CR12: Statistical Learning & Applications

Algorithms for Clustering

Lecturer: J. Salmon *Scribe: A. Alcolei*

Setting: given a data set $X \in \mathbb{R}^{n \times p}$ where n is the number of observation and p is the number of features, we want to separate these data into K classes (clusters), *i.e.*we want to learn :

1. the centroid (center) of each cluster
2. an assignation function $\mathcal{A} : \{1, \dots, n\} \to \{1, \dots, K\}$, meaning "sample x_i belongs to class $\mathcal{A}(i)$".

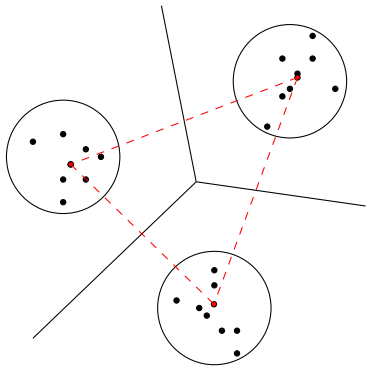


Figure 1: A simple representation of the situation ($n = 25, p = 2, K = 3$)

1 The K-means Algorithm

1.1 The Algorithm

The K-means algorithm [1.1] computes K clusters of a input data set, such that the average (squared) distance from a point to the centre of its cluster, *i.e.*the inertia, is minimized.

Theorem. *K-means monotonically decreases the inertia* $\frac{1}{n} \sum_{j=1}^{K} \sum_{i=1}^{n} \|x_i - c_j\|^2$

Proof. Let $\psi(X^{(t)}) = \frac{1}{n} \sum_{j=1}^{K} \sum_{i=1}^{n} \|x_i, c_j\|^2$ where $X^{(t)}$ is the current partition $X_1^{(t)}, \dots, X_K^{(t)}$ with centroids

Algorithm 1 The K-means Algorithm

Input: a data set $X = \{x_1, \dots, x_n\}$ ($x_i \in \mathbb{R}^p$).
Output: a partition $M = \{X_1, \dots, X_K\}$ of X together with the centroids $c_1, \dots, c_K$ of each cluster.
Initialization: choose $c_1, \dots, c_K$ in X at random
Repeat until convergence:

- **for** $j = 1 \dots K$ **do** $X_j \longleftarrow \emptyset$
- assignment step:
  for $i = 1 \dots n$ **do**
  $\mathcal{A}(x_i) \longleftarrow \underset{j \in \{1,\dots,K\}}{\arg\min} \|x_i - c_j\|^2$
  $X_{\mathcal{A}(x_i)} \longleftarrow X_{\mathcal{A}(x_i)} \cup \{x_i\}$
  done
- re-estimation step:
  for $j = 1 \dots K$ **do**
  $n_j \longleftarrow \sum_{i=1}^{n} \mathbb{1}(x_i \in X_j)$
  $c_j \longleftarrow \frac{1}{n_j} \sum_{i=1}^{n} x_i \mathbb{1}(x_i \in X_j)$
  done

return M, $c_1, \dots, c_K$

$c_1^{(t)}, \dots, c_K^{(t)}$ and assignation function $\mathcal{A}^{(t)}$, then

$$\begin{aligned}
\psi(X^{(t)}) &\geqslant \sum_{j=1}^{K} \sum_{x_i \in X_j^{(t)}} \|x_i, c^{(t)}_{\mathcal{A}^{(t+1)}_{(x_i)}}\|^2 && \text{(since } \mathcal{A}(x_i) \text{ minimizes the quantity } \|x_i - c_j\|^2 \text{ over all } j \in \{1, \dots, K\}) \\
&\geqslant \sum_{j=1}^{K} \sum_{x_i \in X_j^{(t)}} \|x_i, c_j^{(t+1)}\|^2 && \text{(since } c_j^{(t+1)} \text{ minimizes the quantity } \|x_i - c_j\|^2 \text{ over all } x_i \in X_j) \\
&\geqslant \psi(X^{(t+1)})
\end{aligned}$$

□

Corollary. *K-means stops after a finite number of steps.*

Proof. There is no infinite sequence of partitions such that the inertia decreases strictly since there is only a finite number of partitions: $\binom{n}{k}$. Thus the sequence $\psi(X^{(t)})_{t \in \mathbb{N}}$ has a finite number of values, *i.e.*there exists t such that $\psi(X^{(t+1)}) = \psi(X^{(t)})$. This implies that at step t, $X^{(t+1)} = X^{(t)}$ otherwise some elements would be wrongly classified. □

Remark.

- *The above corollary does not tell anything about how quick the algorithm converges, we only have an exponential bound:* $\binom{n}{k}$. *The time needed for the algorithm to converge depend on the initialization, some heuristic can be find in the literature to get better result.*
- *Similarly, the solution found by the algorithm is only a local optimal, since in general the inertia overall all partitions is not a convex function. The result depends on the initialization. Thus it might be useful to run the algorithm several times and pick the best result as a final answer.*

- *It is possible to parametrize the K-means algorithm for example by changing the way the distance between two points is measured or by projecting points on random coordinates if the feature space is of high dimension.*

1.2 Kernalised K-means

We change the previous algorithm so as to minimize in the reproducing kernel Hilbert space $\mathcal{H}$ associated to $\mathbb{R}^p$ instead of minimizing in $\mathbb{R}^p$. Using $\varphi : \mathbb{R}^p \to \mathcal{H}$, the algorithm remains the same except for:

- The initialization step: we choose $c_1, \dots, c_K$ in $\mathcal{H}$ instead of $\mathbb{R}^p$.
- The assignment step: we compute $\mathcal{A}_{x_i} \in \underset{j\in\{1,\dots,K\}}{\arg\min} \|\varphi(x_i) - c_j\|^2$ instead of $\mathcal{A}_{x_i} \in \underset{j\in\{1,\dots,K\}}{\arg\min} \|x_i - c_j\|^2$.

Remark. *We do not need to compute explicitly $\varphi(x_i)$ for each $x_i \in X$, all we need to know are the values $\langle \varphi(x_i), \varphi(x_j) \rangle$ for every pair $x_i, x_j \in X$.*

2 Gaussian Mixture and EM Algorithm

2.1 Gaussian maximum likelihood

The density of a Gaussian random variable over $\mathbb{R}^p$ is given by

$$\varphi_{\mu,\Sigma}(x) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

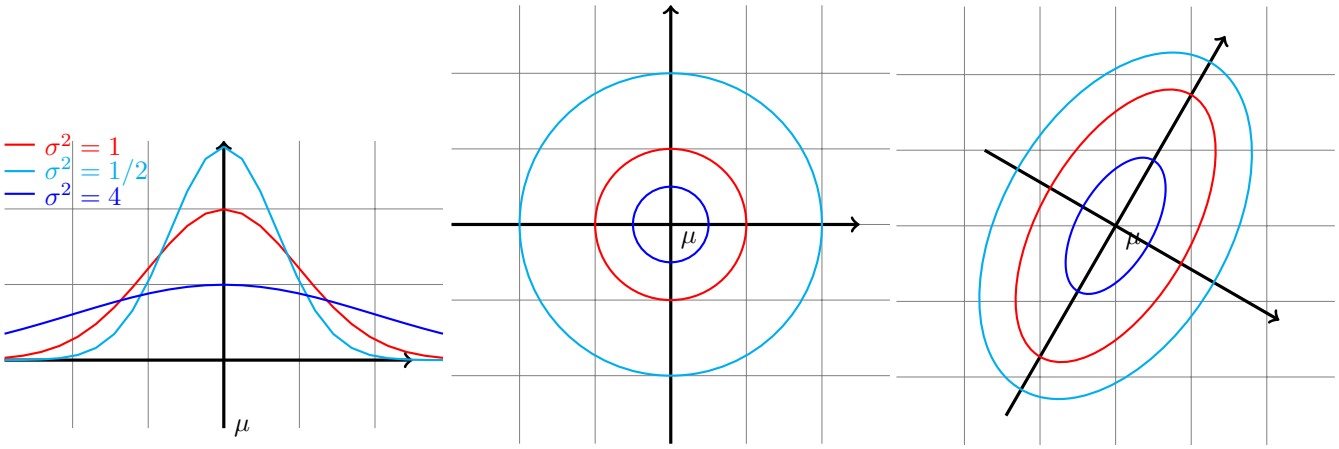
where μ is the mean of the variable ($\mu \in \mathbb{R}^p$) and Σ is the co-variance matrix ($\Sigma \in \mathbb{R}^{p\times p}$). Σ is positive definite so $rk(\Sigma) = p$. This formula satisfy the conditions for being a probability distribution:

1. $\forall x \in \mathbb{R}^p, \varphi_{\mu,\Sigma}(x) \geqslant 0$
2. $\int_{x\in\mathbb{R}^p} \varphi_{\mu,\Sigma}(x)dx = 1$

Example.
- $p = 1$, $\Sigma = \sigma^2$, $\mu = 0$: $\varphi_{\mu,\Sigma}(x) = \frac{\exp(\frac{-x^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}}$ *(cf. figure below for different value of σ)*
- $p = 2$, $\Sigma \in \mathbb{R}^2$, $\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, *the contour lines are described for all c in $\mathbb{R}$ by*

$$\begin{aligned}
\{x \in \mathbb{R}^p \mid \varphi_{\mu,\Sigma}(x) = c\} &= \{x \in \mathbb{R}^p \mid -\ln(\varphi_{\mu,\Sigma}(x)) = c'\} \quad \textit{(for } c' = \ln(c)\textit{)} \\
&= \{x \in \mathbb{R}^p \mid -\ln\left(\frac{1}{\sqrt{(2\pi)^p \det(\Sigma)}}\right) - \frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu) = c'\} \\
&= \{x \in \mathbb{R}^p \mid \sum_{i=1}^{p}\sum_{j=1}^{p} x_i x_j \alpha_{ij} + c'' = 0\} \quad \textit{(for some } a_{ij}\textit{, } c'' \textit{ depending on } \Sigma \textit{ and } c\textit{)}
\end{aligned}$$

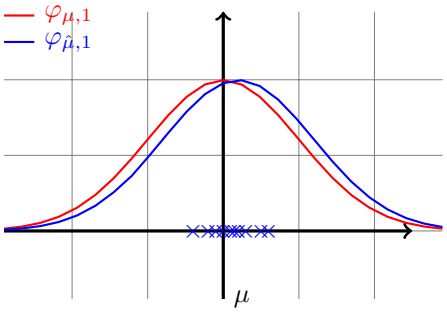
(cf. figures below for $\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$ and for general $\Sigma \in \mathbb{R}^2$ for different values of c).



In statistical machine learning we are interested in the following problem: suppose you observe $(X_1, X_2, \dots, X_n) \underset{iid}{\sim} \varphi_{\mu,\Sigma}$, can you estimate μ and Σ? (*iid* stands for independent and identically distributed)

Idea: Let $\varphi_{\mu,\Sigma}(X_1, \dots, X_n) := \prod_{i=1}^n \varphi_{\mu,\Sigma}(X_i)$, we want to find $(\hat{\mu}, \hat{\Sigma}) \in \underset{\mu,\Sigma}{\arg\max}\ \varphi_{\mu,\Sigma}(X_1, \dots, X_n)$. The quantity $\varphi_{\mu,\Sigma}(X_1, \dots, X_n)$ seen as a function of μ and Σ is called the *likelihood*. The pair $(\hat{\mu}, \hat{\Sigma})$ is called the *maximum likelihood*.

Example. *For $p = 1$, $\Sigma = 1$, we have $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$*



Proposition. *The empirical mean and the empirical co-variance are good estimators, i.e.*

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i \qquad \text{and} \qquad \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n (X_i - \hat{\mu})(X_i - \hat{\mu})^\top$$

Proof. We only show the first equality: Finding $(\hat{\mu}, \hat{\Sigma}) \in \underset{\mu,\Sigma}{\arg\max}\ \varphi_{\mu,\Sigma}(X_1, \dots, X_n)$ is equivalent to finding $(\hat{\mu}, \hat{\Sigma}) \in \underset{\mu,\Sigma}{\arg\min}\ \left[-\ln\left(\varphi_{\mu,\Sigma}(X_1, \dots, X_n)\right)\right] (*)$. Yet, $(*)$ is easier to solve since it involves minimizing over a sum rather than maximizing over a product :

$$(*) = \underset{\mu,\Sigma}{\arg\min}\ \left[c + \frac{1}{2}\cdot tr\left(\sum_{i=1}^n (X_i - \mu)\Sigma^{-1}(X_i - \mu)^\top\right) + \frac{n}{2}\cdot \ln(\det(\Sigma))\right]$$

where c is some constant that does not depend on μ or Σ.

Thus, fixing Σ we get:

$$(*) = \arg\min_{\mu} \left[\frac{1}{2} \cdot tr \left(\sum_{i=1}^{n} (X_i - \mu)\Sigma^{-1}(X_i - \mu)^\top \right) \right]$$

$\sum_{i=1}^{n}(X_i - \mu)\Sigma^{-1}(X_i - \mu)^\top$ is a convex function of μ so its global minimum $\hat{\mu}$ is the unique point that satisfies:

$$\frac{\delta}{\delta\mu} \left(\sum_{i=1}^{n} (X_i - \hat{\mu})\Sigma^{-1}(X_i - \hat{\mu})^\top \right) = 0$$

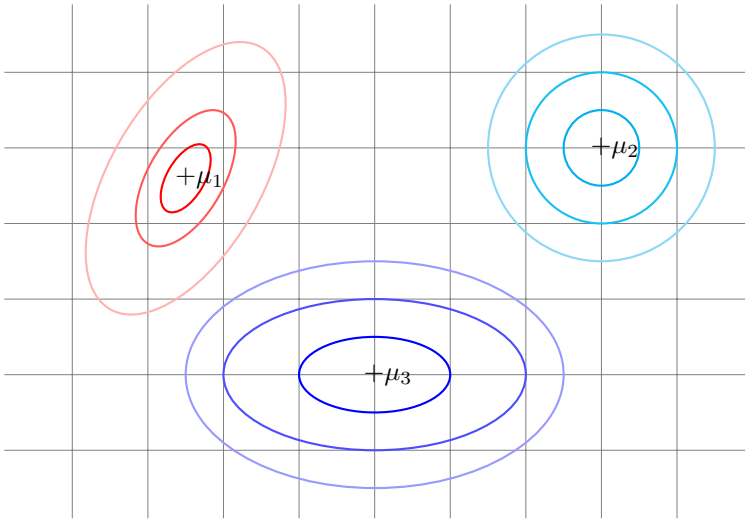
This implies that $\sum_{i=1}^{n} \Sigma^{-1}(X_i - \hat{\mu}) = 0$, that is $\sum_{i=1}^{n} X_i = n\hat{\mu}$ and so $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$ □

2.2 Mixture

We refine the model presented above by regarding the density of $(X_1, \ldots, X_n)$ as a mixture of K weighted gaussian densities, φ_{μ_k,Σ_k}, over $\mathbb{R}^p$:

$$(X_1, \ldots, X_n) \underset{iid}{\sim} f(x) = \sum_{k=1}^{K} \pi_k \cdot \varphi_{\mu_k,\Sigma_k}(x), \quad \text{where } \pi_k \text{ is the weight associated to } \varphi_{\mu_k,\Sigma_k}$$

Example. *In $\mathbb{R}^2$ for $K = 3$, $\pi_k = \frac{1}{3}$ we could have a distribution like the following:*



Drawing $x \in \mathbb{R}^p$ according to the distribution of the Gaussian mixture f is equivalent as drawing x as follows (hierarchical way):

1. *draw k with probability $\{\pi_1, \ldots, \pi_K\}$ over the elements of $\{1, \ldots, K\}$*
2. *draw $x \in \mathbb{R}^p$ according to the distribution associated to k, i.e. according to φ_{μ_k,Σ_k}*

The problem of finding the mixture of K Gaussian distributions from a given set of samples $(X_1, \ldots, X_n)$ can be seen as a generalization of the K-means problem where the distance to the centre of a cluster changes according to the index of the cluster. The Expectation-Maximization algorithm (EM) [2.2] can thus be viewed as a generalization of the K-means algorithm, where the value to maximize is

$$\varphi(\theta) = f_\theta(X_1, \ldots, X_n) = \prod_{i=1}^{n} f_\theta(X_i)$$

We have the same kind of termination property:

Proposition. *Let $\theta^{(t)}$ be the iterates of the EM algorithm and $\varphi(\theta^{(t)})$ be their corresponding inertia, then $\forall t$, $\varphi(\theta^{(t+1)}) \geqslant \varphi(\theta^{(t)})$.*

Proof. We do not give a complete proof here. The idea is the following: since maximizing over the likelihood $\varphi(\theta) = f_\theta(X_1, \dots, X_n)$ is hard, we instead maximize over the log-likelihood $L(\theta) = \ln(\varphi(\theta)) = \sum_{i=1}^n \ln(f_\theta(X_i))$. This is still hard to evaluate except if we knew from which Gaussian density inside the Gaussian mixture each X_i was drawn out. Thus for each $i \in \{1, \dots, n\}$ we define z_i to be the hidden random variable that indicates whether X_i is drawn from the j^{th} Gaussian density, with probability p_{ij} ($\sum_{j=1}^K p_{ij} = 1$), and we try to maximize the parametrized log likelihood $L(\theta, (p_{ij})_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant K}}) = \sum_{i=1}^n \ln\left(\sum_{j=1}^K \mathbb{1}_{z_i=j} \cdot f_{\theta_j}(X_i)\right)$. □

Remark.
- *once again the answer provided by the EM algorithm is only a local optimum and depends on the initialization.*
- *In practice, the EM algorithm is used for recovering missing or incomplete data.*

Algorithm 2 The Expectation-Maximization Algorithm

Input: a data set $X = \{x_1, \dots, x_n\}$ ($x_i \in \mathbb{R}^p$).

Output: $\theta := \begin{pmatrix} \pi_1, \dots, \pi_K \\ \mu_1, \dots, \mu_K \\ \Sigma_1, \dots, \Sigma_K \end{pmatrix}$, a set of weights and Gaussian densities that locally maximize the probability of the x_i's being drawn from the corresponding Gaussian mixture $f_\theta(x) = \sum_{k=1}^K \pi_k \cdot \varphi_{\mu_k, \Sigma_k}(x)$.

Initialization: choose $\theta := \begin{pmatrix} \pi_1, \dots, \pi_K \\ \mu_1, \dots, \mu_K \\ \Sigma_1, \dots, \Sigma_K \end{pmatrix}$ at random.

Let $p_{i,k}$ be the probability that x_i is coming from the k^{th} class.

Repeat until convergence:

- estimation step:
  for $i = 1 \dots n$ **for** $j = 1 \dots K$ **do**
  $p_{i,k} \longleftarrow \frac{\pi_j \cdot \varphi_{\mu_j, \Sigma_j}(x_i)}{f_\theta(x_i)} \left(= \frac{\pi_j \cdot \varphi_{\mu_j, \Sigma_j}(x_i)}{\sum_{k=1}^K \pi_k \cdot \varphi_{\mu_k, \Sigma_k}(x_i)} \right)$
  done
- maximization step:
  for $j = 1 \dots K$ **do**
  $\pi_j \longleftarrow \frac{1}{n} \sum_{i=1}^n p_{i,j}$
  $\mu_j \longleftarrow \frac{\sum_{i=1}^n p_{i,j} x_i}{\sum_{i=1}^n p_{i,j}}$
  $\sigma_j \longleftarrow \frac{\sum_{i=1}^n p_{i,j} (x_i - \mu_j)^\top (x_i - \mu_j)}{\sum_{i=1}^n p_{i,j}}$
  done

return M, $c_1, \dots, c_K$
