Model-based Statistical Learning

Pr. Charles BOUVEYRON

Professor of Statistics
Chair of the Institut 3IA Côte d'Azur
Université Côte d'Azur & Inria

charles.bouveyron@univ-cotedazur.fr
@cbouveyron

EM algorithm for GMM:

The goal of the first step is to find the best mixture parameter estimates for the data at hand:

$$X \xrightarrow[\text{with EM algo}]{\text{learn}} \hat{\theta} = \left\{ \begin{matrix} \hat{\pi}_1, \dots, \hat{\pi}_K, \\ \hat{\mu}_1, \dots, \hat{\mu}_K \\ \hat{\Sigma}_1, \dots, \hat{\Sigma}_K \end{matrix} \right\}$$

The EM algorithm will optimize iteratively the likelihood of this model without directly optimize the function.

The EM algorithm

Starting with the GMM model, we can write the log-likelihood of the model:

$$\mathcal{L}(x;\theta) = \log\left(\prod_{i=1}^{m} p(x_i;\theta)\right)$$

$$= \log\left(\prod_{i=1}^{m} \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_i;\mu_k,\Sigma_k)\right)$$

$$= \sum_{i=1}^{m} \log\left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_i;\mu_k,\Sigma_k)\right)$$

We see here that it is totally possible to evaluate $\mathcal{L}(x;\theta)$ for specific values of x and θ, but the direct optimization is really difficult due to the $\log(\Sigma_k)$

The EM algorithm

The idea of the EM algorithm is to introduce an extra and non-observed (latent) variable Y, encoding the group memberships. Estimating both θ and Y from X is finally easier than just estimating θ from X.

$$Y_i \in \{0,1\}^K \implies Y_i = (0,0,1,0) \implies x_i \text{ belongs to the } 3^{rd} \text{ cluster}$$

$$\begin{cases} Y \sim \mathcal{M}(1;\pi) \\ X \mid Y = k \sim \mathcal{N}(\mu_k, \Sigma_k) \end{cases}$$

integrate over Y

$$\implies p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \Sigma_k)$$

The EM algorithm

This allows us to write another log-likelihood, called the complete data log-likelihood, of the couple (X,Y):

$$= \sum_{i=1}^{n} \log p((x_i, y_i); \theta)$$

$$\mathcal{L}_c(x,y,\theta) = \sum_{i=1}^{n} \left[\log p(y_i \mid x_i; \theta) + \underline{\log p(x_i;\theta)} \right]$$

$$= \mathcal{L}(x;\theta) + \sum_{i=1}^{n} \log p(y_i \mid x_i; \theta)$$

$$\Rightarrow \mathcal{L}(x;\theta) = \underbrace{\mathcal{L}_c(x,y;\theta)} - \sum_{i=1}^{n} \log p(y_i \mid x_i;\theta)$$

$$\Rightarrow \mathcal{L}_c \leq \mathcal{L}$$

The EM algorithm

Rmk: we see here that $\mathcal{L}_c$ is a lower bound of $\mathcal{L}$ and optimizing $\mathcal{L}_c$ over θ will automatically lead in the the optimization of $\mathcal{L}$.

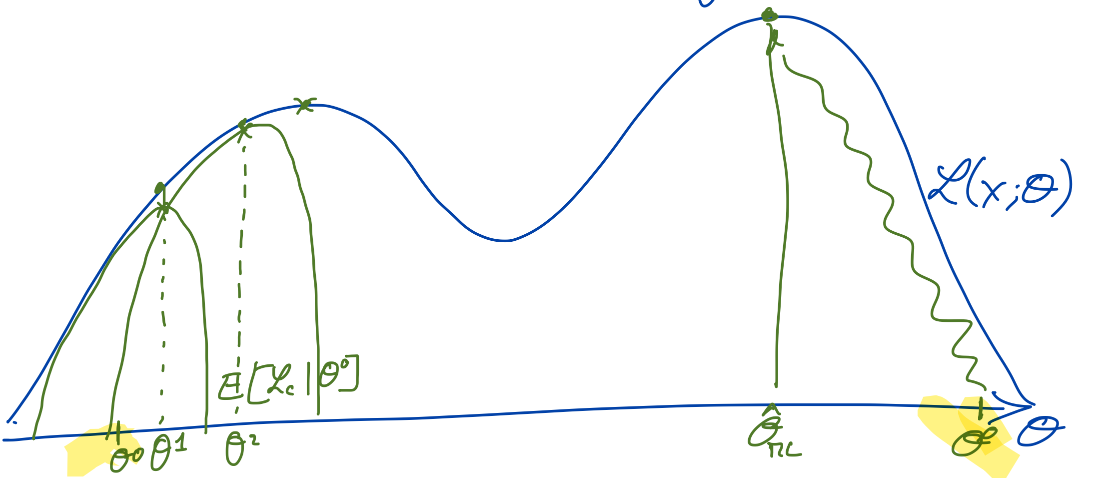
Thanks to this remark, Dempster, Laird and Rubin proposed in 1977 the EM algorithm:

- E step: the E step aims at calculating the expected complete log-likelihood:

$$Q(\theta;\theta^*) = E\left[\mathcal{L}_c(X,Y;\theta) \,\middle|\, X;\theta^*\right]$$

- M step: the M step aims at maximizing this function $Q(\theta;\theta^*)$ over θ to provide a new value for θ^*

Theorem: the sequence of estimates (θ^{*}) over the iterations of the EM algorithm is converging toward a local maximum of the log-likelihood.

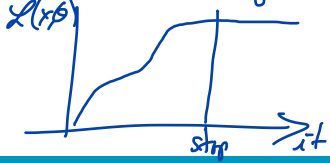


The EM algorithm for GMM

In practice:

1) to avoid being trapped in a local maximum, we usually do several (10) different random initializations and we keep afterward the $\hat{\theta}_{EM}$ with the highest likelihood.

2) to stop the algorithm, we just monitor the evolution of the log. likelihood and we stop when a plateau is detected



The EM algorithm for GMM

Starting with the E step, we need to focus on

$$Q(\theta, \theta^*) = E[\mathcal{L}_c(X, Y; \theta) \mid \theta^*, X]$$

and

$$\mathcal{L}_c(X, Y; \theta) = \sum_{i=1}^{m} \log p(x_i, y_i; \theta)$$

$$= \sum_{i=1}^{m} \log\left[\sum_{k=1}^{K} y_{ik}\, \pi_k\, \mathcal{N}(x_i; \mu_k, \Sigma_k)\right]$$

$$= \sum_{i=1}^{m} \sum_{k=1}^{K} y_{ik} \log\left(\pi_k\, \mathcal{N}(x_i; \mu_k, \Sigma_k)\right)$$

and therefore :

$$E\left[\mathcal{L}_c(X,Y|\theta) \,\middle|\, \theta^*; X\right] = \sum_{i=1}^{m} \sum_{k=1}^{K} \underline{E[y_{ik}|\theta^*, X]} \log\left(\pi_k \, \mathcal{N}(x_i; \mu_k, \Sigma_k)\right)$$

and $\underline{E[y_{ik}|\theta^*; X]} = P(y_{ik}=1 \,|\, X, \theta^*)$

$$\overset{\text{Bayes}}{=} \frac{P(y_{ik}=1|\theta^*) P(x_i | y_{ik}=1; \theta^*)}{p(x)}$$

$$\propto \underline{\pi_k^* \times \mathcal{N}(x_i; \mu_k^*, \Sigma_k^*)}$$

Let's call $t_{ik} = E[y_{ik}|\theta^*; X]$

$$\Rightarrow E\left[\mathcal{L}_c(X,Y;\theta) \,\middle|\, \theta^*; X\right] = \sum_{i=1}^{m} \sum_{k=1}^{K} t_{ik} \log \pi_k \, \mathcal{N}(x_i; \mu_k, \Sigma_k)$$

The EM algorithm for GMM

In the M step, we just have to optimize in θ the $E\left[\mathcal{L}_c(X,Y;\theta) \mid \theta^*; X\right]$:

$$\max_{\theta} E\left[\mathcal{L}_c(X,Y;\theta) \mid \theta^*; X\right] = \sum_{i=1}^{M} \sum_{k=1}^{K} t_{ik} \log \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)$$

$$Q(\theta, \theta^*) =$$

- Finding the update for π_k:

$$\frac{\partial}{\partial \pi_k} Q(\theta; \theta^*) = \frac{\partial}{\partial \pi_k} \left[\sum_i \sum_k t_{ik} \left[\log \pi_k + \log \mathcal{N}(x_i, \text{—}) \right] \right]$$