Algoithmes Stochastiques
SEM, N-SEM, EM

Corrigés mis en ligne au fur et à mesure à l'adresse http://www.cmapx.polytechnique.fr/~benaych/M2AlgosStos/TP_Algo_EM/

1 Simulation des données

Fixer $K \geq 1$, choisir un paramètre $\theta := (\alpha_k, \mu_k, \sigma_k)_{1 \leq k \leq K}$, avec $\mu_1, \ldots, \mu_K \in \mathbb{R}$, $\sigma_1, \ldots, \sigma_K > 0$ et $\alpha_1, \ldots, \alpha_K > 0$ tel que $\alpha_1 + \cdots + \alpha_K = 1$ et tirer au hasard des données $(z_1, x_1), \ldots, (z_n, x_n)$ qui soient des copies indépendantes du vecteur aléatoire (Z, X) de loi $\mathbb{P}_\theta$ donnée par :

- $Z \in \{1, \ldots, K\}$ de loi $(\alpha_1, \ldots, \alpha_K)$,
- pour tout $k \in \{1, \ldots, K\}$, conditionellement à $Z = k$, $X \sim \mathcal{N}(\mu_k, \sigma_k^2)$.

Notons que, en retour, le passage de X à Z se fait avec la formule de Bayes : pour tout $x \in \mathbb{R}$ et $k \in \{1, \ldots, K\}$,

$$\mathbb{P}_\theta(Z = k | X = x) \;=\; \frac{\alpha_k f_{\mu_k, \sigma_k}(x)}{\sum_{l=1}^K \alpha_l f_{\mu_l, \sigma_l}(x)}. \tag{1}$$

avec

$$f_{\mu,\sigma}(x) := \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{2}$$

Les algorithmes SEM, N-SEM, EM ont pour but, d'estimer les paramètres $\mu_1, \ldots, \mu_K \in \mathbb{R}$, $\sigma_1, \ldots, \sigma_K > 0$ et $\alpha_1, \ldots, \alpha_K > 0$ ainsi que les valeurs des z_i par la seule observation des x_i.

2 Estimation des paramètres dans le cas où l'on connaît les z_i

L'EMV donne, étant données les observations $(z_i, x_i)_{1\leq i\leq n}$, les estimations suivantes pour le paramètre $\theta = (\alpha_k, \mu_k, \sigma_k)_{1\leq k\leq K}$: pour tout $k = 1, \ldots, K$,

$$\hat{\alpha}_k = \frac{|\{i\,;\, z_i = k\}|}{n}, \tag{3}$$

$$\hat{\mu}_k = \frac{1}{n\hat{\alpha}_k} \sum_{i\,;\, z_i=k} x_i, \tag{4}$$

$$\hat{\sigma}_k^2 = \frac{1}{n\hat{\alpha}_k} \sum_{i\,;\, z_i=k} (x_i - \hat{\mu}_k)^2. \tag{5}$$

3 Application des algorithmes

Appliquer chacun des 3 algorithmes suivants aux données $x_1, \ldots, x_n$ pour estimer θ. Estimer alors les z_i via (1) : pour tout i, on choisit

$$z_i := \text{argmax}_k\, \mathbb{P}_\theta(Z = k | X = x_i).$$

Comparer alors les valeurs de θ ainsi que des z_i à leurs vraies valeurs.

Algorithme SEM :

- initiallisation arbitraire des classes $z_i \in \{1, \ldots, K\}$
- répéter un grand nombre de fois :
  a) calculer $\hat{\theta}$ via (3), (4) et (5).
  b) pour tout $i = 1, \ldots, n$, tirer la valeur de z_i selon la loi

$$\mathbb{P}_{\hat{\theta}}(Z_i = \bullet | X_i = x_i)$$

  donnée par (1)

Algorithme SEM à N tirages :

- dupliquer N fois le jeu d'observations $(x_1, \ldots, x_n)$, qui devient donc $(x_i^{(j)})_{1\leq i\leq n,\, 1\leq j\leq N}$
- appliquer SEM à ce jeu de données étendu

Algorithme EM :

- initialisation arbitraire du paramètre θ_0
- étant donné θ_t, répéter jusqu'à convergence pour $t = 0, 1, 2, \ldots$:

  a) Calcul de la matrice

$$[\mathbb{P}_{\theta_t}(Z = k|X = x_i)]_{1\leq i\leq n,\, 1\leq k\leq K} = \left[\frac{\alpha_k^t f_{\mu_k^t,\sigma_k^t}(x_i)}{\sum_{l=1}^K \alpha_l^t f_{\mu_l^t,\sigma_l^t}(x_i)}\right]_{1\leq i\leq n,\, 1\leq k\leq K}$$

  b) Calcul de θ_{t+1} : pour tout $k = 1, \ldots, K$,

$$\begin{aligned}
\alpha_k^{t+1} &= \frac{1}{n}\sum_{i=1}^n \mathbb{P}_{\theta_t}(Z = k|X = x_i), \\
\mu_k^{t+1} &= \frac{1}{n\alpha_k^{t+1}}\sum_{i=1}^n \mathbb{P}_{\theta_t}(Z = k|X = x_i)x_i \\
(\sigma_k^{t+1})^2 &= \frac{1}{n\alpha_k^{t+1}}\sum_{i=1}^n \mathbb{P}_{\theta_t}(Z = k|X = x_i)(x_i - \mu_k^{t+1})^2. \qquad (6)
\end{aligned}$$

4 Extension au cas de la dimension supérieure

On va ensuite étendre ce qui précède au cas de la dimension $d \geq 1$. Pour chaque classe, on choisit donc $\mu_k \in \mathbb{R}^d$ et $\sigma_k^2 > 0$ est remplacé par une matrice symétrique réelle définie positive S_k^2. Les formules sont les mêmes, à ceci près que (2) devient (en considérant les données comme des vecteurs colonnes)

$$f_{\mu,S}(x) := \frac{1}{(2\pi)^{d/2}(\det S)^{1/2}} \mathrm{e}^{-\frac{(x-\mu)^\mathsf{T} S^{-1}(x-\mu)}{2}}, \qquad (7)$$

que (5) devient

$$\hat{S}_k = \frac{1}{n\hat{\alpha}_k}\sum_{i\,;\,z_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\mathsf{T} \qquad (8)$$

et que (6) devient

$$S_k^{t+1} = \frac{1}{n\alpha_k^{t+1}}\sum_{i=1}^n \mathbb{P}_{\theta_t}(Z = k|X = x_i)(x_i - \mu_k^{t+1})(x_i - \mu_k^{t+1})^\mathsf{T}. \qquad (9)$$

5 Application à des données non simulées

Nous allons étudier des données issues de cet article :

Bachrach LK, Hastie T, Wang M-C, Narasimhan B, Marcus R. *Bone Mineral Acquisition in Healthy Asian, Hispanic, Black and Caucasian Youth. A Longitudinal Study.* J Clin Endocrinol Metab (1999) 84, 4702-12,

et disponibles à l'adresse http://www.cmapx.polytechnique.fr/~benaych/M2AlgosStos/TPEM/ (fichier **densitesOs.txt**).

Ces données représentent des mesures relatives de densité minérale osseuse spinale sur des adolescents nord-américains. Chaque valeur est la différence de mesures prises sur deux visites consécutives, divisées par la moyenne.

L'idée est de savoir si la population peut être décrite par deux sous-groupes plus homogènes. Nous commencerons par analyser des données simulées puis ce nuage de points.

Les élèves vont donc appliquer à ces données l'algorithme EM, vérifier que le K optimal de la méthode de sélection de modèle vue en cours est bien $K = 2$ et proposer un clustering de ces observations.

6 En utilisant sklearn

Essayer de faire la même chose en utilisant la fonction **sklearn.mixture.GaussianMixture** de la bibliothèque **sklearn**. Cf http://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html