (volume de l'ellipsoïde)
- ses valeurs propres normalisées $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)/v$ (forme)
- ses vecteurs propres normalisés $U = [u_1, \ldots, u_d]$ (orientation)

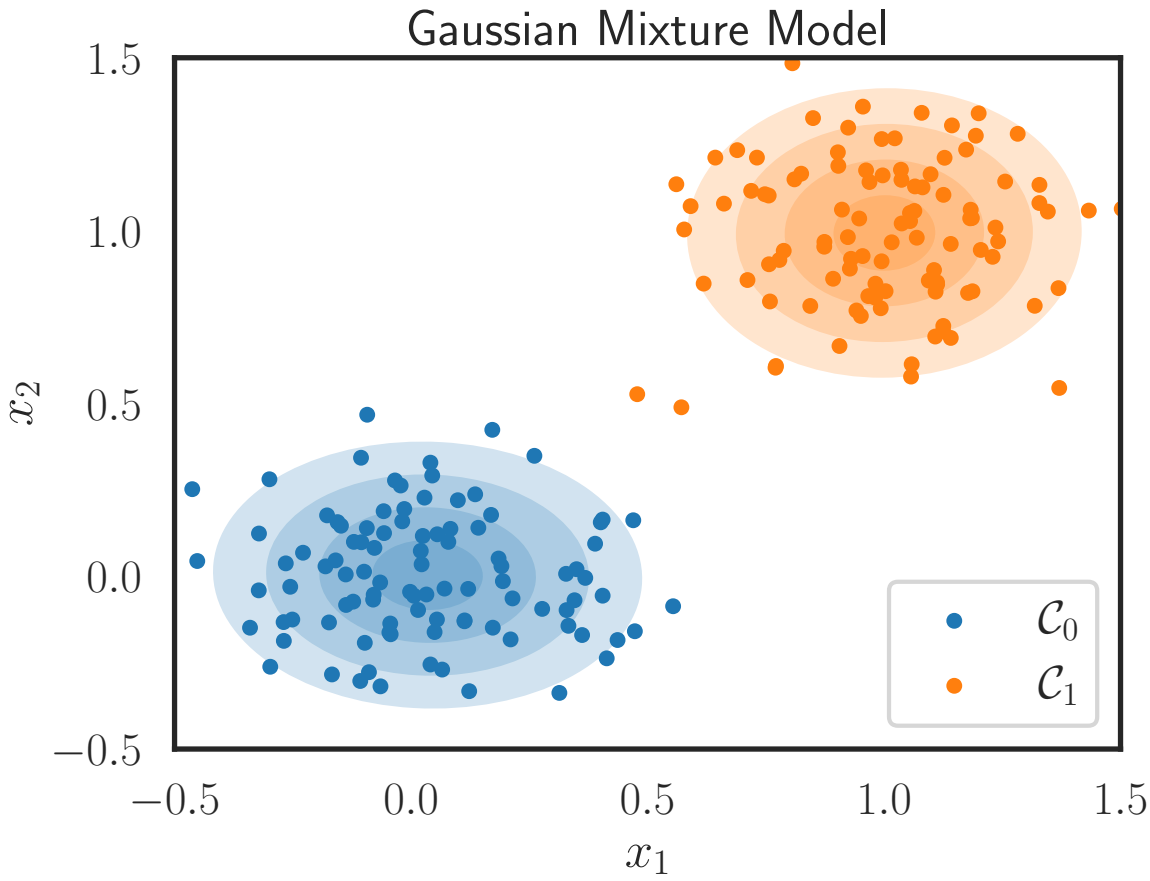


Exemple en 2D

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} u_{1,1} & u_{2,1} \\ u_{1,2} & u_{2,2} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1}\cos(t) \\ \sqrt{\lambda_2}\sin(t) \end{bmatrix}$$

$$\alpha = \arctan\left(\frac{u_{1,2}}{u_{1,1}}\right)$$

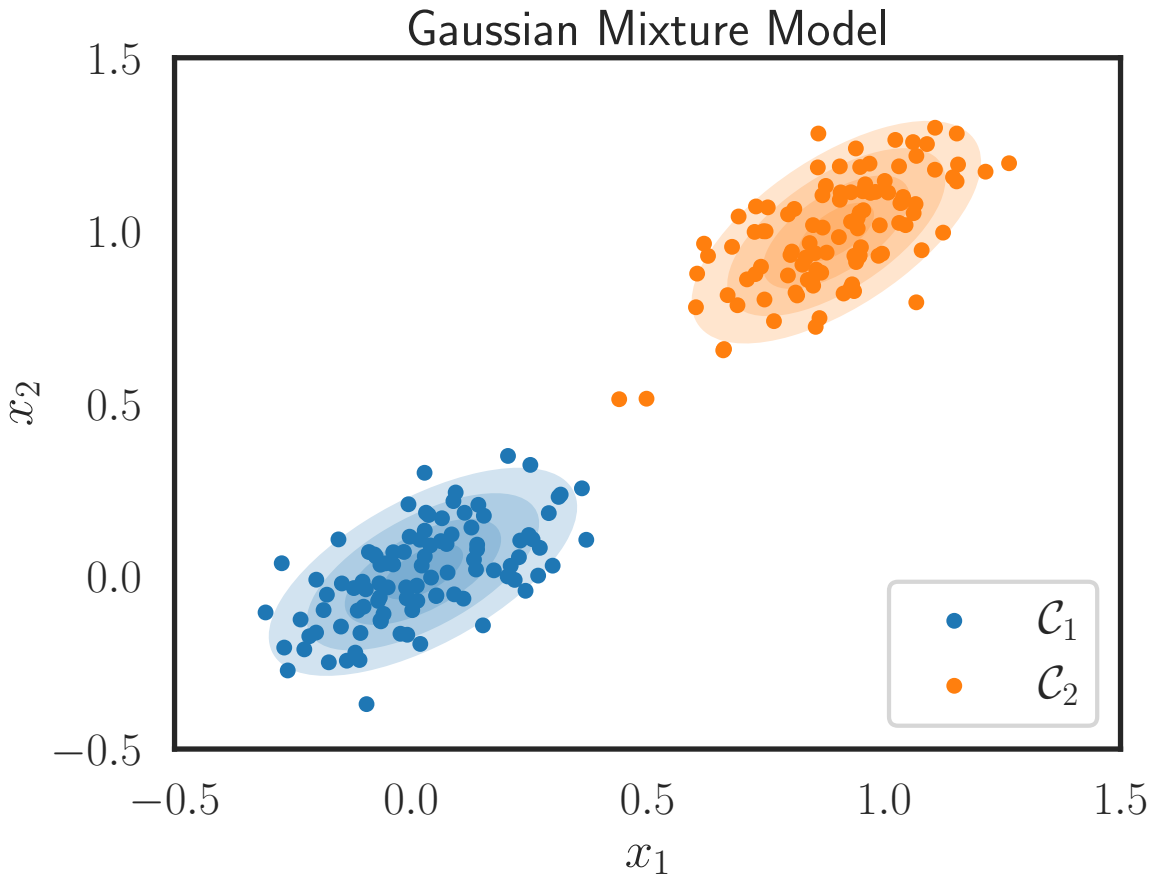
Influence de la matrice de covariance



$D_1 = \text{diag}(1.184, 0.844)$, $v_1 = 0.043$, $\alpha_1 = 175.6$

$D_2 = \text{diag}(1.015, 0.985)$, $v_2 = 0.042$, $\alpha_2 = -135.1$

Influence de la matrice de covariance



$D_1 = \text{diag}(2.124, 0.470)$, $v_1 = 0.018$, $\alpha_1 = -138.9$

$D_2 = \text{diag}(2.138, 0.468)$, $v_1 = 0.018$, $\alpha_2 = 47.1$

Estimation des paramètres

· Paramètres à estimer :

- les coefficients π_m
- les moyennes μ_m
- les matrices de covariance Σ_m
- le nombre de composantes du mélange M (dans un cadre non-paramétrique)

· L'estimation par maximum de vraisemblance est ardue. Sur l'échantillon $x_1, \ldots, x_n$, la log-vraisemblance s'écrit :

$$\mathcal{L}(\theta) = \log \prod_{i=1}^{n} f(x_i; \theta) = \sum_{i=1}^{n} \log \left(\sum_{m=1}^{M} \pi_m \phi(x_i; \mu_m, \Sigma_m) \right)$$

· Pas de formule analytique pour les estimateurs $\widehat{\pi}_m, \widehat{\mu}_m$ et $\widehat{\Sigma}_m$ si $M > 1$

Estimation des paramètres

· Condition d'optimalité pour les moyennes μ_m :

$$\frac{\partial}{\partial \mu_k}\mathcal{L}(\theta) = \sum_{i=1}^{n} \frac{\partial}{\partial \mu_k} \log\left(\sum_{m=1}^{M} \pi_m \phi(x_i; \mu_m, \Sigma_m)\right)$$

$$= \sum_{i=1}^{n} \underbrace{\frac{\pi_k \phi(x_i; \mu_k, \Sigma_k)}{\sum_{m=1}^{M} \pi_m \phi(x_i; \mu_m, \Sigma_m)}}_{r_{i,k}} \Sigma_k^{-1}(x_i - \mu_k) = 0,$$

d'où on en déduit

$$\mu_k = \frac{\sum_{i=1}^{n} r_{i,k}\, x_i}{\sum_{i=1}^{n} r_{i,k}} = \frac{1}{\eta_k} \sum_{i=1}^{n} r_{i,k}\, x_i$$

· Condition d'optimalité pour les matrices de covariance Σ_m :

$$\frac{\partial}{\partial \Sigma_k}\mathcal{L}(\theta) = \sum_{i=1}^{n} \frac{\partial}{\partial \Sigma_k} \log\left(\sum_{m=1}^{M} \pi_m \phi(x_i; \mu_m, \Sigma_m)\right) = 0,$$

d'où on en déduit

$$\Sigma_k = \frac{1}{\eta_k} \sum_{i=1}^{n} r_{i,k}\, (x_i - \mu_k)(x_i - \mu_k)^\top$$

Rem : $r_{i,k}$ et η_k dépendent de μ_k et Σ_k !

Estimation des paramètres

· Condition d'optimalité pour les coefficients π_m (via le multiplicateur de Lagrange) :

$$\begin{aligned}\frac{\partial}{\partial \pi_k}\mathcal{L}(\theta) &= \frac{\partial}{\partial \pi_k}\left[\sum_{i=1}^{n}\log\left(\sum_{m=1}^{M}\pi_m\phi(x_i;\mu_m,\Sigma_m)\right)+\lambda\left(\sum_{i=1}^{M}\pi_m-1\right)\right]\\ &= \sum_{i=1}^{n}\frac{\phi(x_i;\mu_k,\Sigma_k)}{\sum_{m=1}^{M}\pi_m\phi(x_i;\mu_m,\Sigma_m)}+\lambda\\ &= \sum_{i=1}^{n}\frac{r_{i,k}}{\pi_k}+\lambda = 0,\end{aligned}$$

d'où on en déduit $\pi_k\lambda = -\sum_{i=1}^{n} r_{i,k}$

· En effectuant la somme pour tout k :

$$\lambda = \sum_{k=1}^{M}\pi_k\lambda = -\sum_{i=1}^{n}\sum_{k=1}^{M} r_{i,k} = -n$$

· En utilisant l'expression précédente pour remplacer λ, on obtient enfin

$$\pi_k = \frac{\eta_k}{n}$$

Principe de l'algorithme EM

· Maximisation directe de la vraisemblance difficile $\Longrightarrow$ approche alternée (comme pour k-means / algorithme de Lloyd)

Algorithm Algorithme EM (***E**xpectation - **M**aximisation*)

Données d'entré : $x_1, \ldots, x_n$, $M \leq n$

1: Initialiser les paramètres du modèle de mélanges $\mu_m^{(0)}$, $\Sigma_m^{(0)}$ et $\pi_m^{(0)}$ (*e.g.*, au hasard)

2: **for** $k = 0, 1, \ldots$ (jusqu'à convergence) **do**

3: **E-step :** Pour chaque donnée x_i, calculer la probabilité que x_i soit dans le groupe m :

$$\mathbb{P}(y_i = m | X_i = x_i) = \frac{\pi_m^{(k)} \phi(x_i; \mu_m^{(k)}, \Sigma_m^{(k)})}{\sum_{q=1}^{M} \pi_q^{(k)} \phi(x_i; \mu_q^{(k)}, \Sigma_q^{(k)})} \quad (:= r_{i,m}^{(k)})$$

4: **M-step :** Étant données les affectations des données en groupes, estimer les paramètres $\mu_m^{(k+1)}$, $\Sigma_m^{(k+1)}$ et $\pi_m^{(k+1)}$ par maximum de vraisemblance

$$\mathcal{L}(\theta) = \log \prod_{i=1}^{n} f(x_i; \theta) = \sum_{i=1}^{n} \log \sum_{m=1}^{M} \pi_m \phi(x_i; \mu_m, \Sigma_m)$$

Exemple avec $d = 1$ et $M = 2$

$$f(x) = \pi_1 \phi(x; \mu_1, \sigma_1) + \pi_2 \phi(x; \mu_2, \sigma_2)$$

Initialisation. $\pi_1^{(0)}, \pi_2^{(0)} = 1 - \pi_1^{(0)}, \mu_1^{(0)}, \mu_2^{(0)}, \sigma_1^{(0)}, \sigma_2^{(0)}$

E-step. Connaissant $\pi_1^{(k)}, \mu_1^{(k)}, \mu_2^{(k)}, \sigma_1^{(k)}$ et $\sigma_2^{(k)}$, on estime la probabilité que x_i soit dans le groupe $m \in [\![1, 2]\!]$ par (pour tout $i \in [\![1, n]\!]$) :

$$r_{i,m}^{(k)} = \frac{\pi_m^{(k)} \phi(x_i; \mu_m^{(k)}, \sigma_m^{(k)})}{\pi_1^{(k)} \phi(x_i; \mu_1^{(k)}, \sigma_1^{(k)}) + (1 - \pi_1^{(k)}) \phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}$$

M-step. On calcule les nouvelles moyennes et variances :

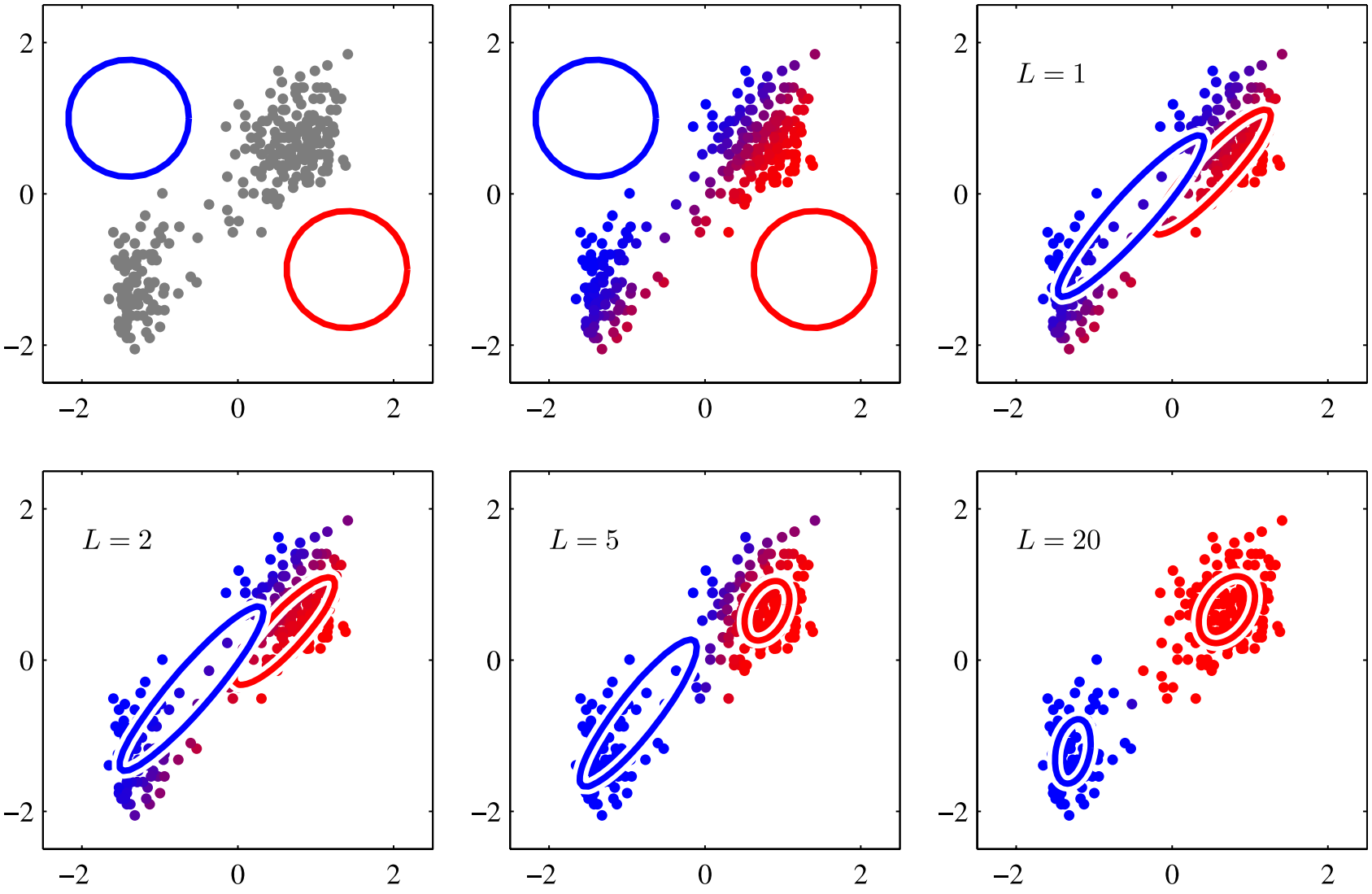
$$\mu_m^{(k+1)} = \frac{\sum_{i=1}^n r_{i,m}^{(k)} x_i}{\sum_{i=1}^n r_{i,m}^{(k)}}, \qquad (\sigma_m^{(k+1)})^2 = \frac{\sum_{i=1}^n r_{i,m}^{(k)} (x_i - \mu_m^{(k)})^2}{\sum_{i=1}^n r_{i,m}^{(k)}}$$

et enfin, probabilité du mélange :

$$\pi_m^{(k+1)} = \frac{1}{n} \sum_{i=1}^n r_{i,m}^{(k)}$$

Rem : $r_{i,2}^{(k)} = 1 - r_{i,1}^{(k)}$

Exemple avec $d = 2$ et $M = 2$



[Bishop, 2006]

Estimation des paramètres

· La maximisation de la vraisemblance présente un problème en raison de la présence de singularités

· Supposons une famille sphérique des covariance $\Sigma_m = \sigma_m^2 I_d$

· Si $\mu_m = x_i$ pour un certain $m \in [\![1, M]\!]$ et $i \in [\![1, n]\!]$, on a

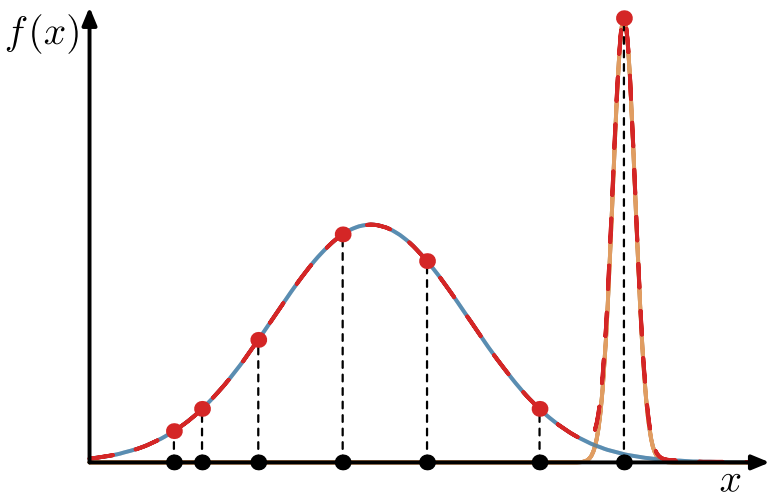
$$\phi(x_i; \mu_m, \Sigma_m) = \frac{1}{\sqrt{(2\pi)^d \sigma_m^d}}$$

· On constate que

$$\phi(x_i; \mu_m, \Sigma_m) \xrightarrow[\sigma_m \to 0]{} \infty,$$

et par conséquent, la vraisemblance divergera (problème mal posé)

· La conclusion s'applique aux matrices de covariance Σ_m générales



Complexité du modèle

· Pour M composantes, avec $x \in \mathbb{R}^d$, les paramètres sont :

- M moyennes μ_m, soit $M \times d$ réels
- M matrice de covariances Σ_m, soit $M \times d(d+1)/2$ réel
- $(M-1)$ coefficients π_m

· Il est possible de rajouter des hypothèses sur Σ_m pour diminuer la complexité :

- **Famille diagonale :** $\Sigma_m = \text{diag}(\sigma^2_{m,1}, \ldots, \sigma^2_{m,d})$ ($M \times d$ paramètres)
- **Famille sphérique :** $\Sigma_m = \sigma^2_m I_d$ (M paramètres)
- **Famille isotrope :** $\Sigma_m = \sigma^2 I_d$ (un seul paramètre, trop restrictif)

Recommandations et logiciels

Recommandations pratiques

- Utiliser k-means pour l'initialisation des paramètres
- Procédure *multi-start* pour trouver le "meilleur" minimum local
- Considérer des heuristiques pour éviter la présence de singularités

Toolboxes

- Python : sklearn.mixture.GaussianMixture
- R : mclust

Lien utile

- Panorama de clustering [Pedregosa et al., 2011] : https://scikit-learn.org/1.5/auto_examples/cluster/

Lien avec k-means

- **k-means**
  - Estimation de K centroïdes
  - Chaque donnée est affectée au centroïde le plus proche
  - La partition obtenue dépend des centroïdes
- **Mélange de gaussiens**
  - Estimation M moyennes et matrices de covariance
  - Chaque donnée est affecté au groupe dont la composante du mélange est la plus probable
  - La partition obtenue dépend des moyennes et des matrices de covariances (qui déterminent la forme des groupes)

Rem : L'algorithme k-means peut-être dérive de la méthode EM pour les mélanges gaussiens quand $\Sigma_m = \epsilon I$ et en faisant tendre $\epsilon \to 0$ [Bishop, 2006]

Lien avec les estimateurs à noyau de la densité

· Posons [Hastie et al., 2009]

$$\widehat{f}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \phi\left(\frac{x - x_i}{h}\right),$$

où $\phi(x) = \phi(x; 0, 1)$

· Dans ce cas

- $M = n$ (autant de composantes que d'observations)
- $\pi_m = \frac{1}{n}$ pour tout $m \in [\![1, n]\!]$
- $\mu_m = x_m$ et $\sigma_m = h$ ("taille de fenêtre" à choisir)

Références

David Arthur and Sergei Vassilvitskii. *k*-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, second edition, 2009.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.