

IFT 4030/7030,  
Machine Learning for Signal Processing  
**Week9: Machine Learning 4**  
**Clustering**

Cem Subakan



UNIVERSITÉ  
**Laval**



- On va publier le devoir 2 bientot. Merci pour votre patience!
  - ▶ Homework 2 will be out soon. (Bear with me)
- J'espère vous avez déjà commencé vos projets.
  - ▶ I really hope you have started working on your projects!.
- Aujourd'hui: Clustering
  - ▶ Today: Clustering

# Clustering

---

- Let's kick things off with clustering. / On va commencer avec clustering.
- We were doing supervised learning for the past two classes. Now we will change.
  - ▶ On faisait de l'apprentissage supervisé pour les deux derniers classes.

# Clustering

---

- Let's kick things off with clustering. / On va commencer avec clustering.
- We were doing supervised learning for the past two classes. Now we will change.
  - ▶ On faisait de l'apprentissage supervisé pour les deux derniers classes.
- What if we do not have the labels?
  - ▶ Et si on n'avait pas d'étiquettes?

# Clustering

---

- Let's kick things off with clustering. / On va commencer avec clustering.
- We were doing supervised learning for the past two classes. Now we will change.
  - ▶ On faisait de l'apprentissage supervisé pour les deux derniers classes.
- What if we do not have the labels?
  - ▶ Et si on n'avait pas d'étiquettes?
- Today's lecture's goal / Le but d'aujourd'hui.
  - ▶ I will try to acclimate you with clustering./Je vais essayer de vous introduire des concepts de base de clustering.
  - ▶ But in a sneaky way I will introduce powerful tools from probabilistic machine learning. / Je vais vous bombarder silencieusement avec des outils de l'apprentissage automatique probabilistique.

# Some motivation

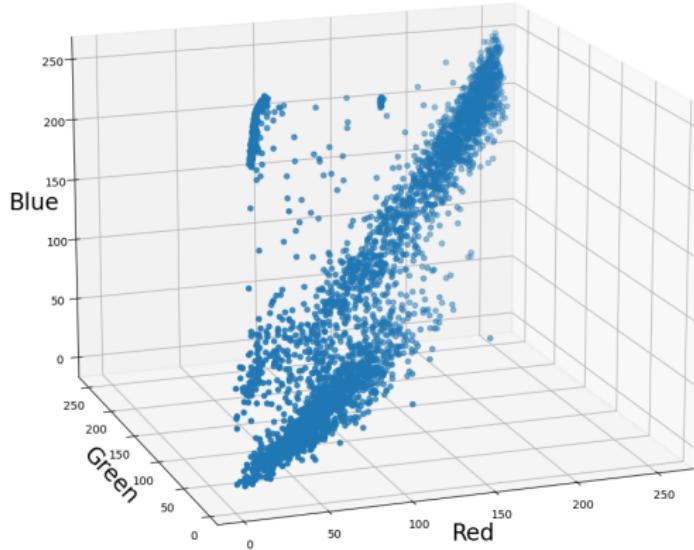
---



El Capitan, Yosemite National Park, California

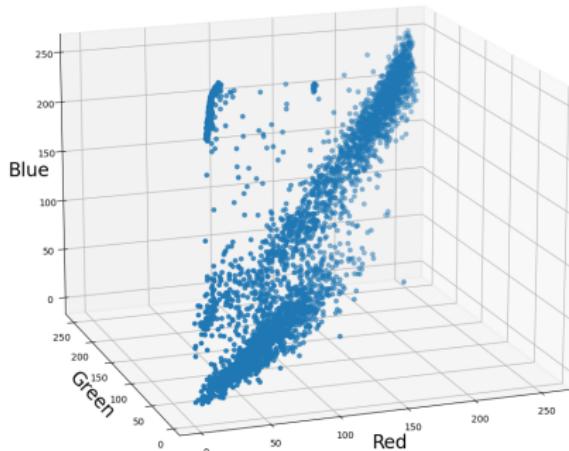
# How many clusters?

---



# Clustering

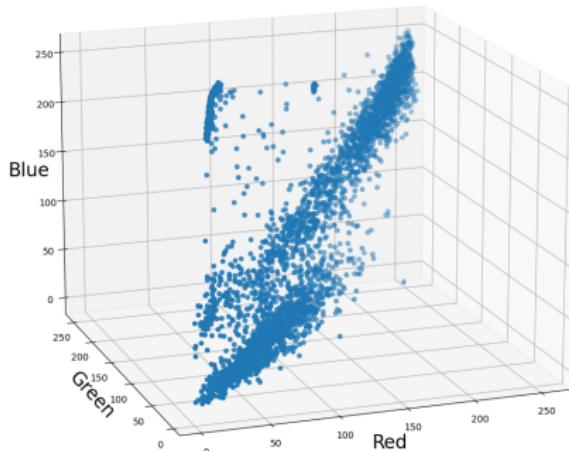
---



- We see clusters, but how do we find them?
  - ▶ On voit des groupes, mais comment on les trouve?
- Can I find an algorithm for this? / Peut-on trouver un algorithme pour cela?

# Clustering

---



- We see clusters, but how do we find them?
  - ▶ On voit des groupes, mais comment on les trouve?
- Can I find an algorithm for this? / Peut-on trouver un algorithme pour cela?
- Clustering!

# Clustering

---

- **Clustering:** We discover clusters/groups in the data.
  - ▶ **Clustering:** On découvre des groupes dans les données.
- Fundamentally **ill defined** problem. There is often no correct solution.
  - ▶ Clustering n'est pas un problème bien défini.
- Relies on user choices.
  - ▶ Ça dépend sur les choix de l'utilisateur.

# Clustering process

---

## ■ Features

Describe your data using features. / Quels features utilise-t-on pour représenter les données?

## ■ Cluster Shapes

Decide what your clusters should look like / Il faut décider comment on veut que les clusters se forment.

## ■ Features

Describe your data using features. / Quels features utilise-t-on pour représenter les données?

## ■ Cluster Shapes

Decide what your clusters should look like / Il faut décider comment on veut que les clusters se forment.

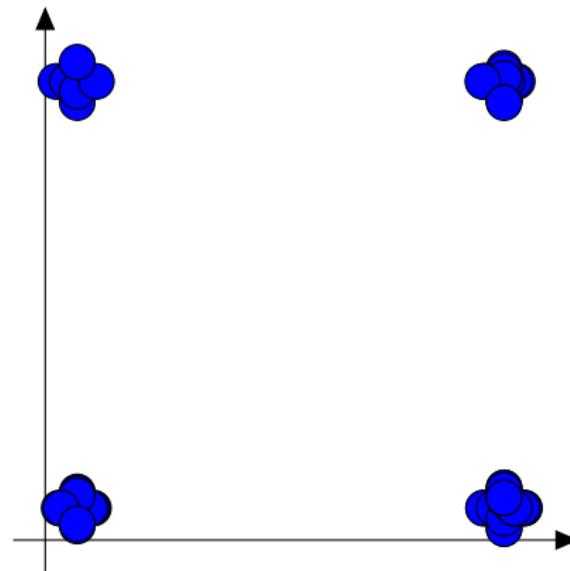
## ■ Distance

Define a distance function / proximity measure.

- ▶ Définissons une notion de distance / proximité.

# Features

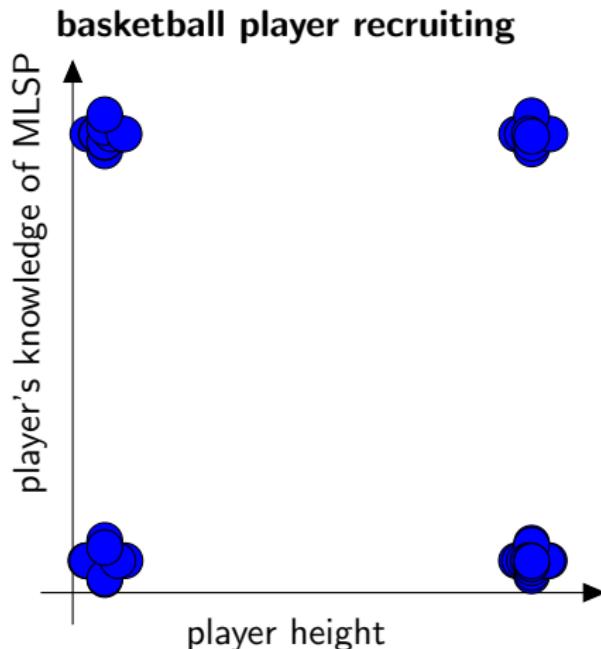
---



- How many clusters? / Combien de groupes?

# Features

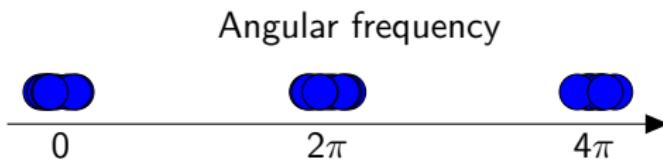
---



- How many clusters? / Combien de groupes?
- How many clusters? / Combien de groupes?

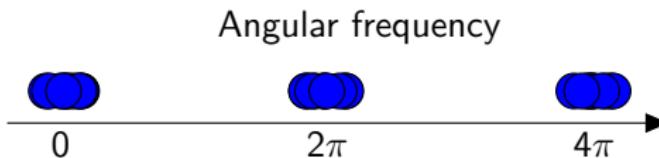
# Use a sensible distance function

---



# Use a sensible distance function

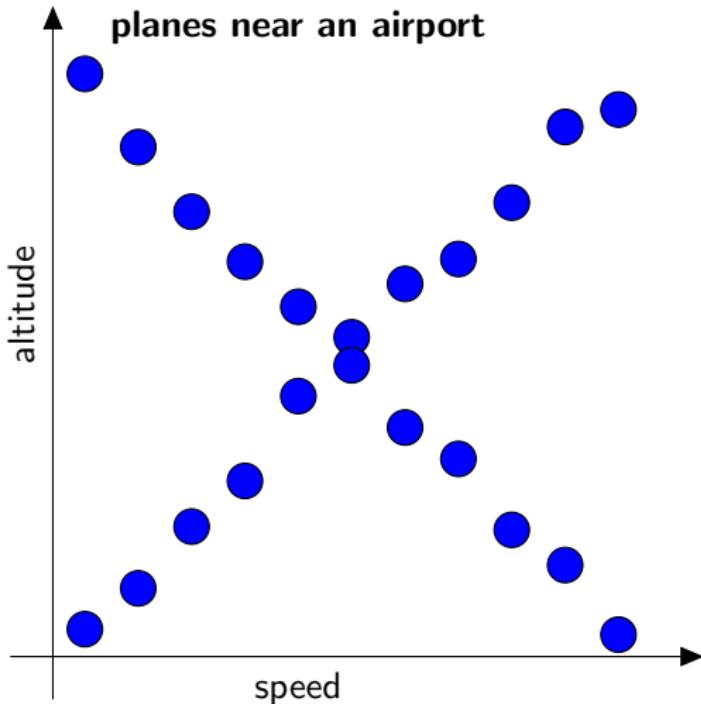
---



- Probably using euclidean distance is not a good idea here!
  - ▶ Très probable qu'il faut pas utiliser la distance euclidienne ici!!

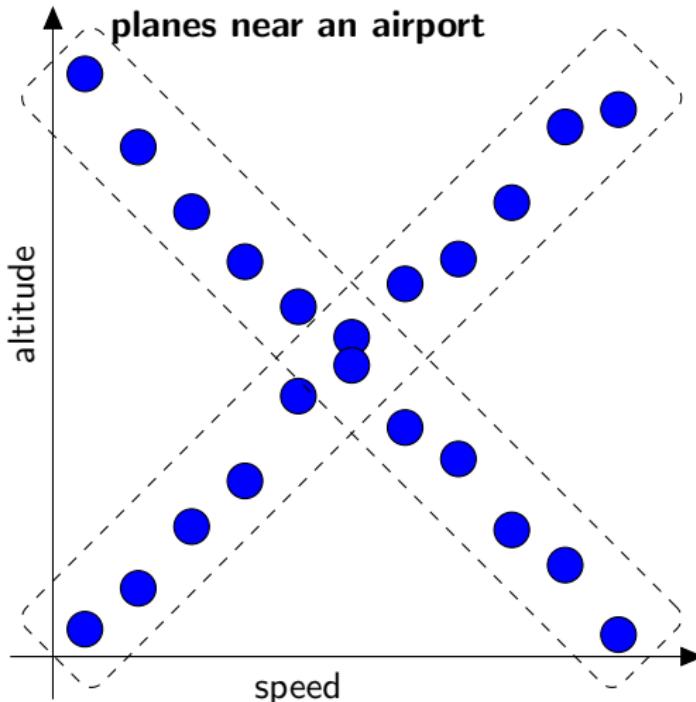
## What forms clusters in your space?

---



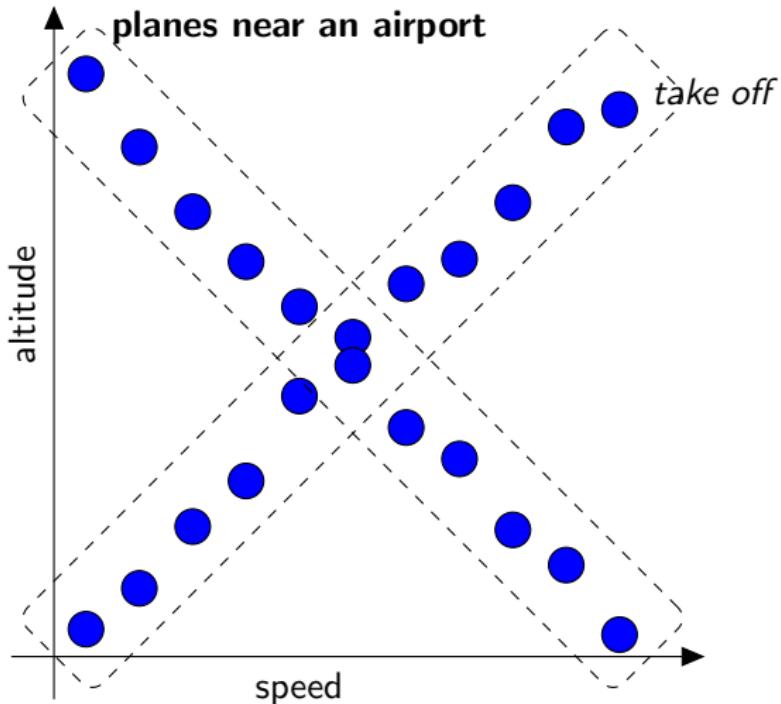
# What forms clusters in your space?

---



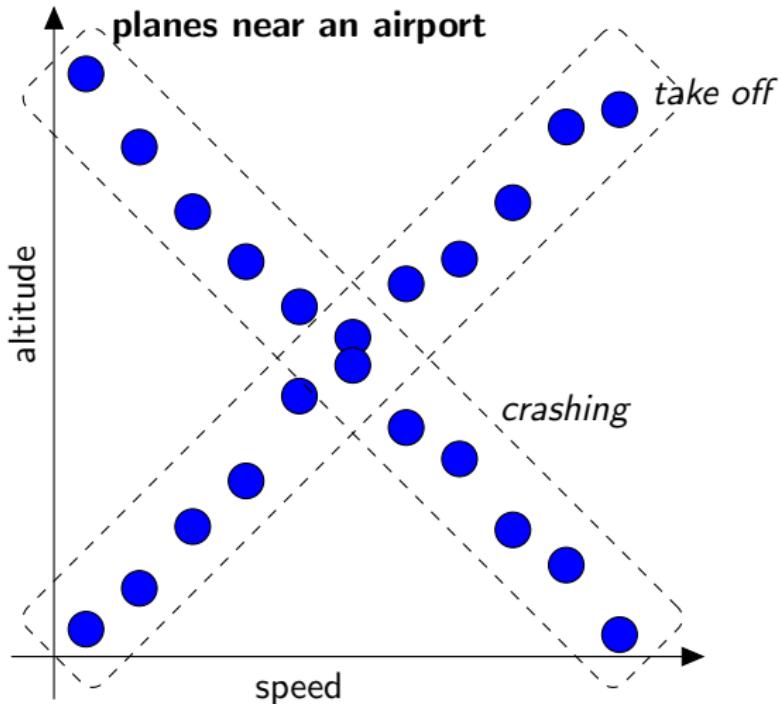
# What forms clusters in your space?

---



# What forms clusters in your space?

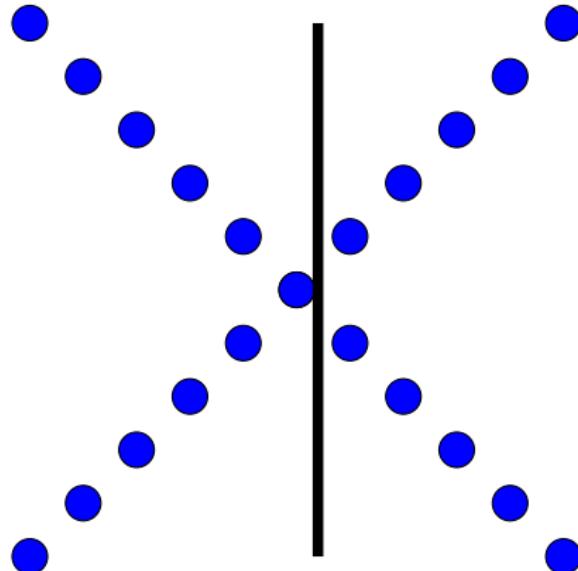
---



# What forms clusters in your space?

---

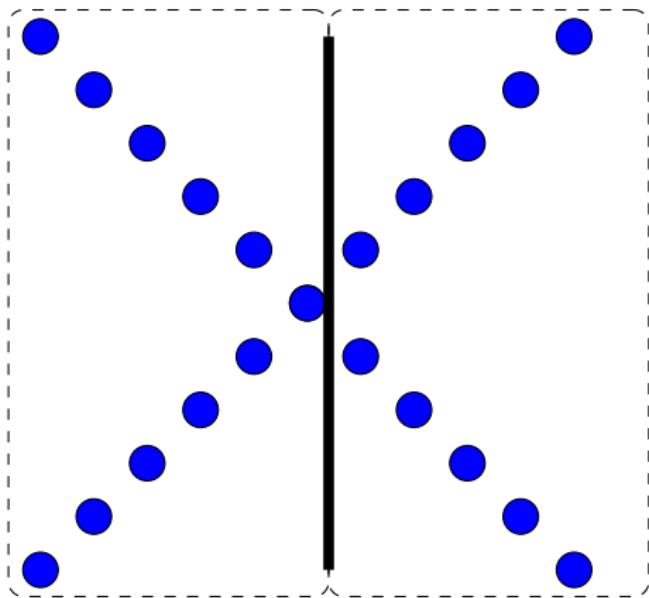
ball trajectories bouncing off a wall



# What forms clusters in your space?

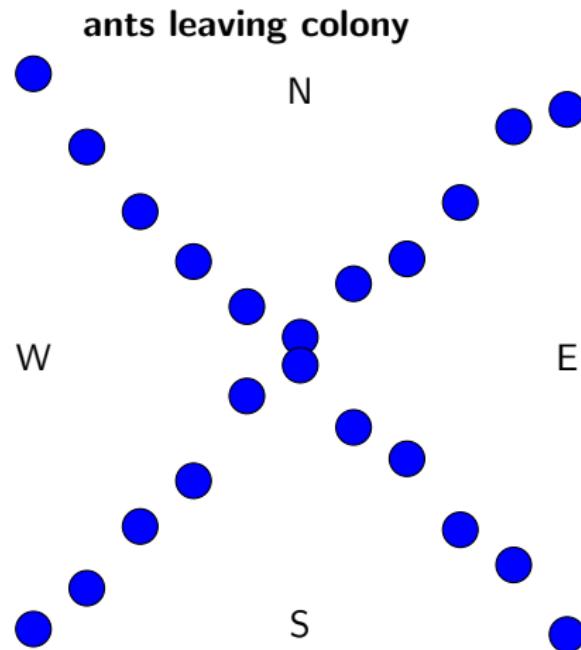
---

ball trajectories bouncing off a wall



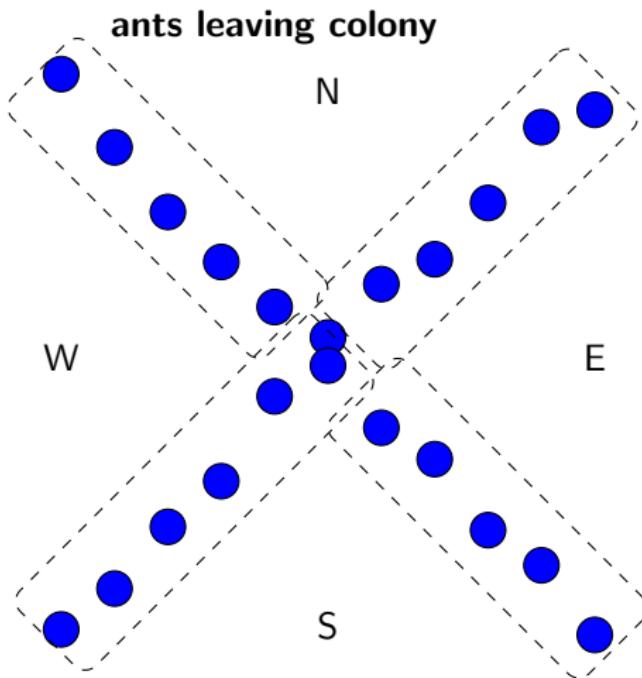
# What forms clusters in your space?

---



# What forms clusters in your space?

---



## There usually are not %100 correct answers

---

- Clustering is partly an art
  - ▶ Clustering de partie est un art
- You need to experiment with different things to get there
  - ▶ On a besoin d'expérimenter afin d'y arriver

# How to cluster

---

- Many approaches. Today we'll talk about
  - ▶ Centroid based approaches
    - ▶ K-means
    - ▶ Mixture Models
  - ▶ Hierarchical clustering
  - ▶ Spectral clustering

# Table of Contents

---

## Centroid based approaches

K-means clustering

Gaussian Mixture Model

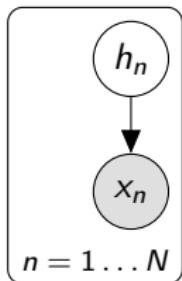
More Advanced GMM Learning Methods

## Spectral Clustering

## Hierarchical Clustering

# Gaussian Mixture Model

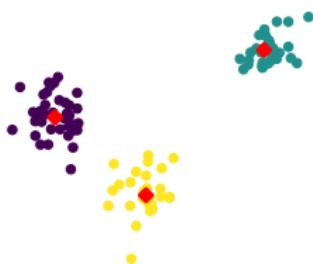
## ■ Model:



$$h_n \sim \text{Discrete}(\pi)$$

$$x_n | h_n \sim \mathcal{N}(x; \mu_{h_n}, \sigma^2 I), \text{ for } n \in \{1, \dots, N\}$$

- $h_n \in \{1, \dots, K\}$ , cluster indicators / indicateur de groupes.
- $x_n \in \mathbb{R}^L$ , observed data items / des données observées.
- $\theta = \{\mu_1, \mu_2, \dots, \mu_K\}$  parameters/cluster centers (or centroids) (les paramètres, centres de groupes).



# Table of Contents

---

Centroid based approaches

K-means clustering

Gaussian Mixture Model

More Advanced GMM Learning Methods

Spectral Clustering

Hierarchical Clustering

# Learning Variant 1 for GMM

---

- Find cluster indicators  $\hat{h}_{1:N}$  and parameters  $\hat{\theta}$  such that: / On trouve des indicateurs de clusters et des centroids telle que:

$$\hat{h}_{1:N}, \hat{\theta} = \arg \max_{h_{1:N}, \theta} p(x_{1:N} | h_{1:N}, \theta)$$

# Learning Variant 1 for GMM

---

- Find cluster indicators  $\hat{h}_{1:N}$  and parameters  $\hat{\theta}$  such that: / On trouve des indicateurs de clusters et des centroids telle que:

$$\hat{h}_{1:N}, \hat{\theta} = \arg \max_{h_{1:N}, \theta} p(x_{1:N} | h_{1:N}, \theta)$$

- Write down log-likelihood: / On écrit le log-likelihood:

$$\begin{aligned}\log p(x_{1:N}, h_{1:N} | \theta) &= \log \prod_{n=1}^N p(x_n | h_n, \theta) p(h_n | \theta) \\ &= \log \prod_{n=1}^N \left( \prod_{k=1}^K \mathcal{N}(x_n; \mu_k, \sigma^2 I)^{[h_n=k]} \times \prod_{k=1}^K \pi_k^{[h_n=k]} \right) \\ &= + \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)\end{aligned}$$

# How to learn with this objective function?

---

$$\mathcal{L}(\mu_{1:K}, \pi_{1:K}, h_{1:N}) = \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)$$

- Notice that  $h_{1:N}$  are discrete variables. / Notez que  $h_{1:N}$  sont discrets.
- We can not directly take the gradient and optimize. / On ne peut pas juste calculer le gradient et l'optimiser.

# How to learn with this objective function?

---

$$\mathcal{L}(\mu_{1:K}, \pi_{1:K}, h_{1:N}) = \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)$$

- Notice that  $h_{1:N}$  are discrete variables. / Notez que  $h_{1:N}$  sont discrets.
- We can not directly take the gradient and optimize. / On ne peut pas juste calculer le gradient et l'optimiser.
- Any ideas? / Idées?

## Learning Variant 1 for GMM

---

- Algorithm: Fix  $\theta$ , update  $h$ . Fix  $h$ , update  $\theta$ , repeat until convergence (and fix  $\pi_k = 1/K$ ). / On alterne entre l'optimization des paramètres  $\theta$  et les indicateurs  $h$ .

# Learning Variant 1 for GMM

- Algorithm: Fix  $\theta$ , update  $h$ . Fix  $h$ , update  $\theta$ , repeat until convergence (and fix  $\pi_k = 1/K$ ). / On alterne entre l'optimization des paramètres  $\theta$  et les indicateurs  $h$ .
- Update  $\mu_{k'}$ : compute the gradient while  $h_{1:N}$  is fixed: / calcule le gradient par rapport à  $\mu_{k'}$  quand  $h_{1:N}$  est fixé.

$$\begin{aligned}\frac{\partial \log p(x_{1:N}, h_{1:N} | \theta)}{\partial \mu_k} &= \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} \\ &= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}\end{aligned}$$

set the gradient equal to 0 / mettez le gradient à 0, solve for / résoudre pour

$$\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n = k'] x_n}{\sum_{n=1}^N [h_n = k']}.$$

# Learning Variant 1 for GMM

- Algorithm: Fix  $\theta$ , update  $h$ . Fix  $h$ , update  $\theta$ , repeat until convergence (and fix  $\pi_k = 1/K$ ). / On alterne entre l'optimization des paramètres  $\theta$  et les indicateurs  $h$ .
- Update  $\mu_{k'}$ : compute the gradient while  $h_{1:N}$  is fixed: / calcule le gradient par rapport à  $\mu_{k'}$  quand  $h_{1:N}$  est fixé.

$$\begin{aligned}\frac{\partial \log p(x_{1:N}, h_{1:N} | \theta)}{\partial \mu_k} &= \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} \\ &= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}\end{aligned}$$

set the gradient equal to 0 / mettez le gradient à 0, solve for / résoudre pour

$$\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n = k'] x_n}{\sum_{n=1}^N [h_n = k']}.$$

- Update  $h_{1:N}$  while  $\mu_{k'}$  is fixed: / Mettez  $h_{1:N}$  à jour quand  $\mu_{k'}$  est fixé:

$$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|_2^2,$$

we therefore assign  $h_n$  as the index of the mean closest to  $x_n$ . / On assigne  $h_n$  au centroid plus proche.

# Learning Variant 1 for GMM

- Algorithm: Fix  $\theta$ , update  $h$ . Fix  $h$ , update  $\theta$ , repeat until convergence (and fix  $\pi_k = 1/K$ ). / On alterne entre l'optimization des paramètres  $\theta$  et les indicateurs  $h$ .
- Update  $\mu_{k'}$ : compute the gradient while  $h_{1:N}$  is fixed: / calcule le gradient par rapport à  $\mu_{k'}$  quand  $h_{1:N}$  est fixé.

$$\begin{aligned}\frac{\partial \log p(x_{1:N}, h_{1:N} | \theta)}{\partial \mu_{k'}} &= \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K [h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} \\ &= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N [h_n = k'] \frac{x_n}{\sigma^2} - [h_n = k'] \frac{\mu_{k'}}{\sigma^2}\end{aligned}$$

set the gradient equal to 0 / mettez le gradient à 0, solve for / résoudre pour

$$\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n = k'] x_n}{\sum_{n=1}^N [h_n = k']}.$$

- Update  $h_{1:N}$  while  $\mu_{k'}$  is fixed: / Mettez  $h_{1:N}$  à jour quand  $\mu_{k'}$  est fixé:

$$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|_2^2,$$

we therefore assign  $h_n$  as the index of the mean closest to  $x_n$ . / On assigne  $h_n$  au centroid plus proche.

- Looks like a familiar algorithm? / Vous connaissez ça?

# Kmeans Clustering

---

Randomly initialize  $\mu_{1:K}$ .

**while** Not converged **do**

**E-step:**

$$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|_2^2$$

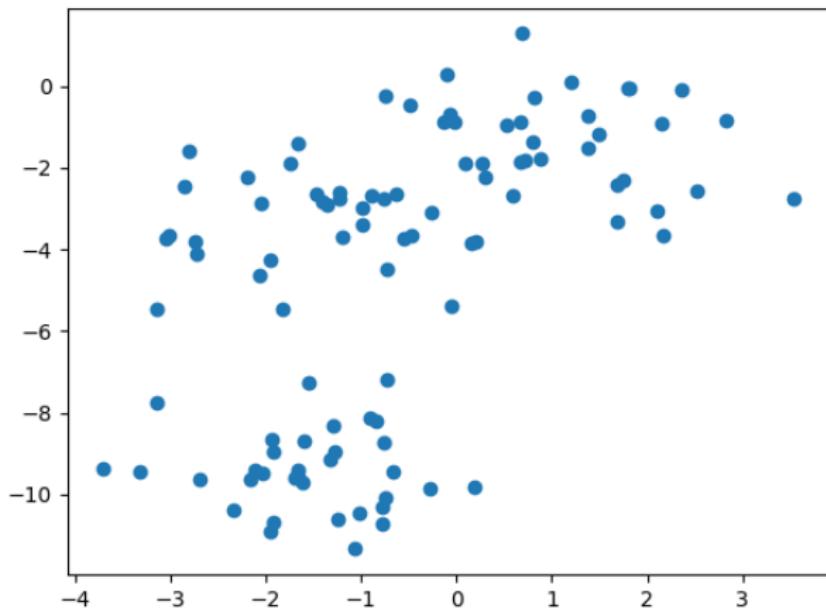
**M-step:**

$$\hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n = k'] x_n}{\sum_{n=1}^N [h_n = k']}$$

**end while**

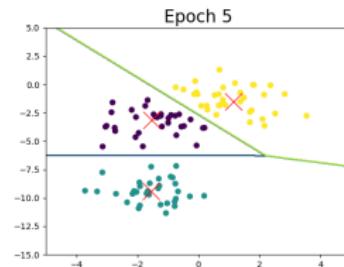
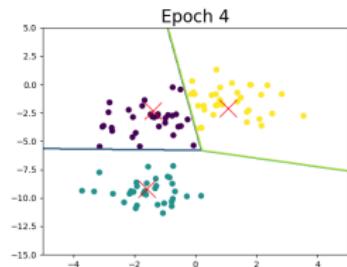
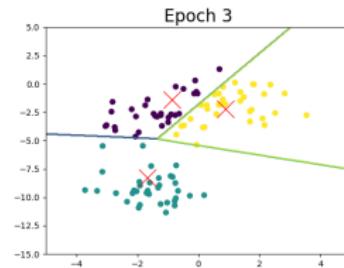
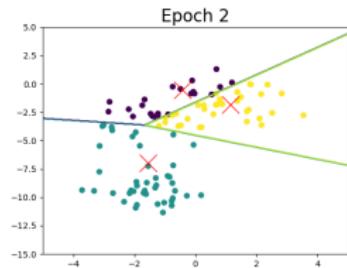
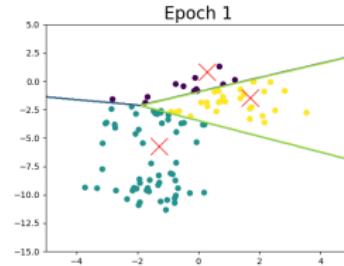
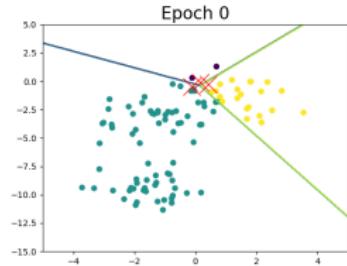
# Kmeans Example

---



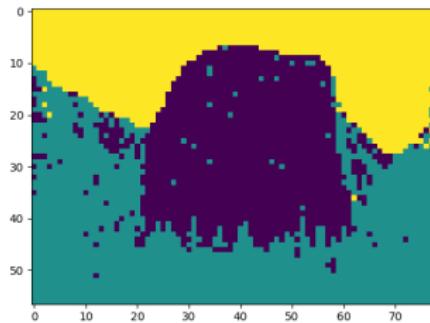
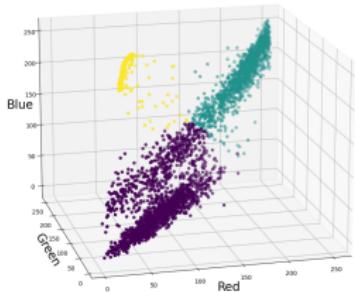
# Kmeans Updates

---



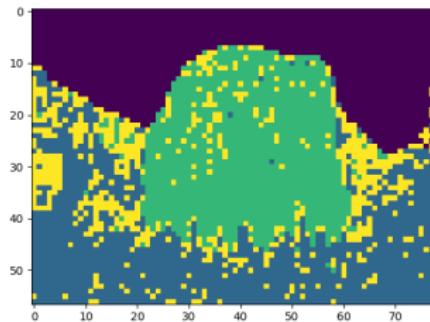
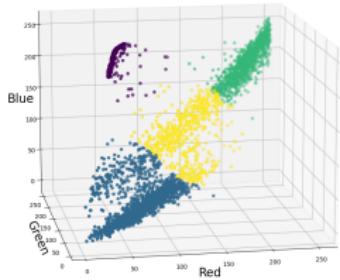
# Applying K-means on El-Capitan ( $K = 3$ )

---



# Applying K-means on El-Capitan ( $K = 4$ )

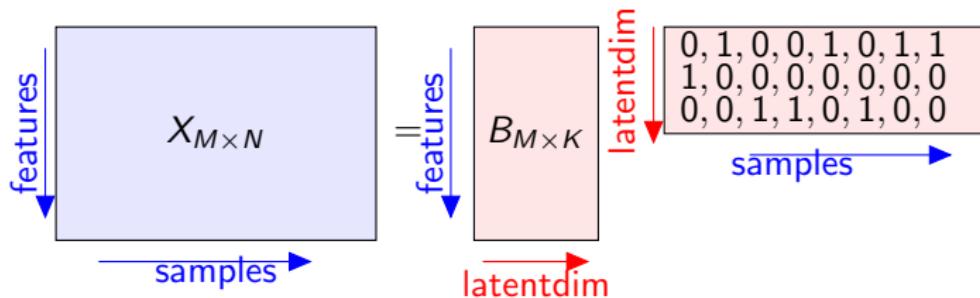
---



## A sidenote

---

- K-means is a matrix-factorization algorithm

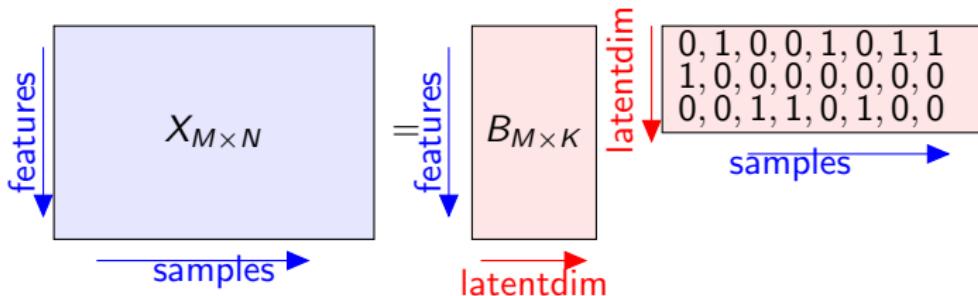


- $B$  in this case has the cluster means in columns /  $B$  dans ce cas-ci a les cluster means dans ses colonnes.

## A sidenote

---

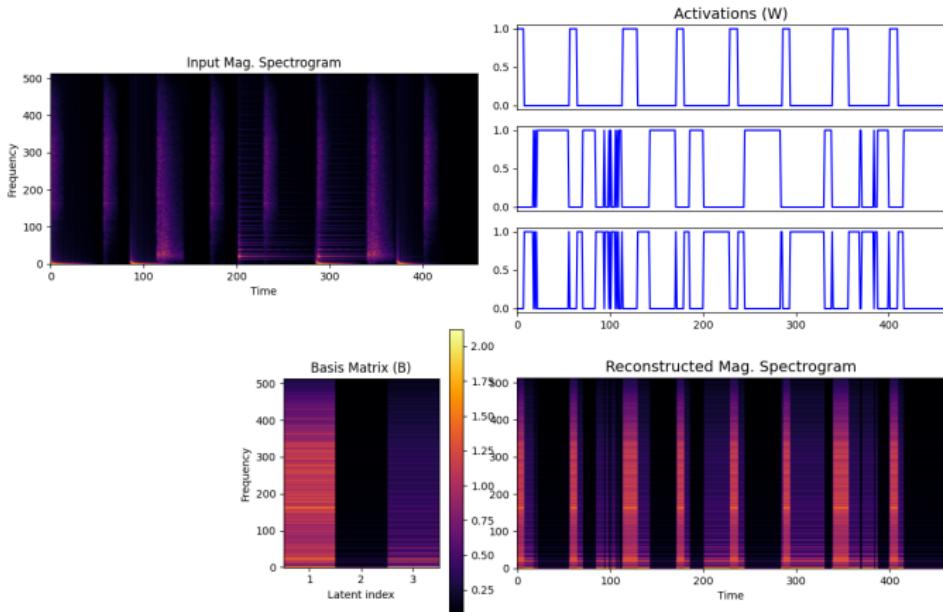
- K-means is a matrix-factorization algorithm



- $B$  in this case has the cluster means in columns /  $B$  dans ce cas-ci a les cluster means dans ses colonnes.
- Kmeans is a tokenizer! (It's a buzzword these days)

# Kmeans on a familiar picture

---



# Table of Contents

---

## Centroid based approaches

K-means clustering

Gaussian Mixture Model

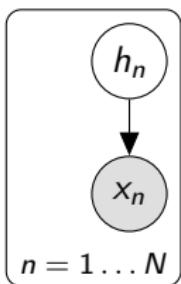
More Advanced GMM Learning Methods

Spectral Clustering

Hierarchical Clustering

# Extending K-means

- Kmeans is good and all, but only learns Isotropic Gaussians. /  
Kmeans ne peut apprendre que des Gaussiennes Isotropiques.

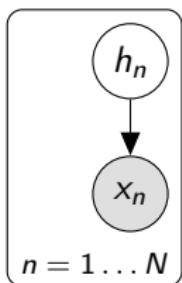


$$h_n \sim \text{Discrete}(\pi)$$
$$x_n | h_n \sim \mathcal{N}(x; \mu_{h_n}, \Sigma_{h_n}), \text{ for } n \in \{1, \dots, N\}$$

- $h_n \in \{1, \dots, K\}$ , cluster indicators / indicateur de groupes.
- $x_n \in \mathbb{R}^L$ , observed data items / des données observées.
- $\theta = \{\mu_1, \mu_2, \dots, \mu_K, \Sigma_1, \Sigma_2, \dots, \Sigma_K, \pi_1, \pi_2, \dots, \pi_K\}$  parameters (les paramètres).

# Extending K-means

- Kmeans is good and all, but only learns Isotropic Gaussians. /  
Kmeans ne peut apprendre que des Gaussiennes Isotropiques.



$$h_n \sim \text{Discrete}(\pi)$$
$$x_n | h_n \sim \mathcal{N}(x; \mu_{h_n}, \Sigma_{h_n}), \text{ for } n \in \{1, \dots, N\}$$

- $h_n \in \{1, \dots, K\}$ , cluster indicators / indicateur de groupes.
- $x_n \in \mathbb{R}^L$ , observed data items / des données observées.
- $\theta = \{\mu_1, \mu_2, \dots, \mu_K, \Sigma_1, \Sigma_2, \dots, \Sigma_K, \pi_1, \pi_2, \dots, \pi_K\}$  parameters (les paramètres).
- Full Covariance matrix for each cluster, and cluster prior probabilities are estimated also.
  - Une matrice de covariance est appris pour chaque cluster, puis les probabilités a prioris de chaque cluster.

## Learning Variant 2 for GMM

---

- In addition, we want to estimate probabilities for  $h_{1:N}$ . / On estimer des probabilités pour des indicateurs des clusters.
- Find cluster indicator parameters  $\hat{\theta}$  while integrating out hidden variables, such that: / On marginalise sur  $h_{1:N}$

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} p(x_{1:N} | \theta) \\ &= \arg \max_{\theta} \sum_{h_{1:N}} p(x_{1:N}, h_{1:N} | \theta)\end{aligned}$$

## Learning Variant 2 for GMM

---

- In addition, we want to estimate probabilities for  $h_{1:N}$ . / On estimer des probabilités pour des indicateurs des clusters.
- Find cluster indicator parameters  $\hat{\theta}$  while integrating out hidden variables, such that: / On marginalise sur  $h_{1:N}$

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} p(x_{1:N} | \theta) \\ &= \arg \max_{\theta} \sum_{h_{1:N}} p(x_{1:N}, h_{1:N} | \theta)\end{aligned}$$

- Write down/Écrit log-likelihood:

$$\log p(x_{1:N} | \theta) = \log \sum_{h_{1:N}} \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} q(h_{1:N}) = \log \mathbb{E}_q \left[ \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right]$$

## Learning Variant 2 for GMM

- In addition, we want to estimate probabilities for  $h_{1:N}$ . / On estime des probabilités pour des indicateurs des clusters.
- Find cluster indicator parameters  $\hat{\theta}$  while integrating out hidden variables, such that: / On marginalise sur  $h_{1:N}$

$$\hat{\theta} = \arg \max_{\theta} p(x_{1:N} | \theta)$$

$$= \arg \max_{\theta} \sum_{h_{1:N}} p(x_{1:N}, h_{1:N} | \theta)$$

- Write down/Écrit log-likelihood:

$$\begin{aligned}\log p(x_{1:N} | \theta) &= \log \sum_{h_{1:N}} \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} q(h_{1:N}) = \log \mathbb{E}_q \left[ \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] \\ &\geq VLB := \mathbb{E}_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] =^+ \mathbb{E}_q [\log p(x_{1:N}, h_{1:N} | \theta)] \\ &=^+ \sum_{n=1}^N \left( \sum_{k=1}^K \mathbb{E}_q[h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)\end{aligned}$$

## Learning Variant 2 for GMM

---

- Algorithm: Fix  $\theta$ , update  $q$ . Fix  $q$ , update  $\theta$ , repeat until convergence. / On alterne entre l'optimization de  $q$  et  $\theta$ .

## Learning Variant 2 for GMM

---

- Algorithm: Fix  $\theta$ , update  $q$ . Fix  $q$ , update  $\theta$ , repeat until convergence. / On alterne entre l'optimization de  $q$  et  $\theta$ .
- Update  $\mu_{k'}$ : compute the gradient while  $q(h_{1:N})$  is fixed:

$$\begin{aligned}\frac{\partial VLB}{\partial \mu_{k'}} &= \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K \mathbb{E}[h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} \\ &= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N \mathbb{E}[h_n = k'] \frac{x_n}{\sigma^2} - \mathbb{E}[h_n = k'] \frac{\mu_{k'}}{\sigma^2}\end{aligned}$$

set the gradient equal to 0, solve for  $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N \mathbb{E}[h_n=k'] x_n}{\sum_{n=1}^N \mathbb{E}[h_n=k']}$ .

## Learning Variant 2 for GMM

---

- Algorithm: Fix  $\theta$ , update  $q$ . Fix  $q$ , update  $\theta$ , repeat until convergence. / On alterne entre l'optimization de  $q$  et  $\theta$ .
- Update  $\mu_{k'}$ : compute the gradient while  $q(h_{1:N})$  is fixed:

$$\begin{aligned}\frac{\partial VLB}{\partial \mu_{k'}} &= \frac{\partial \sum_{n=1}^N \left( \sum_{k=1}^K \mathbb{E}[h_n = k] \left( \frac{-\|x_n - \mu_k\|_2^2}{2\sigma^2} + \log \pi_k \right) \right)}{\partial \mu_{k'}} \\ &= \sum_{n=1}^N [h_n = k'] \frac{(x_n - \mu_{k'})}{\sigma^2} = \sum_{n=1}^N \mathbb{E}[h_n = k'] \frac{x_n}{\sigma^2} - \mathbb{E}[h_n = k'] \frac{\mu_{k'}}{\sigma^2}\end{aligned}$$

set the gradient equal to 0, solve for  $\mu_{k'} \rightarrow \hat{\mu}_{k'} = \frac{\sum_{n=1}^N \mathbb{E}[h_n=k'] x_n}{\sum_{n=1}^N \mathbb{E}[h_n=k']}$ .

- Update  $\hat{\sigma}_{k'} = \frac{\sum_{n=1}^N \mathbb{E}[h_n=k'] (x_n - \mu_k)^2}{\sum_{n=1}^N \mathbb{E}[h_n=k']}$ .
- Update  $\hat{\pi}_{k'} = \frac{1}{N} \sum_{n=1}^N \mathbb{E}[h_n = k']$ .

## Learning Variant 2 for GMM - optimal $q(h)$

---

- Update  $q(h_{1:N})$  while  $\theta$  is fixed. Notice that:

$$VLB = \mathbb{E}_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] = KL(q(h) \| p(x, h | \theta)).$$

What is the variational distribution that would minimize this divergence?

## Learning Variant 2 for GMM - optimal $q(h)$

---

- Update  $q(h_{1:N})$  while  $\theta$  is fixed. Notice that:

$$VLB = \mathbb{E}_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] = KL(q(h) \| p(x, h | \theta)).$$

What is the variational distribution that would minimize this divergence?

- The derivation.

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial}{\partial q} \left( \int q(h) \log p(x, h | \theta) dh - \int q(h) \log q(h) dh + \lambda \left( \int q(h) dh - 1 \right) \right)$$

$$= \log p(x, h) - \log q(h) - 1 + \lambda = 0$$

$$\rightarrow q(h) = \frac{p(x, h | \theta)}{\exp(1 - \lambda)}$$

$$\rightarrow \exp(1 - \lambda) = p(x | \theta)$$

$$\rightarrow q(h) = \frac{p(x, h | \theta)}{p(x | \theta)} = p(h | x, \theta)$$

## Learning Variant 2 for GMM - optimal $q(h)$

---

- Update  $q(h_{1:N})$  while  $\theta$  is fixed. Notice that:

$$VLB = \mathbb{E}_q \left[ \log \frac{p(x_{1:N}, h_{1:N} | \theta)}{q(h_{1:N})} \right] = KL(q(h) \| p(x, h | \theta)).$$

What is the variational distribution that would minimize this divergence?

- The derivation.

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial}{\partial q} \left( \int q(h) \log p(x, h | \theta) dh - \int q(h) \log q(h) dh + \lambda \left( \int q(h) dh - 1 \right) \right)$$

$$= \log p(x, h) - \log q(h) - 1 + \lambda = 0$$

$$\rightarrow q(h) = \frac{p(x, h | \theta)}{\exp(1 - \lambda)}$$

$$\rightarrow \exp(1 - \lambda) = p(x | \theta)$$

$$\rightarrow q(h) = \frac{p(x, h | \theta)}{p(x | \theta)} = p(h | x, \theta)$$

- Note that in our case  $q(h) = q(h_{1:N}) = \prod_n q(h_n)$ , where

$$q(h_n = k) = \frac{p(x_n, h_n = k | \theta)}{p(x_n | \theta)} = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \sigma^2 I)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_n; \mu_{k'}, \sigma^2 I)}$$

# Learning Variant 2 for GMM - Summary for ICM and EM

---

Randomly initialize  $\mu_{1:K}$ .

**while** Not converged **do**

**E-step:**

**if** ICM **then**

$$\hat{h}_n = \arg \max_{h_n} \log p(x_n, h_n | \theta) = \arg \min_k \|x_n - \mu_k\|_2^2$$

**else if** EM **then**

$$q(h_n = k) = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \sigma^2 I)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_n; \mu_{k'}, \sigma^2 I)}$$

**end if**

**M-step:**

**if** ICM **then**

$$\hat{\mu}_{k'} = \frac{\sum_{n=1}^N [h_n=k'] x_n}{\sum_{n=1}^N [h_n=k']}$$

**else if** EM **then**

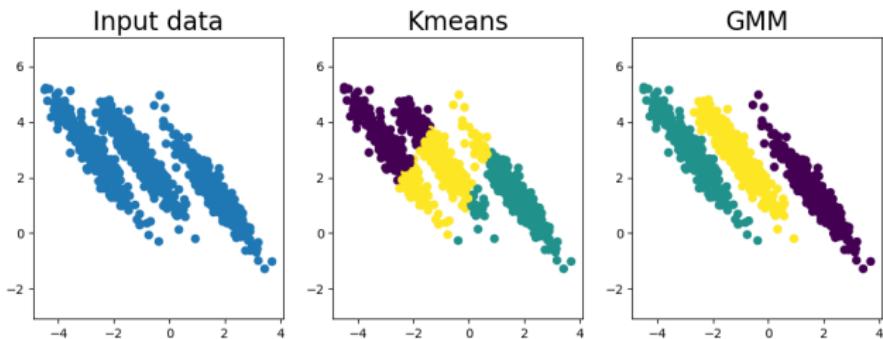
$$\hat{\mu}_{k'} = \frac{\sum_{n=1}^N \mathbb{E}_q[h_n=k'] x_n}{\sum_{n=1}^N \mathbb{E}_q[h_n=k']}$$

**end if**

**end while**

# Kmeans vs GMM

---

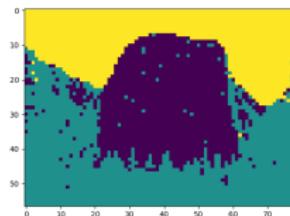
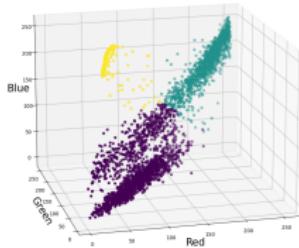


# GMM on El Capitan

---

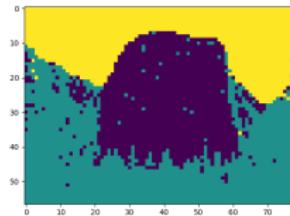
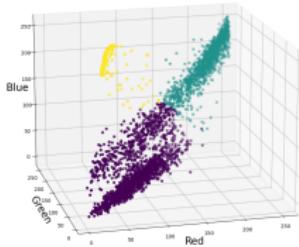


With Kmeans

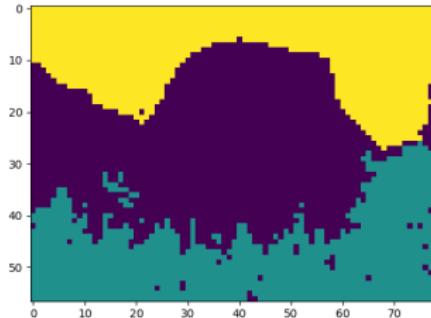
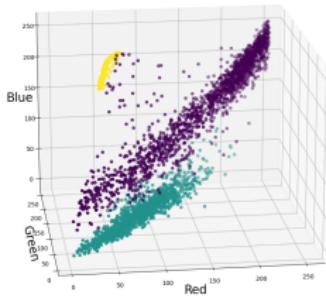


# GMM on El Capitan

With Kmeans



With GMM



# Table of Contents

---

## Centroid based approaches

K-means clustering

Gaussian Mixture Model

More Advanced GMM Learning Methods

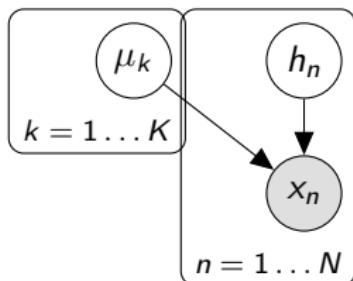
Spectral Clustering

Hierarchical Clustering

# Learning Variant 3 for GMM - Going Full Bayesian

---

## ■ Model:



$$\mu_k \sim \mathcal{N}(\mu_k; 0, \sigma_0^2 I), \text{ for } k \in \{1, \dots, K\}$$

$$h_n \sim \text{Categorical}(\pi)$$

$$x_n | h_n \sim \mathcal{N}(x; \mu_h, \sigma^2 I), \text{ for } n \in \{1, \dots, N\}$$

- $h_n \in \{1, \dots, K\}$ , cluster indicators / indicateur des clusters.
- $x_n \in \mathbb{R}^L$ , observed data items / données observées.
- $\theta = \{\mu_1, \mu_2, \dots, \mu_K\}$  parameters/cluster centers. But we are not treating these guys as parameters anymore. / On ne traite plus ça comme des paramètres mais des distributions.

## Inference for Variant 3 GMM

---

- Approximate the posterior distribution  $p(h, \theta|x)$ , with a variational distribution  $\hat{q}$  such that, / On va approximer la posterior telle que,

$$\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta) \| p(x, h, \theta))$$

- We will use the mean field approximation. English:  
 $q(h, \theta) = q_h(h)q_\theta(\theta)$ .

## Inference for Variant 3 GMM

---

- Approximate the posterior distribution  $p(h, \theta|x)$ , with a variational distribution  $\hat{q}$  such that, / On va approximer la posterior telle que,

$$\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta) \| p(x, h, \theta))$$

- We will use the mean field approximation. English:  
 $q(h, \theta) = q_h(h)q_\theta(\theta)$ .
- Algorithm: Fix  $q_h$ , update  $q_\theta$ . We can show that: (via same process as the EM case)

$$\hat{q}_\theta(\theta) = \arg \min_{q_\theta} KL(q_h(h)q_\theta(\theta) \| p(x, h, \theta)) = \frac{1}{Z} \exp (\mathbb{E}_{q_h} [\log p(x, h, \theta)])$$

where  $Z$  is the normalization constant (constant de normalization).  
Similarly,

## Inference for Variant 3 GMM

---

- Approximate the posterior distribution  $p(h, \theta|x)$ , with a variational distribution  $\hat{q}$  such that, / On va approximer la posterior telle que,

$$\hat{q}(h, \theta) = \arg \min_q KL(q(h, \theta) \| p(x, h, \theta))$$

- We will use the mean field approximation. English:  
 $q(h, \theta) = q_h(h)q_\theta(\theta)$ .
- Algorithm: Fix  $q_h$ , update  $q_\theta$ . We can show that: (via same process as the EM case)

$$\hat{q}_\theta(\theta) = \arg \min_{q_\theta} KL(q_h(h)q_\theta(\theta) \| p(x, h, \theta)) = \frac{1}{Z} \exp (\mathbb{E}_{q_h} [\log p(x, h, \theta)])$$

where  $Z$  is the normalization constant (constant de normalization).  
Similarly,

- Fix  $q_\theta$ , update  $q_h$ :

$$\hat{q}_h(h) = \arg \min_{q_h} KL(q_h(h)q_\theta(\theta) \| p(x, h, \theta)) = \frac{1}{Z} \exp (\mathbb{E}_{q_\theta} [\log p(x, h, \theta)])$$

## Inference for Variant 3 GMM - Specifics:

---

$$\begin{aligned}\log \hat{q}_\theta(\mu_k) &= {}^+\mathbb{E}_{q_h}[\log p(x, h, \mu_k)] \\ &= {}^+ \sum_{n=1}^N \mathbb{E}[h_n = k] \frac{-(x_n^\top x_n - 2x_n^\top \mu_k + \mu_k^\top \mu_k)}{2\sigma^2} - \frac{\mu_k^\top \mu_k}{2\sigma_0^2} \\ &= {}^+ \frac{\sum_{n=1}^N \mathbb{E}[h_n = k] 2x_n^\top \mu_k - (\sum_{n=1}^N \mathbb{E}[h_n = k] + \sigma^2) \mu_k^\top \mu_k}{2\sigma^2 \sigma_0^2} \\ &= {}^+ \log \mathcal{N} \left( \mu_k; \frac{\sum_n \mathbb{E}[h_n = k] x_n}{\sum_n \mathbb{E}[h_n = k] + \sigma^2}, \frac{\sigma^2 \sigma_0^2}{\sum_n \mathbb{E}[h_n = k] + \sigma^2} \right)\end{aligned}$$

## Inference for Variant 3 GMM - Specifics:

---

$$\begin{aligned}\log \hat{q}_\theta(\mu_k) &= {}^+\mathbb{E}_{q_h}[\log p(x, h, \mu_k)] \\ &= {}^+ \sum_{n=1}^N \mathbb{E}[h_n = k] \frac{-(x_n^\top x_n - 2x_n^\top \mu_k + \mu_k^\top \mu_k)}{2\sigma^2} - \frac{\mu_k^\top \mu_k}{2\sigma_0^2} \\ &= {}^+ \frac{\sum_{n=1}^N \mathbb{E}[h_n = k] 2x_n^\top \mu_k - (\sum_{n=1}^N \mathbb{E}[h_n = k] + \sigma^2) \mu_k^\top \mu_k}{2\sigma^2 \sigma_0^2} \\ &= {}^+ \log \mathcal{N} \left( \mu_k; \frac{\sum_n \mathbb{E}[h_n = k] x_n}{\sum_n \mathbb{E}[h_n = k] + \sigma^2}, \frac{\sigma^2 \sigma_0^2}{\sum_n \mathbb{E}[h_n = k] + \sigma^2} \right)\end{aligned}$$

## Inference for Variant 3 GMM - Specifics:

---

$$\log \hat{q}_h(h_n = k) = \left( \frac{\mathbb{E}[-\|x_n - \mu_k\|_2^2]}{2\sigma^2} + \log \pi_k \right)$$
$$\rightarrow \hat{q}_h(h_n = k) = \frac{\exp \left( \frac{\mathbb{E}[-\|x_n - \mu_k\|_2^2]}{2\sigma^2} + \log \pi_k \right)}{\sum_k \exp \left( \frac{\mathbb{E}[-\|x_n - \mu_k\|_2^2]}{2\sigma^2} + \log \pi_k \right)}$$

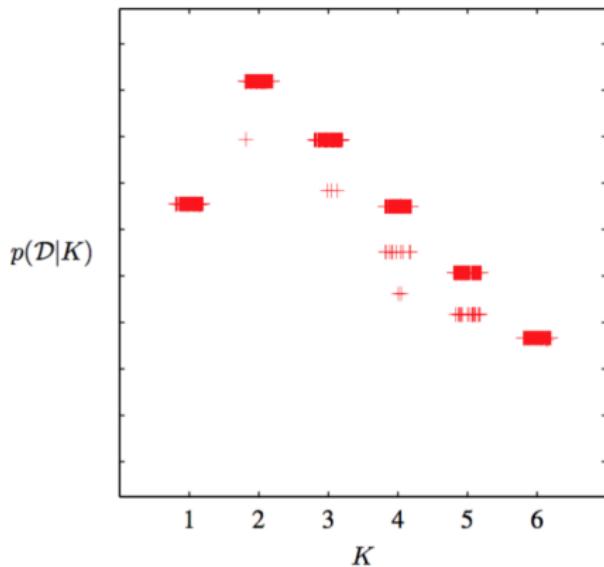
# Inference for Variant 3 GMM - Why:

- Variational lower bound:

$$\int p(x, h, \theta) dh d\theta \geq \mathbb{E}_{q(h)q(\theta)}[\log p(x, h, \theta)] - \mathbb{E}_{q(h)q(\theta)}[\log q(h) + \log q(\theta)]$$

- You can use VLB to determine  $K$ : (plot taken from Bishop, 2006)

Plot of the variational lower bound  $\mathcal{L}$  versus the number  $K$  of components in the Gaussian mixture model, for the Old Faithful data, showing a distinct peak at  $K = 2$  components. For each value of  $K$ , the model is trained from 100 different random starts, and the results shown as '+' symbols plotted with small random horizontal perturbations so that they can be distinguished. Note that some solutions find suboptimal local maxima, but that this happens infrequently.



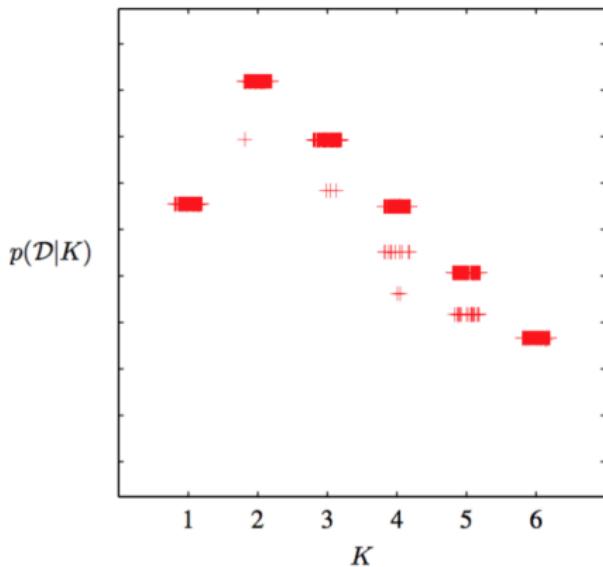
## Inference for Variant 3 GMM - Why:

- Variational lower bound:

$$\int p(x, h, \theta) dh d\theta \geq \mathbb{E}_{q(h)q(\theta)}[\log p(x, h, \theta)] - \mathbb{E}_{q(h)q(\theta)}[\log q(h) + \log q(\theta)]$$

- You can use VLB to determine  $K$ : (plot taken from Bishop, 2006)

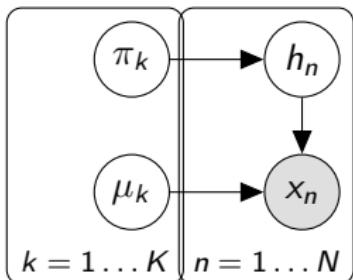
Plot of the variational lower bound  $\mathcal{L}$  versus the number  $K$  of components in the Gaussian mixture model, for the Old Faithful data, showing a distinct peak at  $K = 2$  components. For each value of  $K$ , the model is trained from 100 different random starts, and the results shown as '+' symbols plotted with small random horizontal perturbations so that they can be distinguished. Note that some solutions find suboptimal local maxima, but that this happens infrequently.



- But admittedly the algebra gets tiring./C'est évident que les

## Variant 4 for GMM - Going Ultra Bayesian

- Model:



$$\pi \sim \text{Dirichlet}(1/K, \dots, 1/K)$$

$$\mu_k \sim \mathcal{N}(\mu_k; 0, \sigma_0^2 I), \text{ for } k \in \{1, \dots, K\}$$

$$h_n \sim \text{Categorical}(\pi)$$

$$x_n | h_n \sim \mathcal{N}(x; \mu_h, \sigma^2 I), \text{ for } n \in \{1, \dots, N\}$$

- $h_n \in \{1, \dots, K\}$ , cluster indicators / indicateurs de clusters.
- $x_n \in \mathbb{R}^L$ , observed data items / données.
- $\theta = \{\mu_1, \mu_2, \dots, \mu_K\} \cup \{\pi\}$

## Variant 4 for GMM - Infinite Mixture Model

---

- Integrate out the parameters, sample from the full conditionals / On va éliminer les paramètres, et échantillonner en utilisant les full conditionals:

$$\begin{aligned} p(h_n = k | h_{-n}, x_{1:N}) &\propto \int p(x_{1:N}, h_{1:N}, \pi, \mu_{1:K}) d\mu_{1:K} d\pi \\ &\propto \frac{\alpha/K + N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\}) \end{aligned}$$

- And, sample from these full conditionals!

## Variant 4 for GMM - Infinite Mixture Model

---

- Integrate out the parameters, sample from the full conditionals / On va éliminer les paramètres, et échantillonner en utilisant les full conditionals:

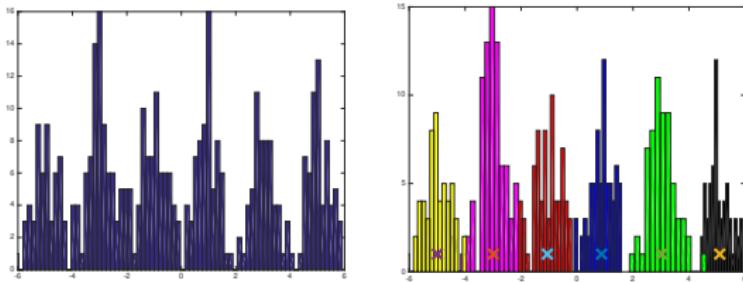
$$\begin{aligned} p(h_n = k | h_{-n}, x_{1:N}) &\propto \int p(x_{1:N}, h_{1:N}, \pi, \mu_{1:K}) d\mu_{1:K} d\pi \\ &\propto \frac{\alpha/K + N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\}) \end{aligned}$$

- Take  $K$  to infinity:

$$\begin{aligned} p(h_n = k, k \text{ occupied} | h_{-n}, x_{1:N}) &\propto \frac{N_k^{-n}}{\alpha + N - 1} p(x_n | \{x_m : m \neq n, h_m = k\}) \\ p(h_n = k, k \text{ empty} | h_{-n}, x_{1:N}) &\propto \frac{\alpha}{\alpha + N - 1} p(x_n) \end{aligned}$$

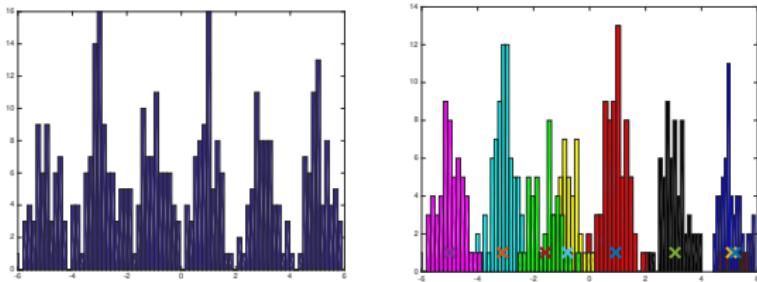
- And, sample from these full conditionals!

# Collapsed Gibbs sampling in Infinite GMM



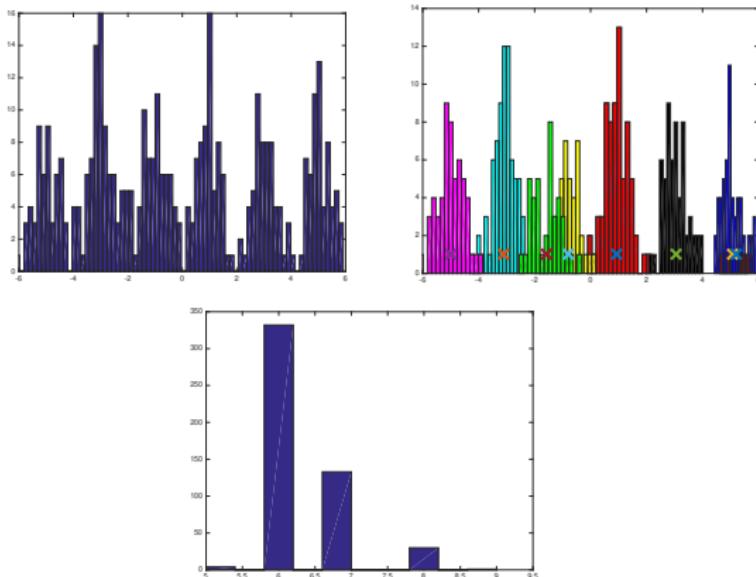
Top left: Histogram of observed data (données observées), Top right: Samples from full conditional of  $h_{1:N}$ , Bottom: Histogram of  $K$  (nombre de clusters)

# Collapsed Gibbs sampling in Infinite GMM



Top left: Histogram of observed data (données observées), Top right: Samples from full conditional of  $h_{1:N}$ , Bottom: Histogram of  $K$  (nombre de clusters)

# Collapsed Gibbs sampling in Infinite GMM



Top left: Histogram of observed data (données observées), Top right: Samples from full conditional of  $h_{1:N}$ , Bottom: Histogram of  $K$  (nombre de clusters)

## What's the point of going all Bayesian then

---

- (Automatic) Model Selection for Unsupervised Learning / Sélection automatique du complexité du modèle

## What's the point of going all Bayesian then

---

- (Automatic) Model Selection for Unsupervised Learning / Sélection automatique du complexité du modèle
- Model Averaging (Model plays all its cards) / On calcule un moyenne sur les modèles

## What's the point of going all Bayesian then

---

- (Automatic) Model Selection for Unsupervised Learning / Sélection automatique du complexité du modèle
- Model Averaging (Model plays all its cards) / On calcule un moyenne sur les modèles
- Principled way of regularization / Un moyenne de regularization

# What's the point of going all Bayesian then

---

- (Automatic) Model Selection for Unsupervised Learning / Sélection automatique du complexité du modèle
- Model Averaging (Model plays all its cards) / On calcule un moyenne sur les modèles
- Principled way of regularization / Un moyenne de regularization
- All of these 4 variants are extendable for other models. We can play with: / Ces 4 idées sont extremement puissants, on peut capturer pleins de modèles.
  - ▶ Distribution of  $h$ .
  - ▶ Impose structure on  $h$ . / Imposition d'une structure sur  $h$  (semaine prochaine)
  - ▶ We can change the conditional distribution  $p(x|h, \theta)$ . (Application decides) / Dépend sur l'output.
  - ▶ We can play with how we do inference and learning. / On peut changer comment on fait apprentissage.

# Table of Contents

---

Centroid based approaches

K-means clustering

Gaussian Mixture Model

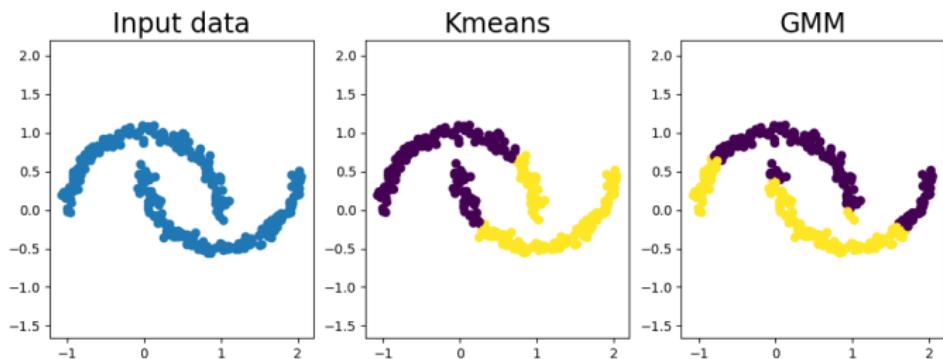
More Advanced GMM Learning Methods

Spectral Clustering

Hierarchical Clustering

# Failure case for Centroid Based Methods

- What if we have something like this? / Et si on avait qqch comme ça?



- Any ideas / Idées?

## Remember KPCA?

---

- Do you remember KPCA? / Vous souvenez-vous de KPCA?
- Let's calculate a pairwise distance matrix

# Spectral Clustering

---

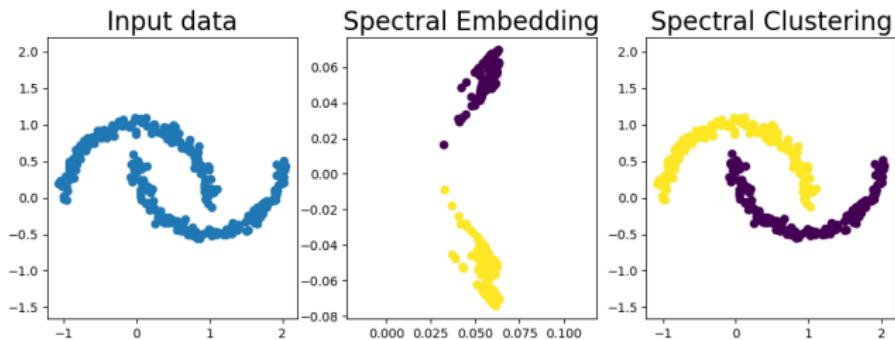
- $A_{ij} = \exp(-\gamma \|x_i - x_j\|_2^2)$ ,  $i \neq j$ .  $A_{ij} = 0$   $i = j$ .
- Compute Graph Laplacian

$$L := D^{-1/2} A D^{-1/2}, \quad D_{ii} = \sum_j A_{ij}$$

- K-means cluster the first k-eigenvectors of L.

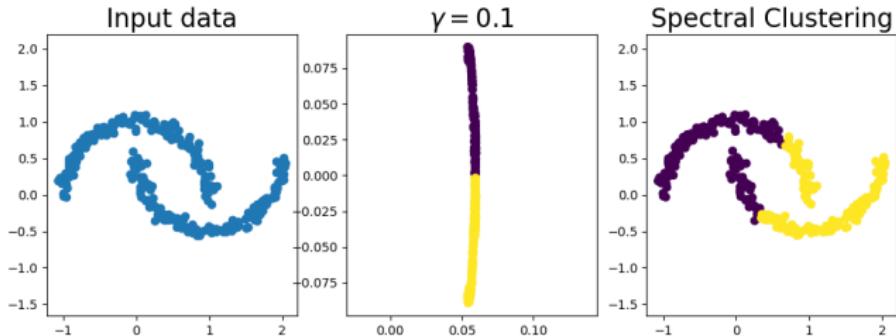
# Spectral Clustering in Action

---



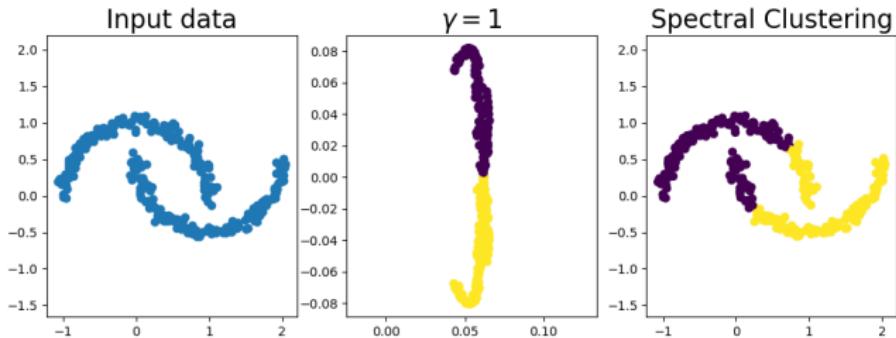
# Spectral Clustering, effect of Gamma

---



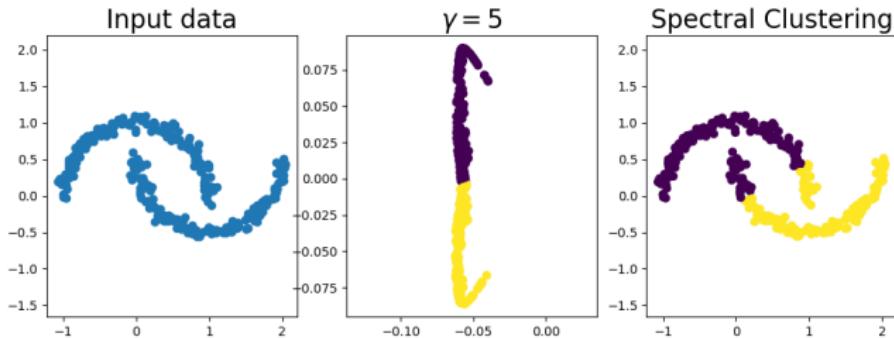
# Spectral Clustering, effect of Gamma

---



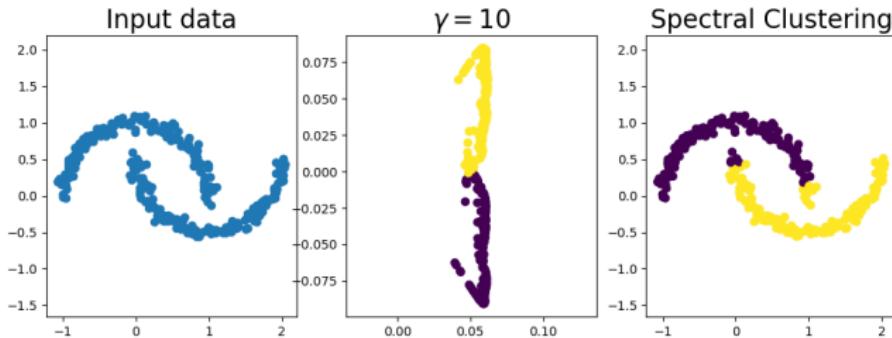
# Spectral Clustering, effect of Gamma

---



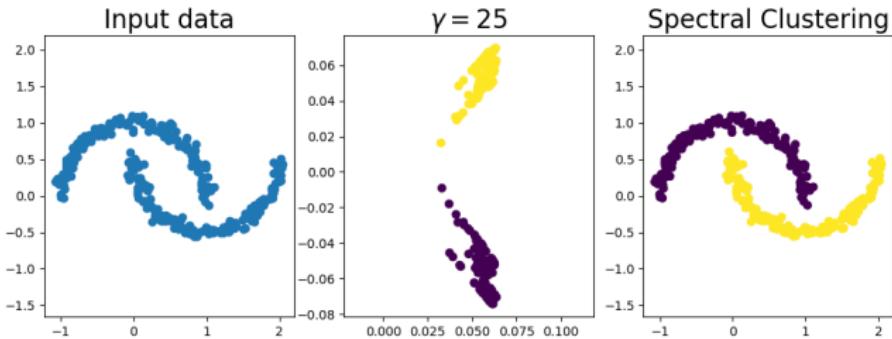
# Spectral Clustering, effect of Gamma

---



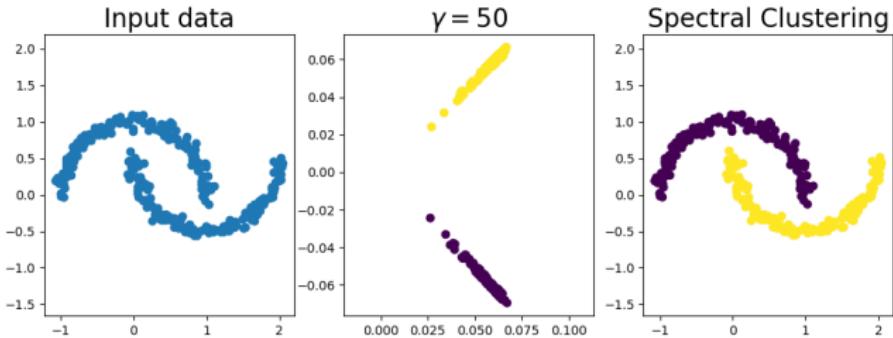
# Spectral Clustering, effect of Gamma

---



# Spectral Clustering, effect of Gamma

---



# Table of Contents

---

Centroid based approaches

K-means clustering

Gaussian Mixture Model

More Advanced GMM Learning Methods

Spectral Clustering

Hierarchical Clustering

# Agglomerative Clustering

---

- Choose  $R_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ . We start from each data item being a cluster. / On commence avec chaque data étant un cluster.

# Agglomerative Clustering

---

- Choose  $R_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ . We start from each data item being a cluster. / On commence avec chaque data étant un cluster.
- For  $t = 1, \dots$

# Agglomerative Clustering

---

- Choose  $R_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ . We start from each data item being a cluster. / On commence avec chaque data étant un cluster.
- For  $t = 1, \dots$ 
  - ▶ Among all clusters in / Parmi tous les clusters dans  $R_{t-1}$ , find cluster pair / on trouve un paire  $\{C_i, C_j\}$  such that / telle que,

$$i, j = \arg \min_{i', j'} d(C_{i'}, C_{j'})$$

.

# Agglomerative Clustering

---

- Choose  $R_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ . We start from each data item being a cluster. / On commence avec chaque data étant un cluster.
- For  $t = 1, \dots$ 
  - ▶ Among all clusters in / Parmi tous les clusters dans  $R_{t-1}$ , find cluster pair / on trouve un paire  $\{C_i, C_j\}$  such that / telle que,

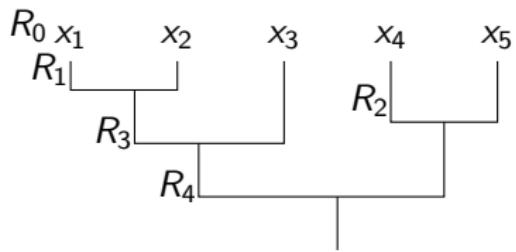
$$i, j = \arg \min_{i', j'} d(C_{i'}, C_{j'})$$

- ▶ Form new cluster pair, remove the pair / Forme un nouvelle paire de clusters, enlève l'ancien,

$$C_q = C_i \cup C_j, R_t = (R_{t-1} - C_i \cup C_j) \cup C_q$$

# Dendrogram

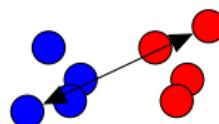
---



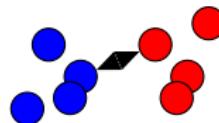
# How do we measure distances between clusters

---

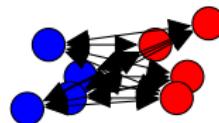
- Linkage functions
  - ▶ Complete linkage



- ▶ Single linkage

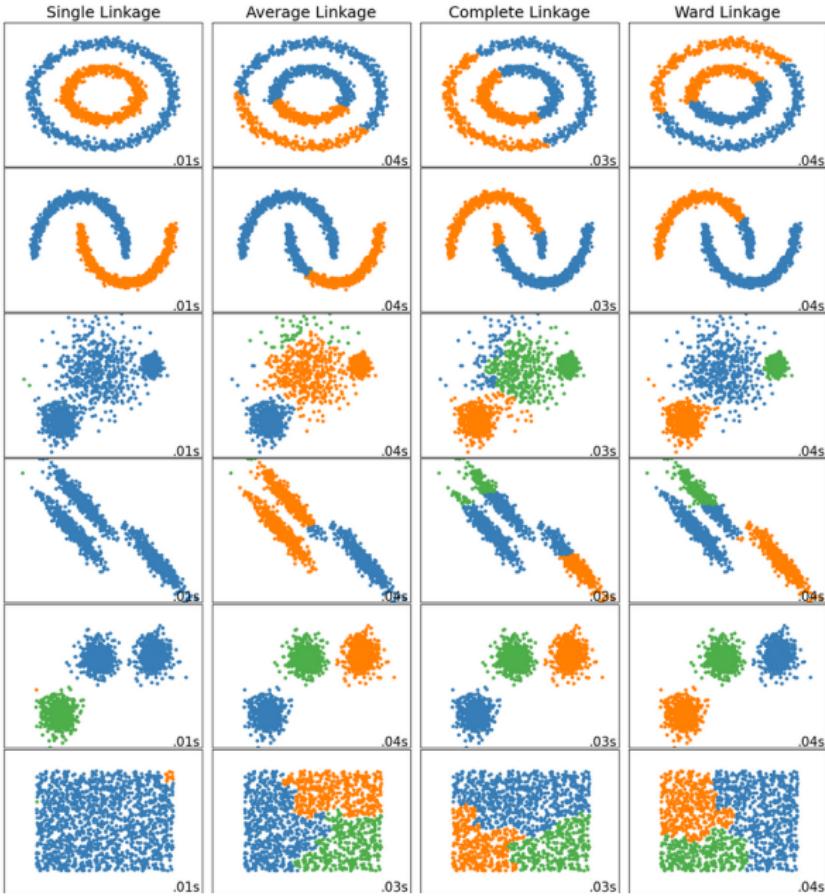


- ▶ Average linkage



- There's more / Il y en a d'autres.

# How does linkage change things



# Computational Complexity

---

- At level  $t$  we have  $N - t$  clusters / Au niveau  $t$ , on a  $N - t$  clusters.
- How many pairs of clusters do we need to deal with? Combien de pairs?

# Computational Complexity

---

- At level  $t$  we have  $N - t$  clusters / Au niveau  $t$ , on a  $N - t$  clusters.
- How many pairs of clusters do we need to deal with? Combien de pairs?
  - ▶  $\binom{N-t}{2} = \frac{(N-t)(N-t-1)}{2}$ .
- In total / Au totale:
  - ▶  $\sum_{t=0}^{N-1} \binom{N-t}{2} = \frac{(N-1)N(N+1)}{6}$

# Computational Complexity

---

- At level  $t$  we have  $N - t$  clusters / Au niveau  $t$ , on a  $N - t$  clusters.
- How many pairs of clusters do we need to deal with? Combien de pairs?
  - ▶  $\binom{N-t}{2} = \frac{(N-t)(N-t-1)}{2}$ .
- In total / Au totale:
  - ▶  $\sum_{t=0}^{N-1} \binom{N-t}{2} = \frac{(N-1)N(N+1)}{6}$
- So  $N^3$ . In the original El-Capitan Image we had  $N \approx 500000$ . That would give something like  $2.083 \times 10^{16}$ . / Dans l'image originale El-Capitan on avait  $N \approx 500000$ . Donc.. merci mais non merci..

# Divisive Clustering

---

- We start with the whole dataset forming a giant cluster. / On commence avec une seule cluster.
- Then we divide by picking out the least similar clusters / Puis on divise en choisissant les cluster les moins similaires.
- They say that divisive usually works better than agglomerative since it sees the global picture better.
  - ▶ C'est accepté que la divisive travaille mieux car il voit le paysage globale mieux.
- Agglomerative is typically faster / Agglomerative est typiquement plus rapide.

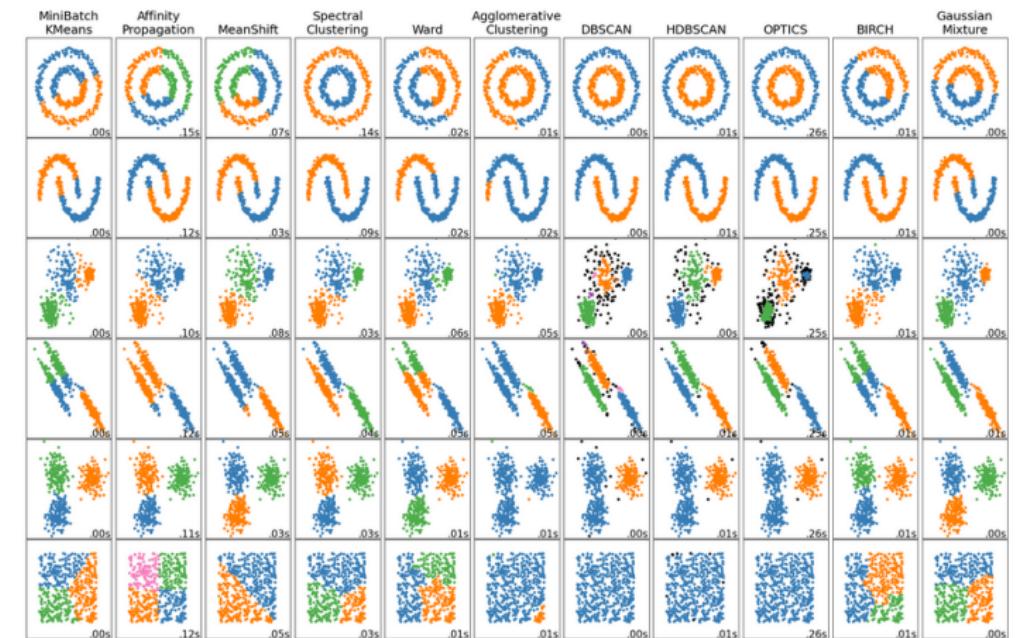
# Clustering recap

---

- Kmeans
  - ▶ Reliable (Fiable), quick and dirty
- GMMs
  - ▶ More powerful than Kmeans but still a centroid method in heart / Plus puissant que Kmeans mais une méthode de centroids si on y pense.
- Spectral Clustering
  - ▶ The non-linear get around to find manifold-like clusters / Une cheminement alternative pour trouver des clusters qui sont comme des manifolds. Not suitable for large datasets / N'est pas approprié pour des grands datasets.
- Hierarchical Clustering
  - ▶ Gives a dendrogram, but expensive! Donne un dendrogram, est un bon utile, mais cher!

# There's more!

---



## Suggested reading

---

- Spectral clustering:  
<http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf>
- Bishop chapter 9.

## Next week

---

- We will add connections between  $h_n$ 's!

