Machine Learning for biology

V. Monbet

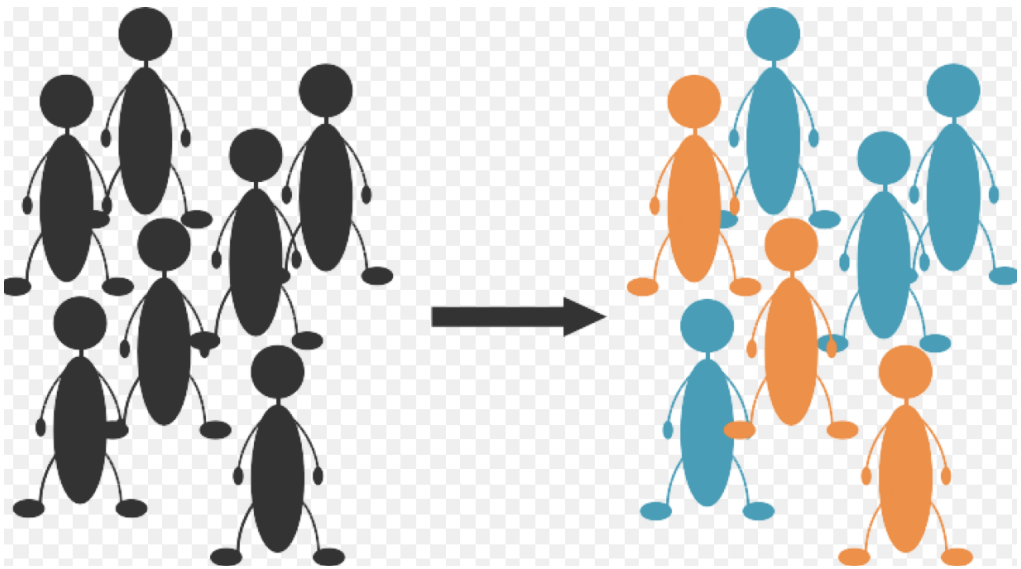
UFR de Mathématiques
Université de Rennes 1

Outline

Outline

Clustering

- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).
- Main methods
  - Hierarchical classification
  - k-means
  - Gaussian mixture models



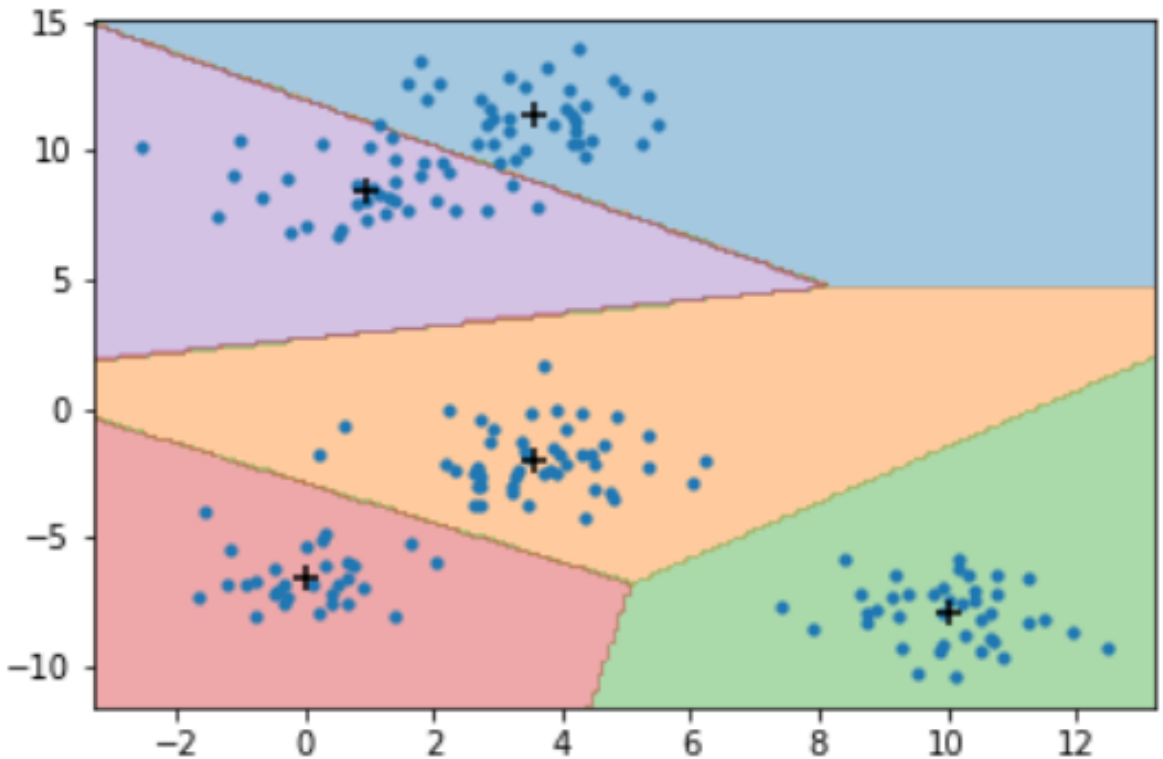
Outline

k-means algorithm

Ideas of the k-means algorithm

One gives, as inputs, the data set (the points) and a number of expected groups k.
The algorithms returns k centers.

Each point is associated to the closest center. It defines the groups.



k-means

Given

- a set of observations $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, $\mathbf{x}_1 \in \mathbb{R}^p$
- and the number of clusters k

k-means clustering aims to partition the n observations into k sets $\mathcal{C} = \{C_1, \cdots, C_k\}$.
In practice, the k-means algorithms searches the centers $\mu_1, \cdots, \mu_k$ which minimize the sum of the distances to the centers

$$\min_{\mu_1, \cdots, \mu_k} \sum_{\ell=1}^{k} \sum_{\mathbf{x} \in C_\ell} ||\mathbf{x} - \boldsymbol{\mu}_\ell||^2.$$

where

$\mu_\ell = \frac{1}{n_\ell} \sum_{i \in C_\ell} \mathbf{x}_i$ is the center (the mean) of cluster C_ℓ. n_ℓ is the number of points in C_ℓ.

$||\mathbf{x} - \boldsymbol{\mu}_\ell||^2 = \sum_{i \in C_\ell} (\mathbf{x}_i - \mu_\ell)^T (\mathbf{x}_i - \mu_\ell)$ is proportional to the variance of cluster C_ℓ (= mean of the square distance of the points to the mean).

k-means algorithm

k-means is solved by an iterative algorithm.

k-means algorithm

Given an initial set of k means $\mu_1^{(0)}, \cdots, \mu_k^{(0)}$

Repeat until convergence,
for i in 1 to n,

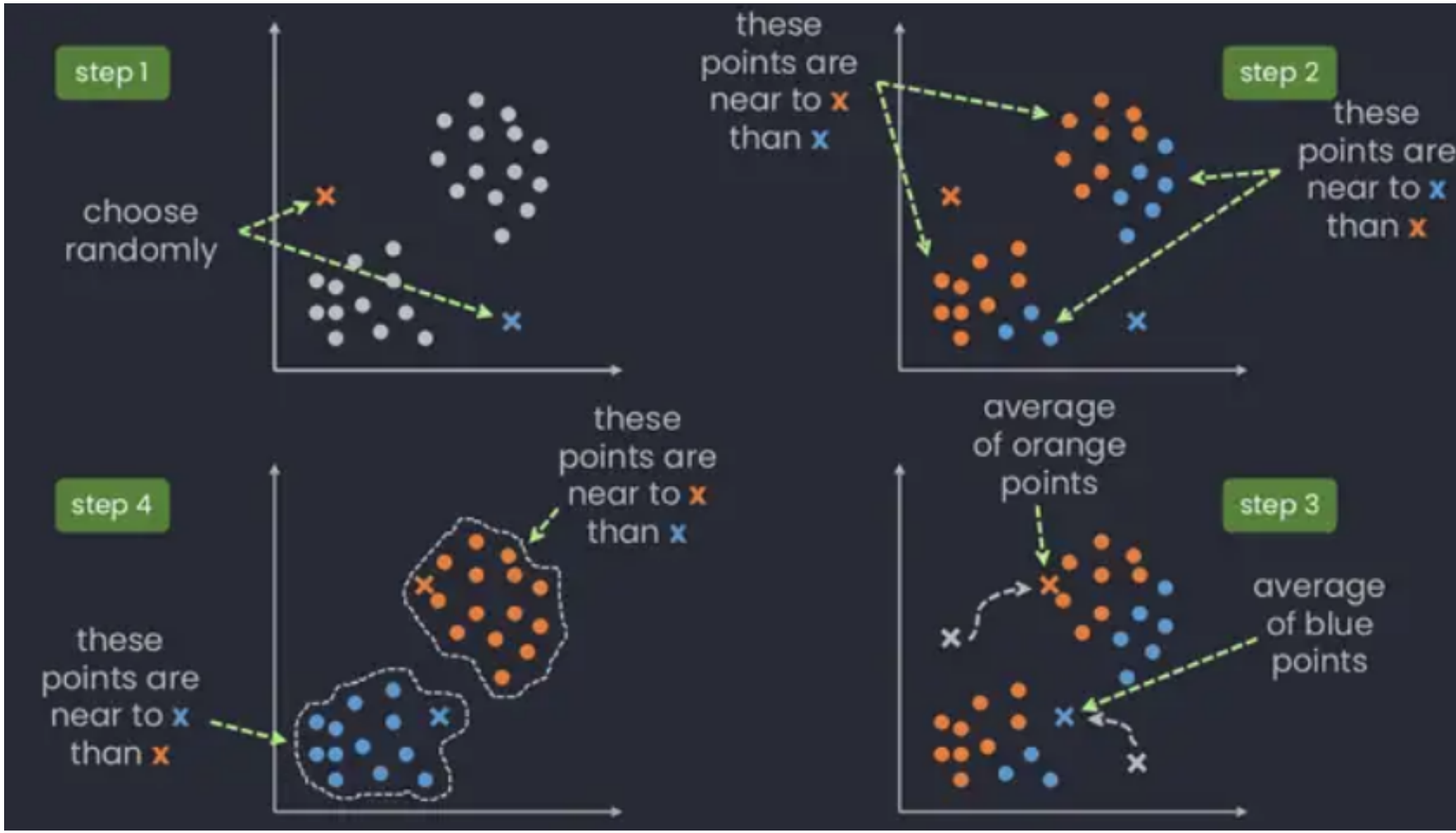
Assignment step

allocate $\mathbf{x}_i$ to C_ℓ if $||\mathbf{x}_i - \boldsymbol{\mu}_\ell|| \leq ||\mathbf{x}_i - \boldsymbol{\mu}_j||, \forall j = 1, \cdots, k$

Update step

compute the new centroids $\boldsymbol{\mu}_\ell^{(t)} = \frac{1}{|C_\ell|} \sum_{\mathbf{x} \in C_\ell} \mathbf{x}$

First steps of the k-means algorithm



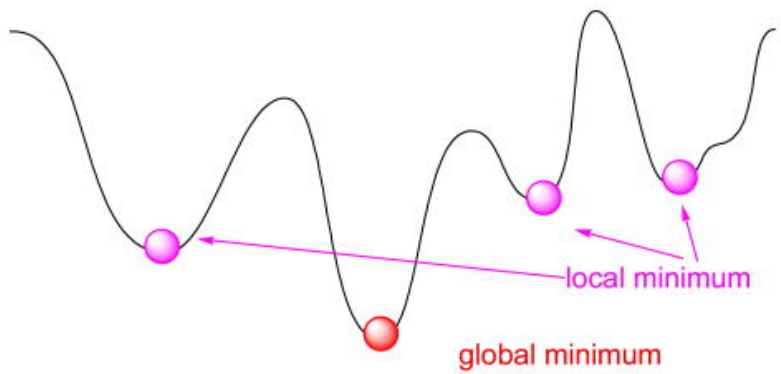
https://dinhanhthi.com/k-means-clustering/

Here is an interactive visualization :

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Some remarks about the k-means algorithms

- The Assignment step assigns each observation to the cluster whose mean yields the least within-cluster variance.
  Since the arithmetic mean is a least-squares estimator, the Update step also minimizes the within-cluster variance objective.
  That guarantees the **convergence of the algorithm to a local minimum**.
- Choice of the **initial cluster centers** can have an impact on the final cluster formation.
  This means that the outcome of clustering can be different each time the algorithm is run even on the same data set if the initial centers are chosen randomly.

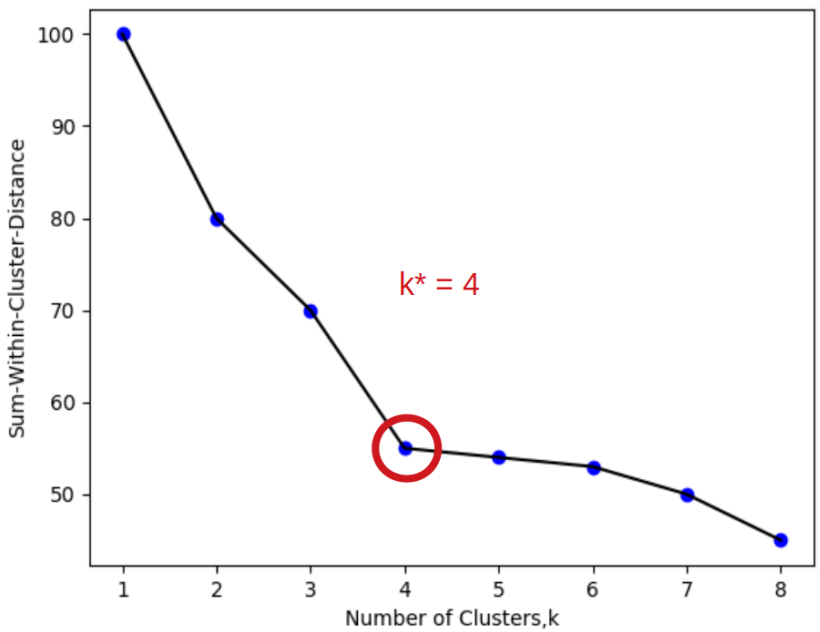


How to choose the number of clusters?

- The **number of clusters** k you want to group your data points into, has to be predefined.
  Within-cluster variance is a measure of compactness of the cluster. Lower the value of within cluster variance, higher the compactness of cluster formed.

$$W = \sum_{\ell=1}^{k} \frac{1}{|C_\ell|} \sum_{\mathbf{x} \in C_\ell} ||\mathbf{x} - \boldsymbol{\mu}_\ell||^2$$

where $|C_\ell|$ is the cardinal of C_ℓ (number of points in C_ℓ).



Error measure (within-cluster variance) decreases with increase in cluster number.
After a particular point, $k = 4$, W starts flattening.
Cluster number corresponding to that particular point, $k = 4$, should be considered as optimal number of clusters.

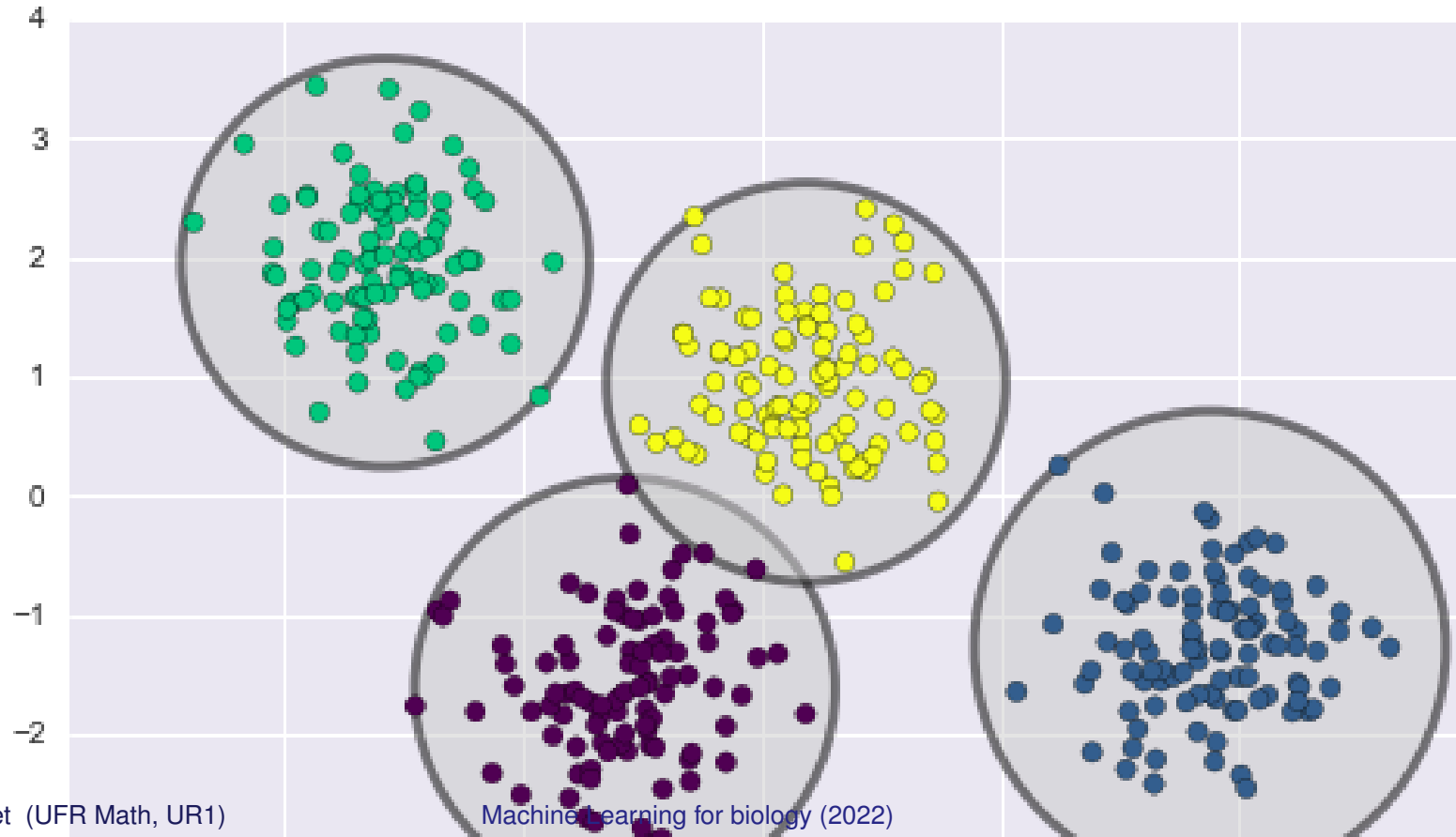
Trade-off between low W and interpretability.

Some remarks about the k-means algorithms

- kmeans tends to build clusters of similar size (or similar variance).
- Cluster formation is based on the computation of the means. It is very sensitive to the **presence of outliers**. Outliers pull the cluster towards itself, thus affecting optimal cluster formation.
  See (again) some examples here: `https://www.naftaliharris.com/blog/visualizing-k-means-clustering/`

k-means, toy example

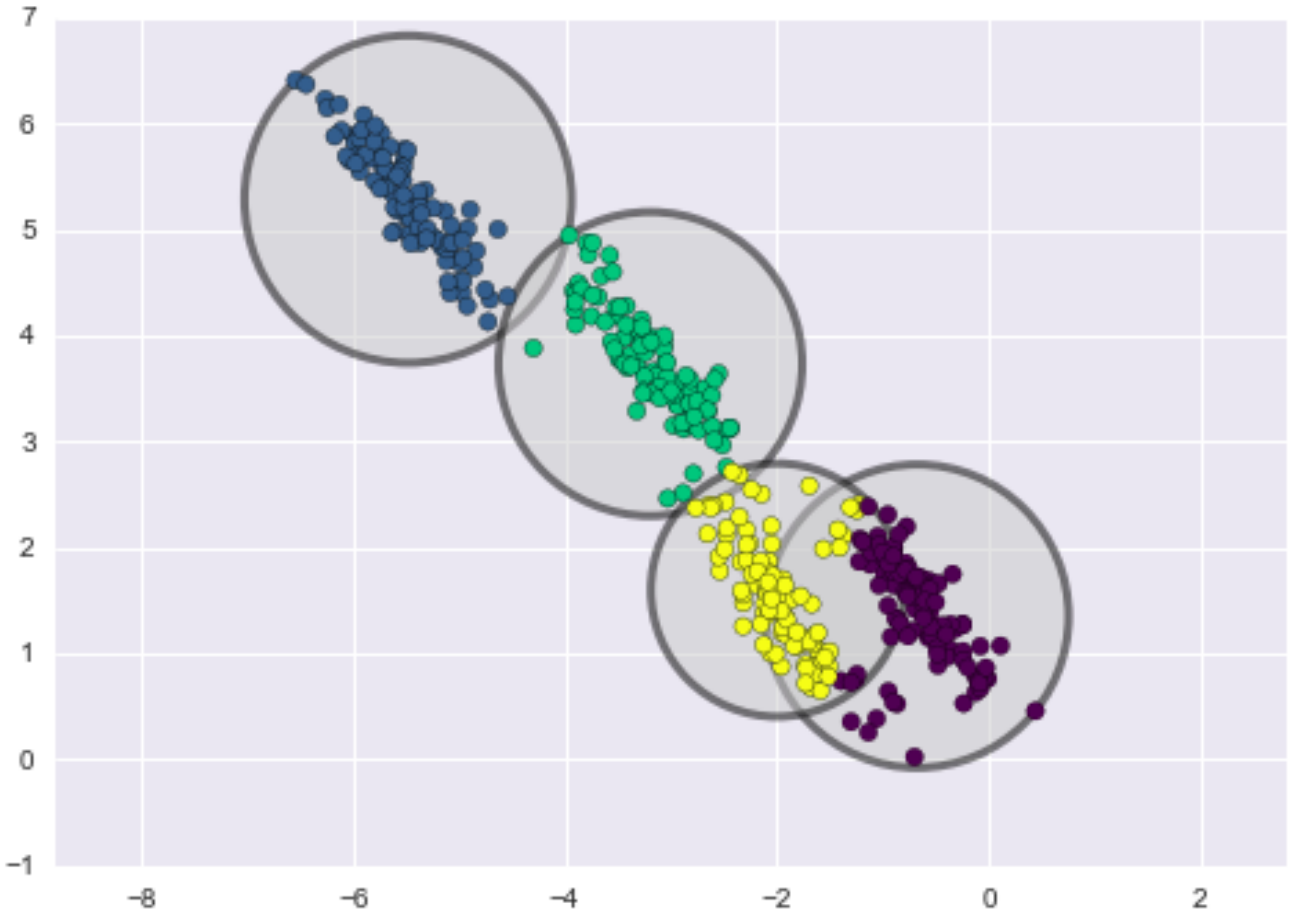
- Advantages of this algorithm:
  - simple
  - converges quickly
  - easy to adapt using a distance other than euclidean distance
- Drawbacks:
  - the number of clusters k may be hard to choose
  - it leads to linear frontiers between the clusters (see p. 6)
  - the clusters are expected to be of similar size



Outline

kmeans or Gaussian mixture models (GMM)?

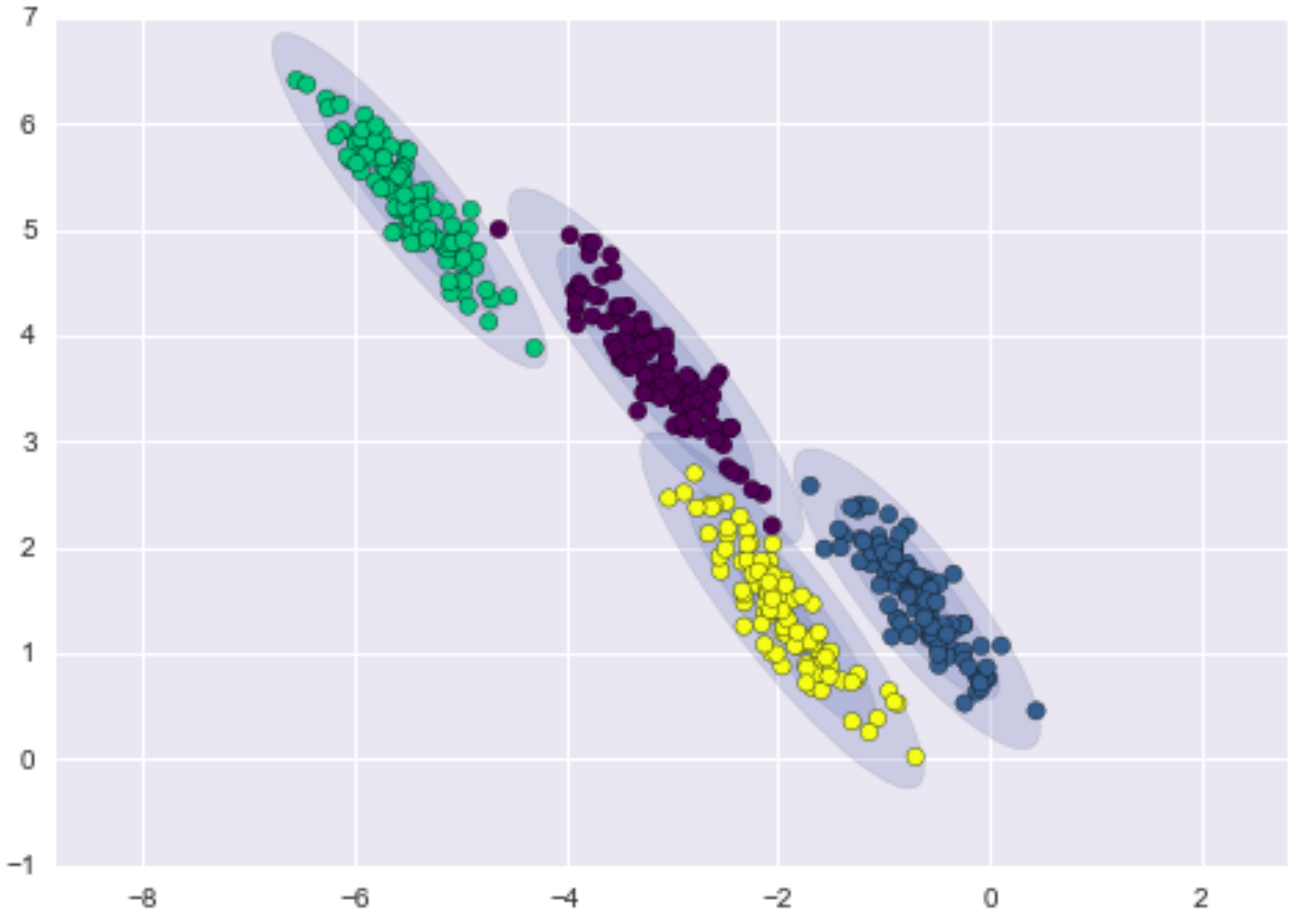
- k-means works fine for when your data is circular. However, when your data takes on different shape, you end up with something like this.



source: `https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e`

kmeans or Gaussian mixture models (GMM)?

- In contrast, Gaussian mixture models can handle even very oblong clusters.



source: `https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e`

Gaussian mixture models (GMM)

- Let $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^p$ be the observed variables and $C \in \{1, \cdots, k\}$ the cluster variable. GMM is a probabilistic model for $(\mathbf{X}, C)$ such that
  - $P(C = \ell) = \pi_\ell$
  - $X|C = \ell$ has a Gaussian distribution $\mathcal{N}(\mu_\ell, \Sigma_\ell)$ for all $\ell = 1, \cdots, k$.
- From the theorem of total probabilities Link,

$$P(\mathbf{X} \in A) = \sum_{\ell=1}^{k} P(X \in A | C = \ell) P(C = \ell)$$

so that the density is a weighted sum of Gaussian densities ϕ

$$f(\mathbf{x}) = \sum_{\ell=1}^{k} \pi_\ell \phi(\mathbf{x}; \mu_\ell, \Sigma_\ell)$$

- According to Bayes theorem Link, one can prove that

$$P(C = \ell | \mathbf{X} = \mathbf{x}) = \frac{\pi_\ell \phi(\mathbf{x}; \mu_\ell, \Sigma_\ell)}{\sum_{j=1}^{k} \pi_j \phi(\mathbf{x}; \mu_j, \Sigma_j)}$$

- GMM can be interpreted as *soft* version of k-means because it estimates $P(C = \ell | \mathbf{X} = \mathbf{x})$ instead of allocating "pure" clusters.

Gaussian mixture models algorithm

k-means algorithm

Given an initial set of k means $\mu_1^{(0)}, \cdots, \mu_k^{(0)}$
Repeat until convergence,

Assignment step- for i in 1 to n, allocate $\mathbf{x}_i$ to C_ℓ if $||\mathbf{x}_i - \boldsymbol{\mu}_\ell|| \leq ||\mathbf{x}_i - \boldsymbol{\mu}_j||, \forall j = 1, \cdots, k$

Update step- compute the new centroids $\boldsymbol{\mu}_\ell^{(t)} = \frac{1}{|C_\ell|} \sum_{\mathbf{x} \in C_\ell} x$

GMM algorithm

Given initial parameters
$\theta^{(0)} : \pi_\ell^{(0)}, \mu_\ell^{(0)}, \Sigma_\ell^{(0)}, \ \ell = 1, \cdots, k$
Repeat until convergence
E step- for i in 1 to n,
estimate posterior probabilities

$$\begin{aligned} T_{\ell,i} &= P(C = \ell | \mathbf{X} = \mathbf{x}_i, \theta^{(t)}) \\ &= \frac{\pi_\ell^{(t)} \phi(\mathbf{x}_i; \mu_\ell^{(t)}, \Sigma_\ell^{(t)})}{\sum_{j=1}^k \pi_j^{(t)} \phi(\mathbf{x}_I; \mu_j^{(t)}, \Sigma_j^{(t)})} \end{aligned}$$

M step- Compute new estimates

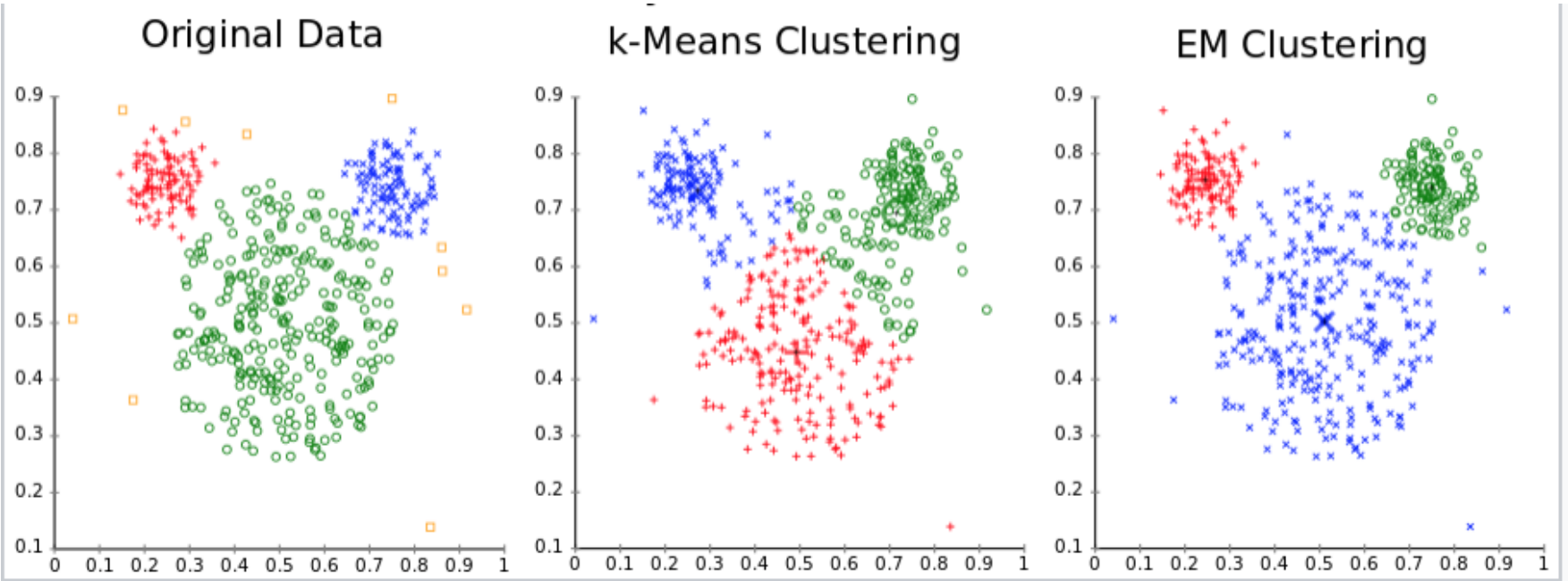
$$\begin{aligned} \pi_\ell^{(t+1)} &= \frac{1}{n} \sum_{i=1}^n T_{\ell,i} \\ \mu_\ell^{(t+1)} &= \frac{\sum_{i=1}^n T_{\ell,i} \mathbf{x}_i}{\sum_{i=1}^n T_{\ell,i}} \\ \Sigma_\ell^{(t+1)} &= \frac{\sum_{i=1}^n T_{\ell,i} (\mathbf{x}_i - \mu_\ell^{(t+1)})(\mathbf{x}_i - \mu_\ell^{(t+1)})^T}{\sum_{i=1}^n T_{\ell,i}} \end{aligned}$$

GMM, remarks

- BIC or AIC criteria can be used to estimate the number of clusters k.

$$\mathrm{BIC} = -2\ln(\widehat{L}) + \ln(n)k$$

  has to be minimum. $\widehat{L}$: maximum log likelihood.
- In GMM, the cluster variances are estimated: it allows different cluster sizes.
- Different variances leads to quadratic frontiers between clusters.
- Main drawback: when dimension of the data is large, there is too much parameters to be estimated.



GMM for large dimension dataset

Note that the GMM requires the estimation of one probability, one mean vector an one **variance** for each class.

For example, for the digits dataset $p = 784$:

- $\mu_\ell \in \mathbf{R}^p$, $\ell in \{1, \cdots, 10\}$.
- $\Sigma_\ell \in \mathbf{R}^{p,p}$, $\ell in \{1, \cdots, 10\}$.

Then, 10$\times$(1+784+784*783/2) $\simeq$ 3 millions parameters have to be learned.

The trick to manage such problems is to simplify the variance models by adding some constraints.

Choice of covariance model

- In GMM, the choice of variance model may be critical.
  - spherical = diagonal (the variables are supposed be independant to each other)+same for all variables (all the variables are supposed to have the same variance)
    here it looks elliptical because of the scale of the axis)
    This choice is good if the dimension of the data is large (large number of variables compared to the number of observations).
  - diag = diagonal (the variables are supposed be independant to each other)
    A relaxed version to the previous one: each variable have its own variance.
  - tied = full (the variables can have their own variance and be dependant to each other) but same for all clusters in order to control the number of parameters.
  - full = every things is free.

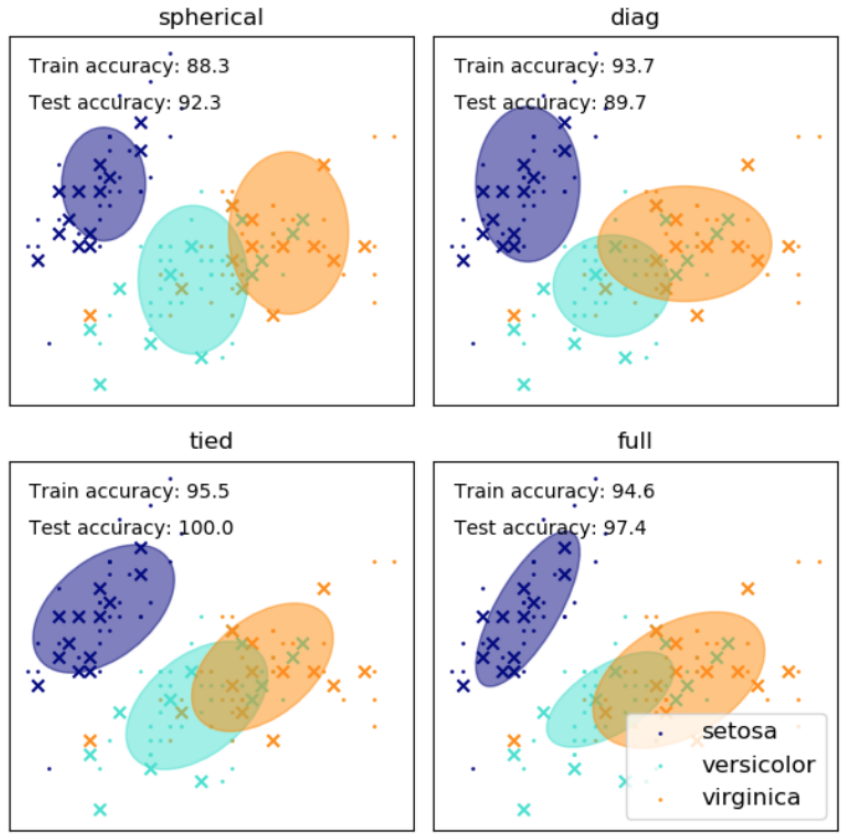


Figure from Python doc

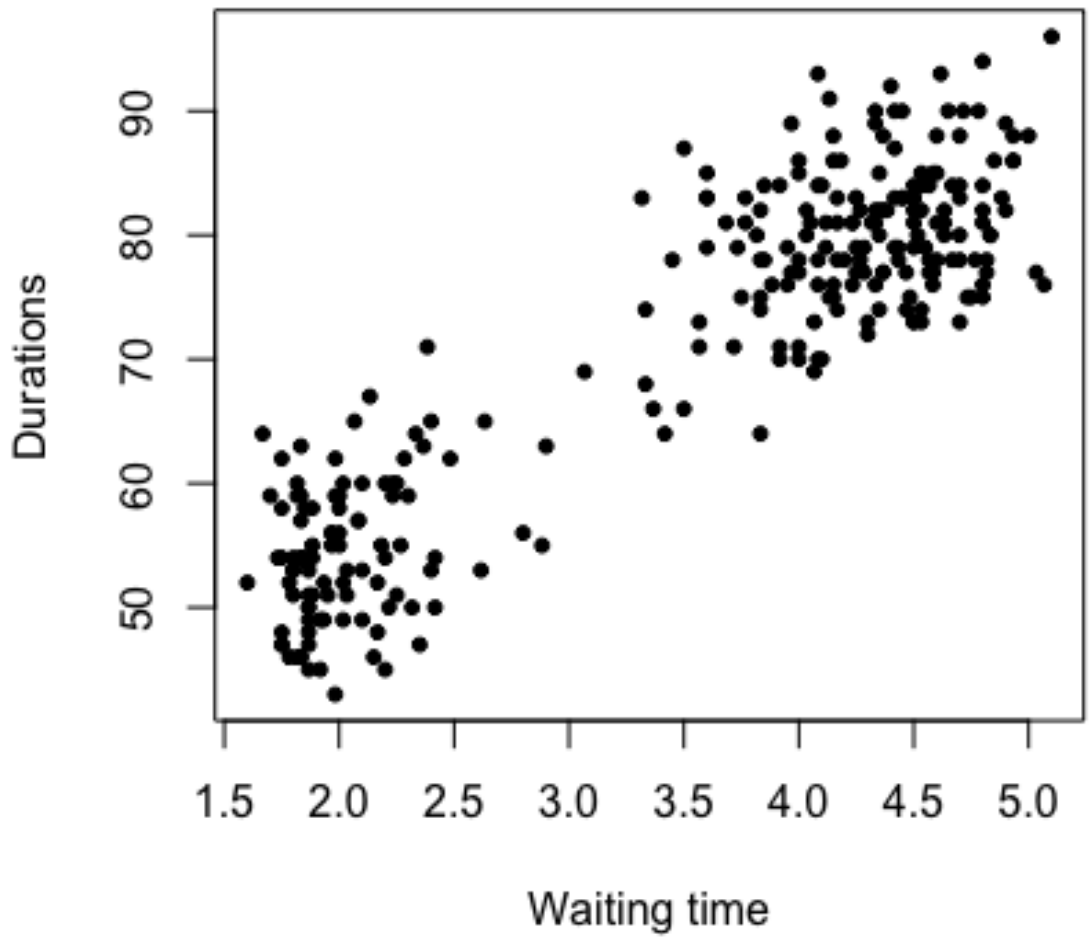
Example: faithful data

- Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

Source: CNN

Example: faithful data

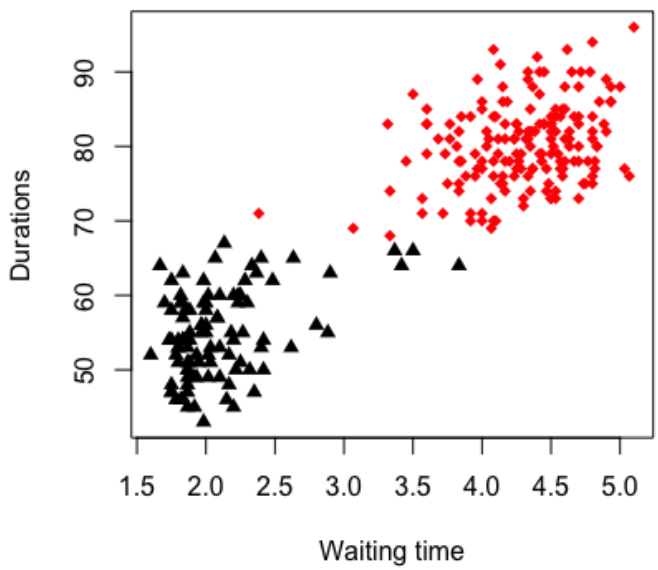
- Scatter plot of original data.



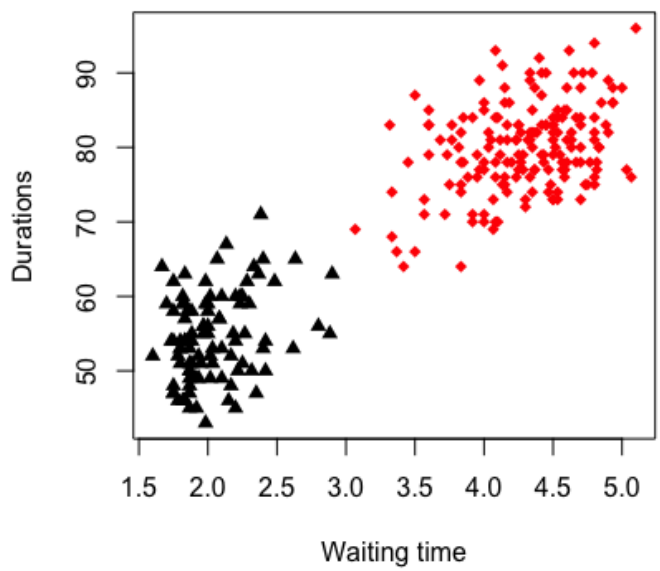
Example: faithful data

- The clusterings are different
- Kmeans tends to build clusters with same variances.

kmeans

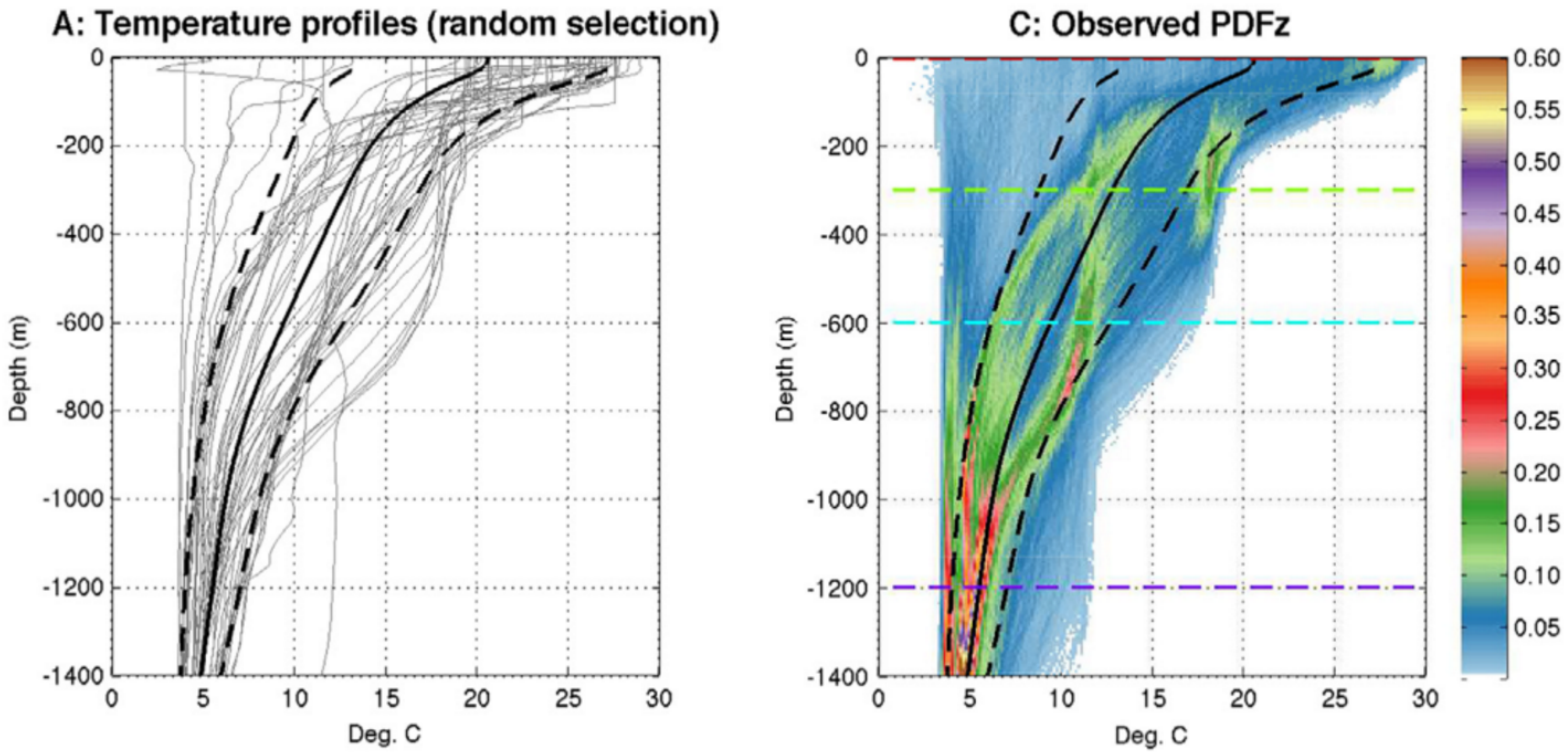


GMM



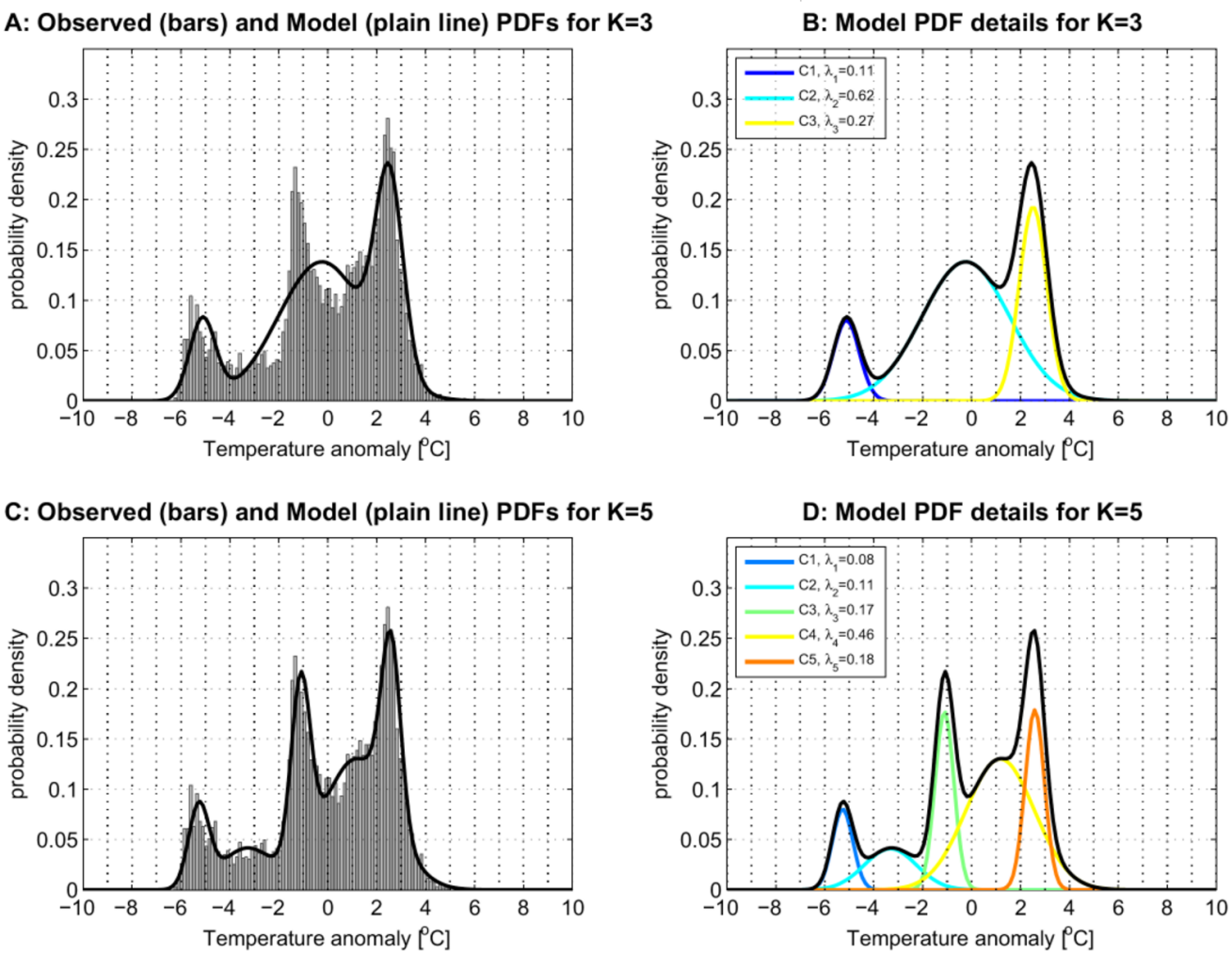
Argo temperature profiles in the North Atlantic Ocean

- Identify remarkable heat patterns in the horizontal and vertical plans.
- Data: Argo Heat Temperature Profiles. About 7000 profiles.



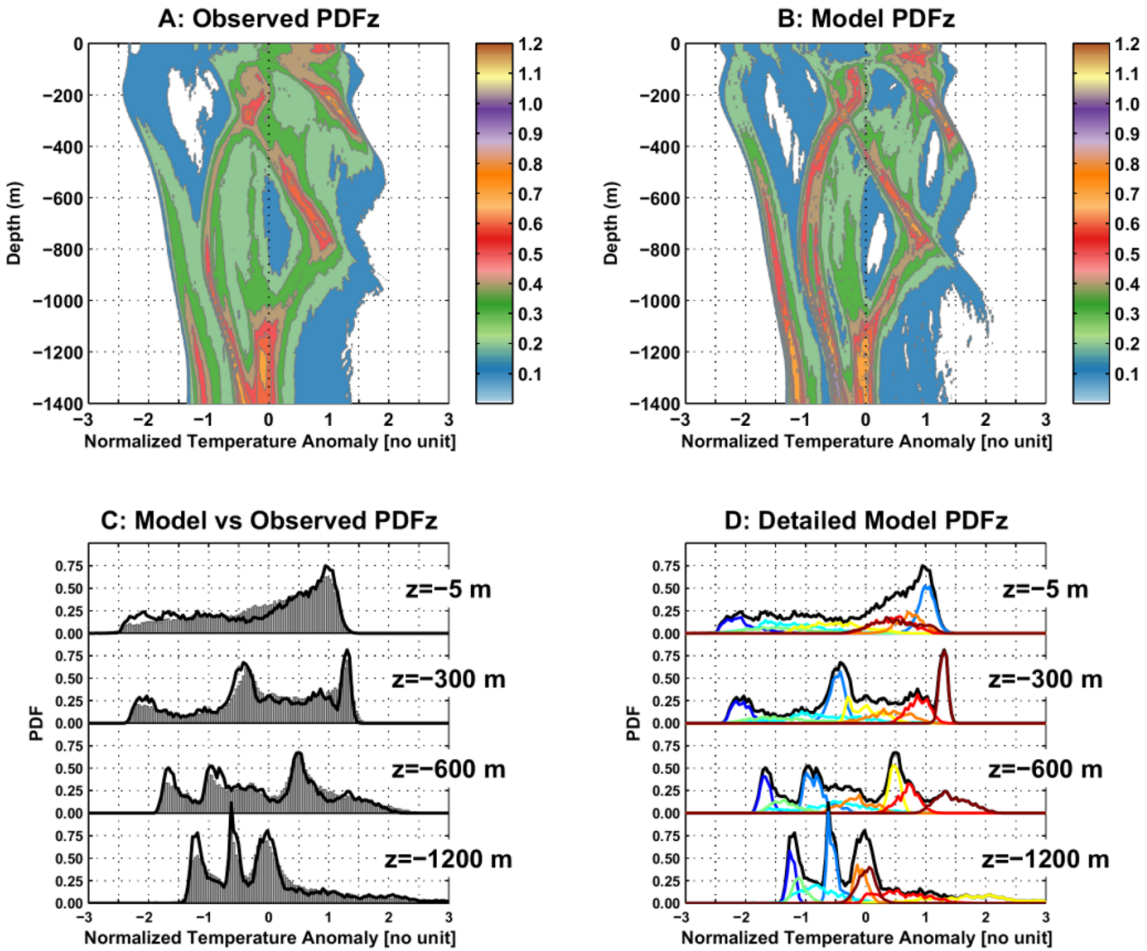
Method and results

- Example based on the surface temperatures (only 1 variable)
- Comparison of data and model pdfs helps to choose the number of clusters.



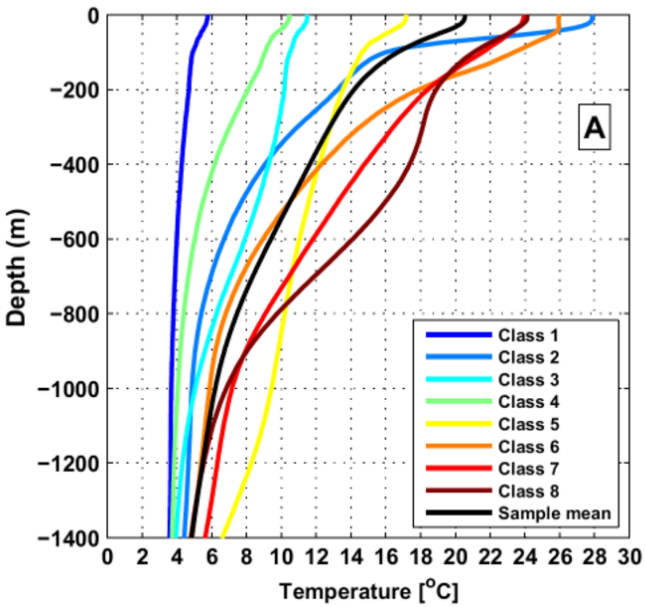
Method and results

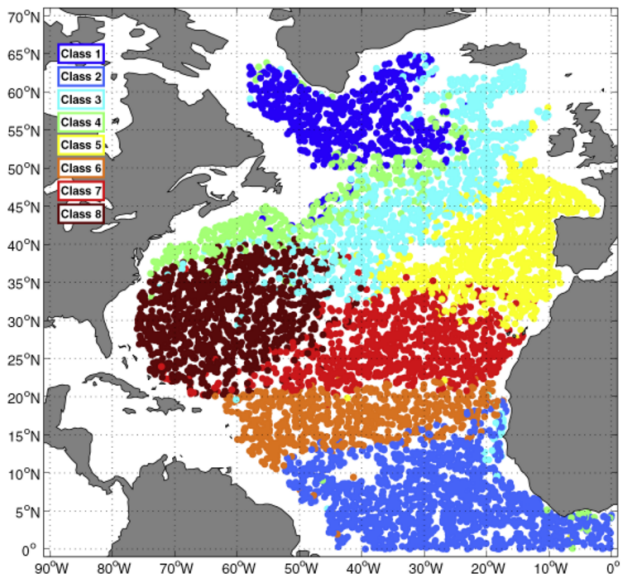
- Considering all the depths, the dimension is (too) high: a PCA is run first ($q = 11$ selected according to the reconstruction error)
- Pdfs at different depths.



Method and results

- Considering all the depths, the dimension is (too) high: a PCA is run first ($q = 11$ selected according to the reconstruction error)
- Mean profile for each cluster and geographical repartition of the clusters.





Concluding remarks

This lecture focused on "model based" algortihms.

- k-means is a very simple and powerfull algorithm for clustering.
- GMM is more flexible and provide more tools to chose the best model.
- GMM can be interpreted as a generalization of k-means.
- For using GMM (and k-means) preprocessing of the data may be needed, in particular dimension reduction.
- GMM (resp k-means) are usually introduced under Gaussian assumptions (resp euclidean distance) but this hypothesis can be relaxed.

There are other approaches clustering algorithms (Hierarchical Ascendent Clustering, DBSCAN, ...) for which it may be easier to change the considered distance.