Devoir d'Analyse de données – M1 MAEF

Alissa DJEMA & Jean-Philippes HOUNTON

18 Mai 2020

Contents

Introduction

L'algorithme EM, de l'anglais expectation-maximization algorithm, a été établit pour la première fois en 1977, par Dempster, Laird, Rubin. Il s'agit d'un algorithme itératif permettent l'estimation paramétrique du maximum de vraisemblance. Ainsi, lorsque les données que l'on dispose ne permettent pas de maximiser analytiquement la vraisemblance, cette algorithme est une alternative. Son principe se résume généralement à deux étapes qui s'alterne. L'étape E et l'étape M d'où le nom d'algorithme EM. L'objectif de ce projet est de mener une analyse de données via un algorithme EM que l'on aura écrit et implémenté. Pour finir nous l'appliquerons à des données pour illustrer son fonctionnement.

1. Algorithme EM

La première étape de notre travail est la rédaction de l'algorithme EM. Pour ce faire, il est nécessaire d'écrire la vraisemblance de notre modèle. Dans cette étude on considère un n-échantillon $(X_1, \ldots, X_n)$ provenant d'un mélange de Poisson

$$f_\theta(x) = \sum_{k=1}^{K} \pi_k f_{\lambda_k}(x)$$

où $\pi_k > 0$, $\sum_{k=1}^{K} \pi_k = 1$ des probabilités du mélange et $f_{\lambda_k} \sim \mathcal{P}(\lambda_k)$ une loi de Poisson. Le paramètre du modèle que nous souhaitons estimer est le suivant.

$$\theta = (\pi_1, \ldots, \pi_{K-1}, \lambda_1, \ldots, \lambda_K) \in]\,0,1\,[^{K-1} \times \mathbb{R}_+^K$$

Ainsi, nous pouvons calculer la vraisemblance du modèle. La vraissemblance complète s'écrit :

$$\mathcal{L}(x_1, ..., x_n, z_1, ..., z_n, \theta) = \prod_{i=1}^{n} f_\theta(x_i)$$

$$= \prod_{i=1}^{n} (\pi_1 f_{\lambda_1}(x_i))^{\mathbb{1}_{z_i=1}} . (\pi_2 f_{\lambda_2}(x_i))^{\mathbb{1}_{z_i=2}} ..... (\pi_K f_{\lambda_K}(x_i))^{\mathbb{1}_{z_i=K}}$$

$$= \prod_{i=1}^{n} \prod_{k=1}^{K} (\pi_k f_{\lambda_k}(x_i))^{\mathbb{1}_{z_i=k}}$$

la log-vraissemblance s'écrit alors

$$\ln(\mathcal{L}(x_1, ..., x_n, z_1, ..., z_n, \theta)) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{z_i=k} \ln(\pi_k f_{\lambda_k}(x_i))$$

Dans un second temps, nous devons calculer l'espérance conditionnellement aux données. En effet, il s'agit de l'étape E. Cette dernière consiste à calculer l'espérance de la log-vraisemblance complète, conditionnellement aux données observées, pour un θ^{m-1} fixé, où m est le nombre d'étape réalisé. Ainsi, pour l'étape 1, il s'agit de θ^0 provenant de l'initialisation. Pour ce faire, nous avons initialisé nos paramètres, et fixé une valeur pour K. Notons que nombre de classe, ici K n'est pas connu à l'avance.

L'espérance de la log-vraissemblance complete est donc:

$$\mathbb{E}_{\theta^{m-1}}[ln(\mathcal{L}(x_1^n, z_1^n, \theta)|x_1^n]$$

$$= \mathbb{E}_{\theta^{m-1}}\left[\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{z_i=k} (\ln(\pi_k) + \ln(f_{\lambda_k}(x_i))|x_1^n)\right]$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}_{\theta^{m-1}}(\mathbb{1}_{z_i=k}|x_1^n)(\ln(\pi_k) + \ln(f_{\lambda_k}(x_i))$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{P}_{\theta^{m-1}}(z_i = k|x_1^n)(\ln(\pi_k) + \ln(f_{\lambda_k}(x_i))$$

En posant $\mathbb{P}_{\theta^{m-1}}(z_i = k|x_1^n) = \delta_i^k(\theta^{m-1})$, on peut écrire lespérance de la log vraissemblance comme

$$\mathbb{E}_{\theta^{m-1}}[ln(\mathcal{L}(x_1^n, z_1^n, \theta)|x_1^n] = \sum_{i=1}^{n} \sum_{k=1}^{K} \delta_i^k(\theta^{m-1})(\ln(\pi_k) + \ln(f_{\lambda_k}(x_i))$$

L'étape E est toujours suivis de l'étape M qui consiste à maximiser la vraisemblance. Cette maximisation est maintenant possible puisque l'on utilise l'estimation des données inconnues obtenu à l'étape précédente. Suite à cette phase, nous avons $\hat{\theta^m}$. Ainsi, on peut mettre à jour les valeurs des paramètres pour la prochaine itération.

On a donc:

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \delta_i^k(\theta^{m-1})$$

et

$$\hat{\lambda}_k = \frac{\sum_{i=1}^{n} \delta_i^k(\theta^{m-1}) x_i}{\sum_{i=1}^{n} \delta_i^k(\theta^{m-1})}$$

1.1 Initialisation

Pour commencer, on initialise θ tel que :

$$\theta^0 = (\frac{1}{K}, ..., \frac{1}{K}, 1, ..., 1).$$

En effet, $\forall k \in [1, K], \pi_k^0 = \frac{1}{K}$ et $\lambda_k^0 = 1$.

1.2 E-Step

Calcul de

$$\forall m \geq 1, \delta_i^k(\theta^{m-1}) = \frac{\pi_k^{m-1} f_{\lambda_k^{m-1}}(x_i)}{f_{\theta^{m-1}}(x_i)}$$

1.3 M-Step

Calculs des mises à jour pour $m \geq 1$

$$\pi_k^m = \frac{1}{n} \sum_{i=1}^{n} \delta_i^k(\theta^{m-1})$$

$$\lambda_k^m = \frac{\sum_{i=1}^{n} \delta_i^k(\theta^{m-1}) x_i}{\sum_{i=1}^{n} \delta_i^k(\theta^{m-1})}$$

2. Implementation

Nous pouvons à présent implémenter l'algorithme, et réaliser plusieurs tests pour vérifier la bonne convergence de l'algorithme. On sait notamment que l'algorithme est très dépendant de l'étape d'initialisation. D'où l'intérêt de réaliser plusieurs initialisations.

3. Validation de l'algorithme par des simulations

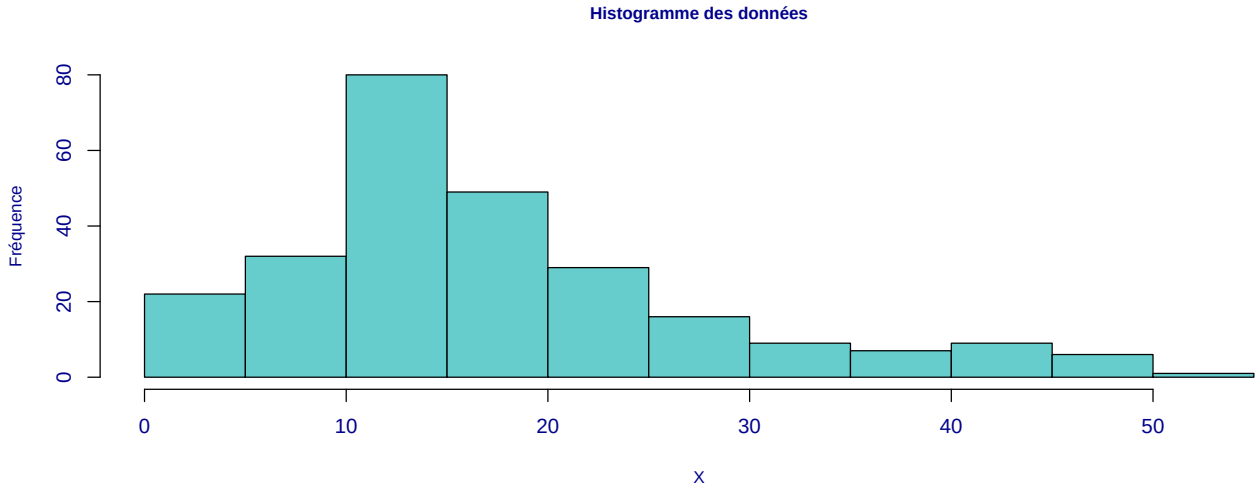
Pour sélection le nombre de composante de notre modèle. On propose d'utiliser un critère de vraisemblance pénalisé, le BIC.

$$BIC(\mathcal{M}) = -2 \ln p_{\mathcal{M}}\left(x_1^n\right) + |\mathcal{M}| \ln n$$

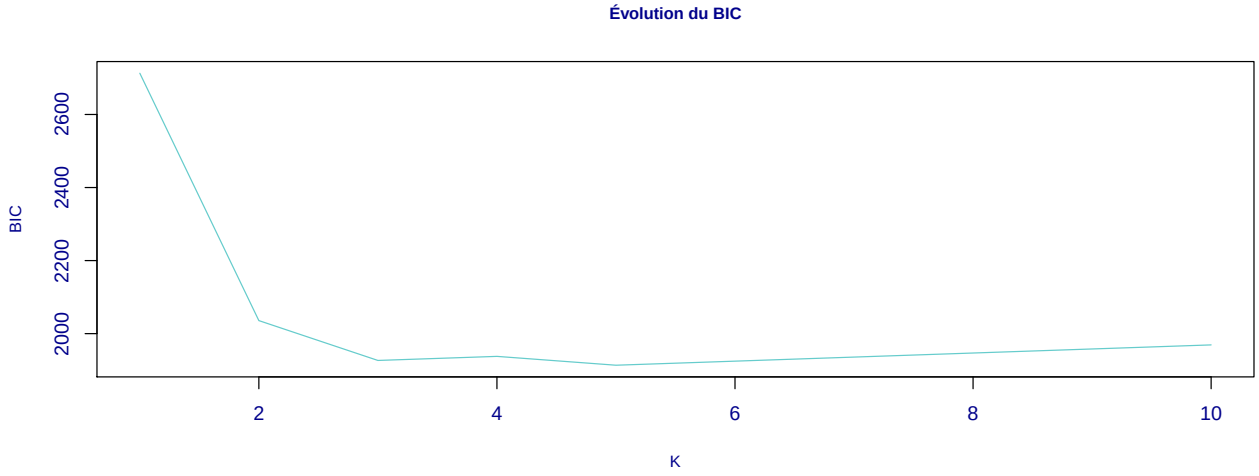
où M est un modèle de mélange à K classes, et $|\mathcal{M}|$ est le nombre de paramètre du modèle. Ainsi, on définit le nombre de classe K du modèle comme étant le minimum.

4. Application à un jeu de données (Tennis.csv)

On propose à présent d'implémenter notre algorithme, sur le de jeu de données tennis. Une première intuition avant de lancer notre algorithme serai de visualiser les données dans un histogramme. À la lecture des données, on pourrait supposer environ 5 classes coupées en décile.



Après notre agorithme et l'utilisation du critére BIC, on constate qu'il existe différents régimes. Ceux-ci sont au nombre de **4**. Voici ci-dessous les paramètres de notre modèle. On affiche également le graphique de la convergance du BIC.




```
## $pi
## [1] 0.1133145 0.2784287 0.2274940 0.1078782 0.2728845
##
## $lambda
## [1]  4.620736 13.523735 23.498118 40.946602 13.523627
```