Unsupervised clustering with E.M.

Summary

We describe here the important framework of mixture models. These mixture models are rich, flexible, easy to handle, and possess a surprisingly large spectrum of possible applications.

Moreover, these mixture models may be easily interpreted, and estimated efficiently with the well known Expectation Maximization algorithm (E.M. for short). The purpose of this session is to detail theoretical and practical aspects of the clustering problem with a Bayesian modelling that involves mixtures.

1 Introduction

1.1 Definition of the clustering problem

Clustering is sometimes also called classification. In fact, we should pay attention at the very beginning of this lecture at the statistical framework : the classification problem can be supervised or unsupervised.

- K class supervised classification : we observe in the training set $\mathcal{D}_n := \{(X_1, Y_1) \ldots, (X_n, Y_n)\}$ both positions of the observations $X_i \in \mathbb{R}^p$ and labels for these observations $Y_i \in \{1, \ldots, K\}$. In this framework, we need to build the best classifier possible ϕ to minimize the risk loss $\mathcal{R}$

$$\mathcal{R}(\phi) := \mathbb{E}[Y \neq \phi(X)] \simeq \min_{f \in \Phi} \mathbb{E}[Y \neq f(X)],$$

  In a sense, the performance of prediction on a new observation is what matters, but K is known !
- Unsupervised classification : we observe in the training set $\mathcal{D}_n := (X_1, \ldots, X_n)$ only the positions but we know that a latent structure exists in the data, that derives from a natural partition of the state space. We want to predict a partition of the state space and of the data. We also may be interested in giving a confidence probability for an observation to be in a particular class.
  In a sense, the performance on current observations is what matters.

  Important drawback : in general we do not know K !

We will be interested in the unsupervised classification problem, also called clustering problem.

1.2 Notations

$\mathcal{D} := \{X_1, \ldots, X_n\}$ is the data set and n is the number of observations.

K is the number of clusters $K \leq n$ and we are looking for

$$\Delta = \{C_1, \ldots, C_k\} \qquad \text{a partition of} \qquad \mathcal{D}$$

We can produce two results for the clustering :

- Hard clustering : The method says that an observation i belongs to only 1 cluster. We thus predict the partition of the observations $\{C_1, \ldots, C_k\}$.
- Soft clustering : The method says that an observation is more likely to belong to one of the K cluster by producing a probability distribution :

$$\mathbb{P}[X_i \in C_k] = \gamma_{k,i} \qquad \text{with} \qquad \sum_{k=1}^{K} \gamma_{k,i} = 1.$$

We can see in Figure 1 that some cases may be so simple that it is possible to produce a reliable Hard clustering (figure on the left). In other cases, the picture is not so clear and we only produce a probability to belong to one of the several classes. This is the purpose of the Soft clustering (figure on the right is an illustration).

From a state of the art point of view, the old fashioned algorithms like hierarchical clustering or K means produce some Hard clustering.

The purpose of the E.M. clustering is to propose a Soft clustering method.

1.3 What we won't discuss on

1.3.1 *Single linkage algorithm*

The Hierarchical clustering algorithm (also called Single linkage algorithm) is certainly one of the simplest method to produce a Hard clustering method. It only requires the knowledge of a distance between observations. Then, the policy is forward from bottom to up with a recursive grouping strategy.

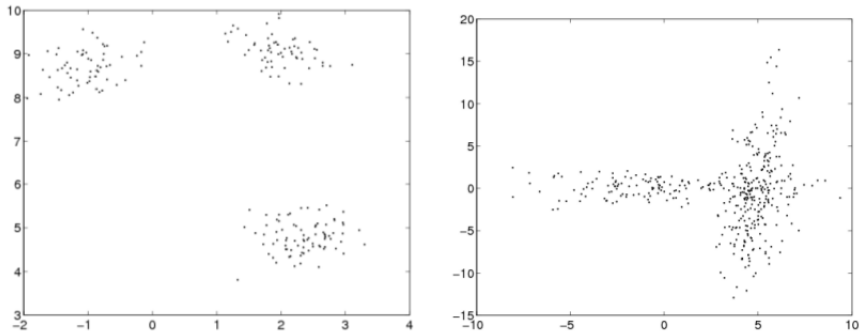


FIGURE 1 – An example of two situations where the clustering problem may be solved, but with certainly two different meanings (from a visual point of view).

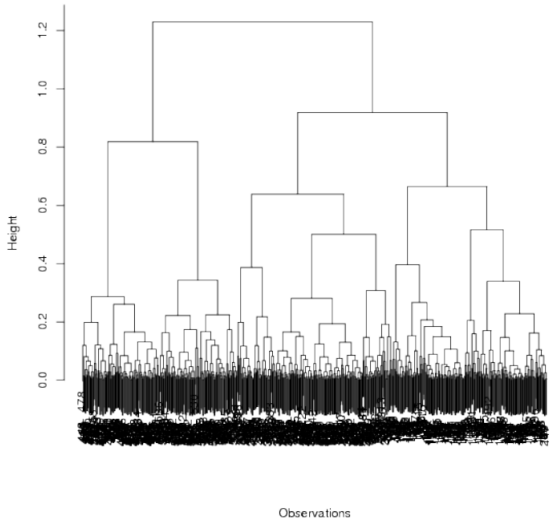


FIGURE 2 – An example of two situations where the clustering problem may be solved, but with certainly two different meanings (from a visual point of view).

Algorithm 1: Single linkage algorithm / Hierarchical clustering

Data: Dataset $\mathcal{D} = \{X_1, \ldots, X_n\} \subset \mathbb{R}^p$. Distance d.

Initialization : Form the set of n individual groups

$$\{X_1\}, \ldots \{X_n\} := \mathcal{G}_0$$

for $k = 1 \ldots n-1$ **do**

  Aggregate the two groups of observations $(A, B) \in \mathcal{G}_{k-1}^2$ such that

$$d(A, B) := min_{i \in A, j \in B} d(X_i, X_j).$$

  is as small as possible.

  Build the new set of groups :

$$\mathcal{G}_k := \mathcal{G}_{k-1} \setminus (\{A\} \cup \{B\}) \cup \{A \cup B\}.$$

end

Output : Sequence of groups $\mathcal{G}_0, \ldots, \mathcal{G}_{n-1}$.

The complexity of the method is weak since it requires only of the order n^2 iterations (*i.e.* the square of the number of observations). We can observe an execution in R of Algorithm 1 in Figure 2.

You may find many tutorials on hierarchical clustering with R on the www. In particular, the important matter is the choice of a good threshold in the lasting steps of the algorithm. This choice induces the number of groups obtained in the clustering, and the method used to fix this threshold is oftenly questionable.

1.3.2 K-means algorithm

The second classical method is the so-called K-means algorithm (see *e.g.* Lloyd, 1982 [2]). Given an integer K, it produces a recursive algorithm that build and update the groups sequentially.

We briefly provide a pseudo-code of the method in Algorithm 2. Moreover, Figure 3 proposes an illustration of the K-means algorithm until the convergence of the method is reached. Again, the stabilization of the method is quite rapid, but there is scarse theoretical supporting result on this method.

Algorithm 2: K-means clustering

Data: Dataset $\mathcal{D} = \{X_1, \ldots, X_n\} \subset \mathbb{R}^p$. Distance d.

Initialization : Pick randomly K centers in $\mathbb{R}^p$ denoted $\mu_1, \ldots, \mu_K$

Build K groups formed by the minimal distance assignment criterion :

$$X_i \in C_j^0 \Longleftrightarrow d(X_i, \mu_j) = \inf_{1 \leq k \leq K} d(X_i, \mu_k).$$

t=1

while *The groups are modified* **do**

Compute the barycenters of the groups $\mu_1^t, \ldots, \mu_K^t$

$$\mu_k^t = \frac{1}{|C_k^{t-1}|} \sum_{X_i \in C_k^{t-1}} X_i$$

Assign the observations X_i to one of the group according to the minimal distance assignment to each barycenter.

$$X_i \in C_j^t \Longleftrightarrow d(X_i, \mu_j^t) = \inf_{1 \leq k \leq K} d(X_i, \mu_k^t).$$

$t \longleftarrow t + 1$

end

Output : K groups $C_1, \ldots, C_K$.

2 Mixture models

2.1 Latent factor models

A mixture model associated to a clustering problem will consider that one do not face a too much simplistic situation where x belongs to only one class.

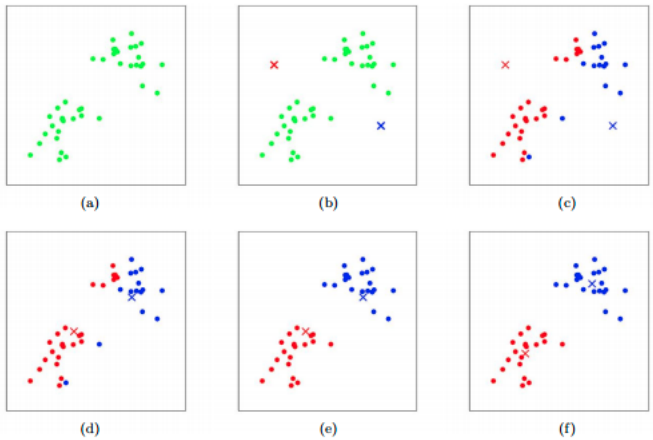


FIGURE 3 – Toy example for the K-means algorithm.

It mimics the fact that given a position x in the space, this point is more likely to belong to one class (in comparison with the other ones). This assumption can be quantified through a conditional probability to be a member of each class.

To be more precise, we consider now a family $(f_\theta)_{\theta \in \Theta}$ of densities over $\mathbb{R}^p$ and a probability distribution over Θ. We assume that the density of the observed phenomenon $(X_i)_{1 \leq i \leq n}$ may be written as

$$\forall x \in \mathbb{R}^p \qquad f(x) = \sum_{k=1}^{K} \pi_k f_{\theta_k}(x), \tag{1}$$

where $(\pi_k)_{1 \leq k \leq K}$ satisfies

$$\forall k \in \{1 \ldots, K\} \qquad \pi_k \geq 0 \qquad \text{and} \qquad \sum_{k=1}^{K} \pi_k = 1.$$

The latent factor assumption is that the class k (among $1 \ldots K$) has a probability π_k to be used in the initial mixture model (1).

DÉFINITION 1. — *[Parameters of the mixture] In Equation (1), $(\pi_k)_{1\leq k\leq K}$ is called the proportion of the mixture. The value of θ gather the locations of the mixture model.*

A typical example of mixture model is GMM (Gaussian Mixture Model) where

$$\forall k \in \{1,\dots,K\} \qquad \theta_k = (\mu_k, \Sigma_k).$$

Once the distribution π and location θ is determined, the mixture model permits to automatically give the degree of membership of *any* point of $\mathbb{R}^p$ through the Bayes rule :

$$\mathbb{P}[C_k|x] = \frac{\mathbb{P}[C_k] \times \mathbb{P}[x|C_k]}{\mathbb{P}[x]} = \frac{\pi_k f_{\theta_k}(x)}{\sum_{j=1}^{k} \pi_j f_{\theta_j}(x)} \tag{2}$$

The value of this function (and in particular the most typical class at point x) provides a natural partition of the state space :

$$C_k := \left\{ x \ : \pi_k f_{\theta_k(x)} > \max_{j\neq k} \pi_j f_{\theta_j(x)} \right\}$$

We represent in Figure 4 an example of three Gaussian densities mixed with coefficients $1/3, 1/3, 1/3$ in dashed lines. The corresponding global mixture model density is represented in straight line.

Of course, in general the parameters $\pi = (\pi_1,\dots,\pi_K)$ and $\theta = (\theta_1,\dots,\theta_K)$ are unknown and need to be recovered !

2.2 Likelihood of the model

An efficient estimation in the model (1) is needed to make the mixture approach efficient for clustering. We briefly outline the likelihood approach below.

Assume $X_1,\dots,X_n$ are generated according to an unknown distribution $\mathbb{P}_\theta$, where $\theta \in \Theta$ has to be estimated. The M.L.E. principle is to compute

$$\text{Likelihood}(\theta|\mathcal{D}) = \mathbb{P}_\theta(\mathcal{D}) = \prod_{i=1}^{n} \mathbb{P}_\theta(X_i).$$

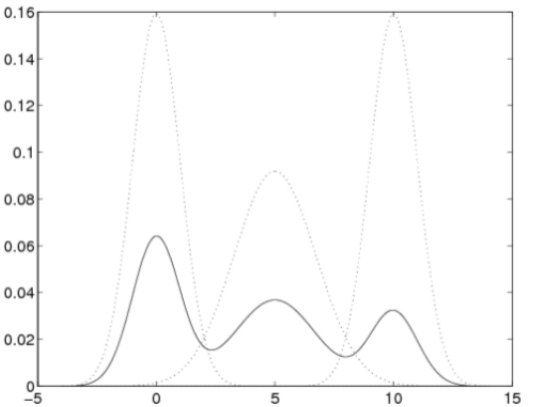


FIGURE 4 – Toy Gaussian mixture model.

An equivalent formulation involves the log-likelihood (denoted $\ell = \log L$). When the observations are assumed to be i.i.d., ℓ may be written as

$$\ell(\theta|\mathcal{D}) = \sum_{i=1}^{n} \ell\mathbb{P}_\theta(X_i).$$

What is called M.L.E. is the estimator $\hat{\theta}_n$ that maximises ℓ (or equivalently L) :

$$\hat{\theta}_n := \arg\max_{\theta\in\Theta} \ell(\theta|\mathcal{D}).$$

Of course, it is estimated on the data.

In particular, for the mixture model (1), we can write

$$(\hat{\pi}_n, \hat{\theta}_n) := \arg\max_{(\pi,\theta)} \ell((\pi,\theta)|\mathcal{D}), \tag{3}$$

with

$$\ell((\pi,\theta)|\mathcal{D}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k f_{\theta_k}(x)\right)$$

We can see that we need to solve a certainly complicated maximization problem in the variables π and θ. This may be handled directly by gradient ascent in (π, θ). Nevertheless, it is worthwhile saying that this approach has no theoretical fundation because of the nonconvexity of ℓ. An alternative relies on the use of the Expectation-Maximization algorithm, that aims to adopt a Minorization/Maximization procedure on the variational problem derived from (3).

2.3 E-M Algorithm

This algorithm has been introduced in the seminal work of [1]. It is maybe one of the most famous and useful academic work in statistics. We introduce the unobserved random variables Z_i that are the indicator variables from whom the observation X_i is sampled.

$$Z_i = k \Longleftrightarrow X_i \sim f_{\theta_k}.$$

We associate to the initial log-likelihood ℓ the complete log likelihood defined as

$$\ell((\pi, \theta)|X, Z) := \log \mathbb{P}((X, Z)|(\pi, \theta)).$$

This complete likelihood is now equal to

$$\begin{aligned} \ell((\pi, \theta)|X, Z) &:= \sum_{i=1}^{n} \log \left(\mathbb{P}(Z_i|\pi) \times \mathbb{P}[X_i|Z_i, \theta] \right) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}_{Z_i=k} \times [\log \pi_k + \log f_{\theta_k}(X_i)] \end{aligned}$$

since for each observation i, the probability to obtain (x, z) is to first choose class k with probability π_k, and then sample x with probability $f_{\theta_k}(x)$. This formula repeats each time Z is equal to k.

The interest of this formulation relies on the fact that the optimization may be splitted in a recursive two-step procedure that built a sequence of estimators !

Of course, the variables Z are not observed, otherwise the problem would be straightforward. It is nevertheless possible to compute the expected complete log likelihood : If we use the notation Θ for the current possible values of the parameters (in the mixture)

$$Q(\Theta|\Theta^{Old}) := \mathbb{E}_{(X,Z)\sim\mathbb{P}_{\Theta^{Old}}} \ell(X, Z, |\Theta). \tag{4}$$

Above, what is unknown are the values of Z. In particular, if $\Theta = (\pi, \theta)$:

$$Q(\Theta|\Theta^{Old}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}_{(X,Z)\sim\mathbb{P}_{\Theta^{Old}}} \left[\mathbf{1}_{Z_i=k} \times [\log \pi_k + \log f_{\theta_k}(X_i)] \right],$$

which in turn yields

$$Q(\Theta|\Theta^{Old}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{P}[Z_i = k|X_i, \Theta^{Old}] \times [\log \pi_k + \log f_{\theta_k}(X_i)].$$

E step At the current stage, we can impute the expected responsabilities of each class to $\mathbb{P}[Z_i = k|X_i, \Theta^{Old}]$:

$$\mathbb{P}[Z_i = k|X_i, \Theta^{Old}] = \frac{\pi_k^{Old} f_{\theta_k^{Old}}(X_i)}{\sum_{j=1}^{K} \pi_j^{Old} f_{\theta_j^{Old}}(X_i)} := \gamma_{i,k} \tag{5}$$

This formula is similar to Equation (2) and is just the conditional distribution of Z_i given X_i and the old value of the parameter Θ^{Old}.

M step We now maximise with respect to Θ the expectation of the complete likelihood $Q(\Theta|\Theta^{Old})$. Writing now $\Theta = (\pi, \theta)$, this is made possible since everything is splitted :

$$Q((\pi, \theta)|\Theta^{Old}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{i,k} \log[\pi_k + \log f_{\theta_k}(X_i)] = J(\pi) + \sum_{k=1}^{K} J(\theta_k).$$

As you can see, the step E is explicit and do not really deserve any special attention. This is not exactly the same situation for the M step.

Algorithm 3: E-M Algorithm

Data: Dataset $\mathcal{D} = \{X_1, \dots, X_n\} \subset \mathbb{R}^p$. Parametric model $(f_\theta)_\theta$.

1 **Initialization :** Choose an initial distribution π^0 in the probability simplex and pick a parameter θ^0 randomly in its state space.

2 Denote $\Theta^0 = (\pi^0, \theta^0)$.

3 **for** $t = 1 \dots + \infty$ **do**

4 **E Step** Use formula (5) to compute the expected responsabilities on each observation :

$$\forall i \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, K\} \qquad \gamma_{i,k}^t = \frac{\pi_k^{t-1} f_{\theta_k^{t-1}}(X_i)}{\sum_{k=1}^{K} \pi_k^{t-1} f_{\theta_k^{t-1}}(X_i)}.$$

5 **M Step** Update the parameters $\Theta^t = (\pi^t, \theta^t)$ with

$$\pi^t := \arg \max_{\pi : \sum \pi_k = 1} \sum_{k=1}^{K} \log[\pi_k] \left(\sum_{i=1}^{n} \gamma_{i,k}^t \right) \tag{6}$$

and

$$\forall k \in \{1 \dots K\} \qquad \theta_k^t := \arg\max_\theta \sum_{i=1}^{n} \gamma_{i,k}^t \log f_{\theta_k}(X_i). \tag{7}$$

6 **end**

7 **Output :** Limiting values in Θ^t.

2.4 Solving M in EM

A simple remark shows that the M step for the computation of π is simple (see Equation (6)). Using the Lagrange multiplier, we need to solve

$$0 = \partial_\pi \left[\sum_k \log \pi_k \sum_i \gamma_{i,k} + \lambda(1 - \sum_k \pi_k) \right],$$

which after brief algebra leads to

$$\pi_k := \frac{1}{n} \sum_i \gamma_{i,k}.$$

This formula is general and works for any mixture model.

However, in the other maximization procedure in θ given by Equation (7) of the M step, we need to solve the maximization procedure "at hand", depending on the kind of densities involved in the mixture model.

We specify here the M step in some particular cases :

- **Gaussian mixture with unknown means** We assume that f_θ is the density of a Gaussian standard random variable of mean θ :

$$f_\theta(x) = \frac{1}{(2\pi)^{d/2}} e^{-\|x-\theta\|^2/2}.$$

The maximisation over θ is explicit here, and leads to

$$\forall k \in \{1 \dots, K\} \qquad \theta_k^t := \sum_{i=1}^{n} \gamma_{i,j}^t X_i,$$

which is indeed the mean over the observatino of the posterior distribution on the K classes.

- **Gaussian mixture with unknown means and variance** The formula for the mean remains identical for the means recovery

$$\forall k \in \{1 \dots, K\} \qquad \theta_k^t := \sum_{i=1}^{n} \gamma_{i,j}^t X_i,$$

The variance/covariance term is given by

$$\forall k \in \{1 \dots, K\} \qquad \Sigma_k^t := \sum_{i=1}^{n} \gamma_{i,j}^t {}^t(X_i - \theta_k^t)(X_i - \theta_k^t),$$

where the term ${}^t(X_i - \theta_k^t)$ refers to the transpose of the vector $X_i - \theta_k^t$.

- **Poisson mixture with unknown means** The Poisson distribution is a discrete law over $\mathbb{N}$ such that

$$f_\theta(k) = e^{-\theta} \frac{\theta^k}{k!}$$

Again, the maximization (7) leads to

$$\forall k \in \{1 \dots, K\} \qquad \theta_k^t := \sum_{i=1}^{n} \gamma_{i,j}^t X_i.$$

Even though this formula is the same as the one involved for Gaussian distributions, please have in mind that it is not a general matter of fact !

3 Theoretical guarantees for EM Algorithm

Even though the algorithm makes both step (E and M) feasibles from a computational point of view, we need to understand what happens by iteratively solving line 4 and line 5 of Algorithm 3. This understanding relies again on the Minorization-Maximization principle already sketched for the Lasso.

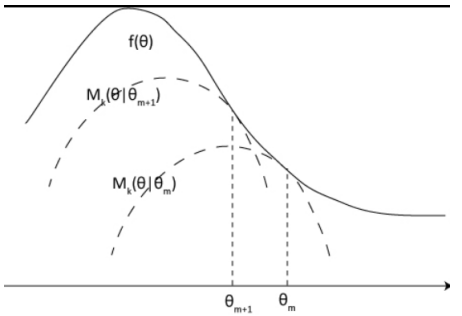
3.1 Principle of the MM algorithm

An algorithm is needed to solve the majorization problem

$$\arg\max_{\Theta \in \mathbb{R}^p} \{\ell(\Theta|X)\}.$$

An efficient method follows the principle of "Majorize Minorization" and is referred to as MM method.

- MM are useful for the minimization of a convex function/maximization of a concave one.
- Geometric illustration



- Idea : Build a sequence $(\Theta_k)_{k \geq 0}$ that converges to the maximum of $\ell(\Theta|X)$.
- MM algorithms are powerful, especially they can convert non-differentiable problems to smooth ones.

1. A function $g(\Theta|\Theta_k)$ is said to *minorize* f at point Θ_k if

$$g(\Theta_k|\Theta_k) = f(\Theta_k) \qquad \text{and} \qquad g(\Theta|\Theta_k) \leq f(\Theta), \forall \Theta \in \mathbb{R}^p.$$

2. Then, we define

$$\Theta_{k+1} = \arg\max_{\Theta \in \mathbb{R}^p} g(\Theta|\Theta_k)$$

3. Need a choice of a function $g(., \Theta_k)$ whose maximization is easy.
4. An example with a quadratic majorizer of a non-smooth function :
5. Important remark : The MM is a descent/ascent algorithm :

$$\begin{aligned} f(\Theta_{k+1}) &= g(\Theta_{k+1}|\Theta_k) + f(\Theta_{k+1}) - g(\Theta_{k+1}|\Theta_k) \\ &\geq g(\Theta_k|\Theta_k) + f(\Theta_{k+1}) - g(\Theta_{k+1}|\Theta_k) \\ &\geq f(\Theta_k) \end{aligned} \tag{8}$$

3.2 Application to E.M. Algorithm

We will establish that the sequence produced by $E.M.$ satisfies

$$\ell(\Theta|X) - \ell(\Theta_k|X) \geq Q(\Theta|\Theta_k),$$

where Q is defined in Equation (4). We compute :

$$\ell(\Theta|X) - \ell(\Theta_k|X) = \log \mathbb{P}_\theta(X)\pi(\theta) - \log \mathbb{P}_{\theta_k}(X)\pi(\theta_k),$$

We can introduce the hidden variables Z :

$$\begin{aligned}\mathbb{P}_\theta(X)\pi(\theta) &= \sum_Z \mathbb{P}_\theta(X,Z)\pi(\theta) = \sum_Z \mathbb{P}(X|\theta,Z)\mathbb{P}(Z|\theta)\pi(\theta) \\ &= \sum_Z \frac{\mathbb{P}(X|\theta,Z)\mathbb{P}(Z|\theta)\pi(\theta)}{\mathbb{P}(Z|X,\theta_k)} \times \mathbb{P}(Z|X,\theta_k)\end{aligned}$$

The map $\mathbb{P}(.|X,\theta_k)$ is a probability distribution and the Jensen inequality yields ($\log$ is a concave function) :

$$\log\left(\mathbb{P}_\theta(X)\pi(\theta)\right) \geq \sum_Z \log\left(\frac{\mathbb{P}(X|\theta,Z)\mathbb{P}(Z|\theta)\pi(\theta)}{\mathbb{P}(Z|X,\theta_k)}\right)\mathbb{P}(Z|X,\theta_k).$$

The second term of the difference may be written as :

$$\log\left(\mathbb{P}(X|\theta_k)\pi(\theta_k)\right) = \sum_Z \mathbb{P}(Z|X,\theta_k)\log\left(\mathbb{P}(X|\theta_k)\pi(\theta_k)\right).$$

Consequently :

$$\begin{aligned}&\log\left(\mathbb{P}_\theta(X)\pi(\theta)\right) - \log\left(\mathbb{P}_{\theta_k}(X)\pi(\theta_k)\right) \\ \geq\ & \sum_Z \mathbb{P}(Z|X,\theta_k)\log\left(\frac{\mathbb{P}(X|\theta,Z)\mathbb{P}(Z|\theta)\pi(\theta)}{\mathbb{P}(Z|X,\theta_k)}\right) \\ & -\sum_Z \mathbb{P}(Z|X,\theta_k)\log\left(\mathbb{P}(X|\theta_k)\pi(\theta_k)\right) \\ =\ & \sum_Z \mathbb{P}(Z|X,\theta_k)\log\left(\frac{\mathbb{P}(X|\theta,Z)\mathbb{P}(Z|\theta)\pi(\theta)}{\mathbb{P}(Z|X,\theta_k)\mathbb{P}(X|\theta_k)\pi(\theta_k)}\right) \\ =\ & \sum_Z \mathbb{P}(Z|X,\theta_k)\log\left(\frac{\mathbb{P}(X,Z|\theta)\pi(\theta)}{\mathbb{P}(X,Z|\theta_k)\pi(\theta_k)}\right) \\ :=\ & Q(\Theta|\Theta_k).\end{aligned}$$

The conclusion may be stated in the following result.

THÉORÈME 2. — *The likelihood of the sequence produced by the E.M. algorithm is increasing.*

Note that the Theorem above provides the convergence of the likelihood of the model, but nothing more can be said more on the convergence of the method (convergence of $(\Theta_k)_{k\geq 0}$, local traps,...).

4 Numerical example

We end the session with a brief numerical illustration through the famous Old Faithful eruption dataset, which concerns the elapse time in minute between each eruptions of a the Old Faithful geyser in the Yellowstone National Park.

We will use the mclust software, that is learning mixture of Gaussian random variables in R with the E.M. algorithm.

```
library(mclust)
plot(faithful)
```

An important point is the practical choice of K, the number of components. It can be made by maximizing the BIC criterion. The package compute first each optimal value of the log likelihood with E.M. and then penalize the score by using suitably the number of components (find how ?).

```
faithfulBIC <- mclustBIC(faithful)

plot(faithfulBIC)
```

What is the optimal number of component in the mixture model ?

```
faithfulSummary <- summary(faithfulBIC,
                  data = faithful)
faithfulSummary
plot(faithfulBIC, G = 1:7, ylim = c(-2500,-2300),
legendArgs = list(x = "bottomright", ncol = 5))
```

We can obtain a visual interpretation through (options 1,2,3 or 4).

```

plot(faithfulMclust)
```

and the predicted cluster may be obtained with

```
faithfulMclust$classification
```

The uncertainty of prediction is computed as well with

```
faithfulMclust$uncertainty
```

Références

[1] Dempster A.P., Laird N.M. et Rubin D.B., *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, Series B **39** (1977).

[2] S. P. Lloyd., *Least squares quantization in PCM*, IEEE Transactions on Information Theory **28** (1982).