Computational statistics
TP 2 : Expectation-Maximisation algorithm – Importance sampling

Thibault de Surrel

Exercice 1

1.

On souhaite générer une variable aléatoire X qui suit la loi discrète suivante :

$$\forall i \in \{1, ..., n\},\ \mathbb{P}(X = x_i) = p_i$$

Pour cela on tire un variable aléatoire $U \sim \mathcal{U}([0, 1])$ et on cherche $i \in \{1, ..., n\}$ tel que

$$U \in \left[\sum_{j=1}^{i-1} p_j, \sum_{j=1}^{i} p_j\right[$$

Alors, on renvoie x_i.

2.3.

Sur le notebook.

Exercice 2

1.

On cherche la vraisemblance $\mathcal{L}$ d'un échantillon $(X_i)_{i\in[\![1,n]\!]}$ qui est décrit dans l'énoncé. On a donc, par indépendance de l'échantillon,

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

De plus, pour $i \in [\![1, n]\!]$, on a

$$\begin{aligned} q_\theta(x_i) &= \mathbb{E}_{Z_i}[q_\theta(x_i \mid z_i)] \quad \text{en conditionnant par rapport à } Z_i \\ &= \sum_{j=1}^{m} \mathbb{P}(Z_i = j) q_\theta(x_i \mid \{z_i = j\}) \\ &= \sum_{j=1}^{m} \alpha_j \varphi(x_i \mid \mu_j, \Sigma_j) \end{aligned}$$

où l'on note $x \mapsto \varphi(x \mid \mu_j, \Sigma_j)$ la densité de $\mathcal{N}(\mu_j, \Sigma_j)$ qui vaut :

$$\varphi(x \mid \mu_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^N \det \Sigma}} \exp\left(-\frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\right)$$

où N est la dimension de la gaussienne.

Ainsi, on a

$$\mathcal{L}(x_1, ..., x_n, \theta) = \prod_{i=1}^{n} \sum_{j=1}^{m} \alpha_j \varphi(x_i \mid \mu_j, \Sigma_j)$$

2.

Sur le notebook.

3.

Ici, on souhaite appliquer l'algorithme EM afin d'estimer le paramètre $\theta = (\mu_j, \Sigma_j, \alpha_j)_{j\in[\![1,m]\!]}$. Pour l'algorithme EM, on procède en deux étapes. Dans un premier temps, on cherche cherche à calculer la fonction B définie par

$$B(\theta, \theta_t) = \mathbb{E}_{z\sim q_{\theta_t}(\cdot|X)}\left[\log q_\theta(z, (x_i)_{i\in[\![1,n]\!]})\right]$$

où q représente la densité. Ensuite, dans un second temps, on cherche à maximiser θ_{t+1} :

$$\theta_{t+1} \in \text{argmax}_{\theta\in\Theta} B(\theta, \theta_t)$$

où Θ est l'ensemble des paramètres possibles.

Commençons par la première étape. Par indépendance de l'échantillon $(X_i)_{i\in[\![1,n]\!]}$, on a

$$\begin{aligned} B(\theta, \theta_t) &= \mathbb{E}_{z\sim q_{\theta_t}(\cdot|X)}\left[\log \prod_{i=1}^{n} q_\theta(z, x_i)\right] \\ &= \sum_{i=1}^{n} \mathbb{E}_{z\sim q_{\theta_t}(\cdot|X)}\left[\log q_\theta(z, x_i)\right] \text{le log transformant le produit en somme} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{P}(Z_i = j \mid X_i, \theta_t) \log q_\theta(z_j = j, x_i) \end{aligned} \tag{1}$$

On calcule séparément les deux termes de cette double somme. D'une part on a :

$$\begin{aligned} q_\theta(z_j = j, x_i) &= \log\left(\alpha_j \varphi(x_i \mid \mu_j, \Sigma_j)\right) \\ &= \log \alpha_j + \log \varphi(x_i \mid \mu_j, \Sigma_j) \\ &= \log \alpha_j - \frac{1}{2}\log((2\pi)^N \det \Sigma_j) - \frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) \end{aligned} \tag{2}$$

D'autre part on a :

$$\begin{aligned} \mathbb{P}(Z_i = j \mid X_i, \theta_t) &= q_{\theta_t}(z_i = j \mid x_i) \\ &= \frac{q_{\theta_t}(z_i = j, x_i)}{q_{\theta_t(x_i)}} \quad \text{par la formule de Bayes} \\ &= \frac{q_{\theta_t}(x_i \mid z_i = j) q_{\theta_t}(z_i = j)}{q_{\theta_t(x_i)}} \quad \text{par la formule de Bayes} \\ &= \frac{\varphi(x_i \mid \mu_j^t, \Sigma_j^t)\alpha_j^t}{q_{\theta_t(x_i)}} \end{aligned} \tag{3}$$

Enfin, on a, en conditionnent par Z_i

$$q_{\theta_t(x_i)} = \mathbb{E}_{Z_i}[q_{\theta_t(x_i|z_i)}] = \sum_{j=1}^{m} \varphi(x_i \mid \mu_j^t, \Sigma_j^t)\alpha_j^t$$

Ainsi, en notant $p_{i,j}^t := \mathbb{P}(Z_i = j \mid X_i, \theta_t)$, on a

$$p_{i,j}^t = \frac{\varphi(x_i \mid \mu_j^t, \Sigma_j^t)\alpha_j^t}{\sum_{j=1}^{m} \varphi(x_i \mid \mu_j^t, \Sigma_j^t)\alpha_j^t}$$

Pour conclure, on a donc, à la fin de l'étape E de l'algorithme d'EM :

$$B(\theta, \theta_t) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{i,j}^t \left(\log \alpha_j - \frac{1}{2}\log((2\pi)^N \det \Sigma_j) - \frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) \right)$$

On peut donc passer à la deuxième étape de l'algorithme : la maximisation de B selon θ. On cherche donc θ_{t+1} vérifiant

$$\theta_{t+1} \in \operatorname{argmax}_{\theta\in\Theta} B(\theta, \theta_t)$$

On le fait pour chaque paramètre $\alpha_j^{t+1}, \mu_j^{t+1}, \Sigma_j^{t+1}$ pour $j \in [\![1, m]\!]$

Pour α_j^{t+1} On a α_j^{t+1} qui est solution d'un problème d'optimisation sous la contrainte $\sum_{j=1}^{m} \alpha_j = 1$. On introduit donc le lagrangien G avec le multiplicateur de Lagrange λ :

$$G(\theta, \theta_t; \lambda) := B(\theta, \theta_t) + \lambda \left(\sum_{j=1}^{m} \alpha_j - 1 \right)$$

On a alors les dérivées partielles suivantes :

$$\frac{\partial G}{\partial \alpha_j} = \sum_{i=1}^{n} p_{i,j}^t \frac{1}{\alpha_j} + \lambda \qquad \frac{\partial G}{\partial \lambda} = \sum_{j=1}^{m} \alpha_j - 1$$

On a alors

$$\frac{\partial G}{\partial \alpha_j} = 0 \iff \alpha_j = -\frac{1}{\lambda}\sum_{i=1}^{n} p_{i,j}^t$$

Or $\sum_{j=1}^{m} \alpha_j = 1$ donc

$$-\frac{1}{\lambda}\sum_{j=1}^{m}\sum_{i=1}^{n} p_{i,j}^t = 1 \iff \lambda = -\sum_{j=1}^{m}\sum_{i=1}^{n} p_{i,j}^t = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{i,j}^t$$

De plus, on a $\sum_{j=1}^{m} p_{i,j}^t = \sum_{j=1}^{m} \mathbb{P}(Z_i = j \mid X_i, \theta_t) = 1$ ainsi, $\lambda = -n$ et donc

$$\alpha_j^{t+1} = \frac{1}{n}\sum_{i=1}^{n} p_{i,j}^t$$

Pour μ_j^{t+1} On a

$$\nabla_{\mu_j} B(\theta, \theta_j) = -\frac{1}{2}\sum_{i=1}^{n} p_{i,j}^t \nabla f_{i,j}(\mu_j)$$

où l'on note $f_{i,j} \colon \mu \mapsto (x_i - \mu)^T \Sigma_j^{-1}(x_i - \mu)$. Or on a, pour $h \in \mathbb{R}^m$

$$\begin{aligned} f_{i,j}(\mu + h) &= (x_i - \mu - h)^T \Sigma_j^{-1}(x_i - \mu - h) \\ &= (x_i - \mu)^T \Sigma_j^{-1}(x_i - \mu) - h^T \Sigma_j^{-1}(x_i - \mu) - (x_i - \mu)^T \Sigma_j^{-1} h + \underbrace{h^T \Sigma_j^{-1} h}_{=o(\|h\|)} \end{aligned} \tag{4}$$

Ainsi,

$$df_{i,j}(\mu) \cdot h = -h^T \Sigma_j^{-1}(x_i - \mu) - (x_i - \mu)^T \Sigma_j^{-1} h = \langle \underbrace{-2\Sigma_j^{-1}(x_i - \mu)}_{\nabla f_{i,j}(\mu)}, h \rangle$$

Et donc

$$\nabla_{\mu_j} B(\theta, \theta_j) = \sum_{i=1}^{n} p_{i,j}^t \Sigma_j^{-1}(x_i - \mu_j)$$

D'où

$$\nabla_{\mu_j} B(\theta, \theta_j) = 0 \iff \sum_{i=1}^{n} p_{i,j}^t x_i = \left(\sum_{i=1}^{n} p_{i,j}^t \right) \mu_j$$

Ainsi, on a

$$\mu_j^{t+1} = \frac{\sum_{i=1}^{n} p_{i,j}^t x_i}{\sum_{i=1}^{n} p_{i,j}^t}$$

Pour $(\sigma_j^{t+1})^2$ Pour ce calcul, on note $\Lambda_j = \Sigma_j^{-1}$ pour tout $j \in [\![1, m]\!]$. On a premièrement la relation suivante : $\det \Sigma_j = (\det \Lambda_j)^{-1}$. On peut alors réécrire notre fonction $B(\theta, \theta_t)$ de la manière suivante :

$$B(\theta, \theta_t) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{i,j}^t \left(\log \alpha_j + \frac{1}{2} \log((2\pi)^N \det \Lambda_j) - \frac{1}{2} (x_i - \mu_j)^T \Lambda_j (x_i - \mu_j) \right)$$

On va différentier cette relation par rapport à Λ_j. Dans un premier temps, on sait que $\nabla \det \Lambda_j = \mathrm{Com}\Lambda_j$ où $\mathrm{Com}\Lambda$ est la comatrice de Λ. Donc, $\nabla \log((2\pi)^N \det \Lambda_j) = \frac{1}{\det \Lambda_j} \mathrm{Com}\Lambda_j = \Lambda_j^{-1}$ d'après la relation $\Lambda_j \mathrm{Com}\Lambda_j^T = \det \Lambda_j I_m$ et d'après le fait que Σ_j, donc Λ_j sont symétriques. Dans un second temps, on a que, pour tout vecteurs u, v, le gradient de $u^T \Lambda_j v$ par rapport à Λ_j est la matrice uv^T. Ainsi, on a que

$$\nabla_{\Lambda_j} B(\theta, \theta_t) = \sum_{i=1}^{n} p_{i,j}^t (\frac{1}{2} \Lambda_j^{-1} - \frac{1}{2} (x_i - \mu_j)(x_i - \mu_j)^T)$$

Ainsi, comme $\Lambda_j^{-1} = \Sigma_j$, on a

$$\nabla_{\Lambda_j} B(\theta, \theta_t) = 0 \iff \left(\sum_{i=1}^{n} p_{i,j}^t \right) \Sigma_j = \sum_{i=1}^{n} p_{i,j}^t (x_i - \mu_j)(x_i - \mu_j)^T$$

et donc

$$\Sigma_j^{t+1} = \frac{\sum_{i=1}^{n} p_{i,j}^t (x_i - \mu_j^{t+1})(x_i - \mu_j^{t+1})^T}{\sum_{i=1}^{n} p_{i,j}^t}$$

On résume les formules trouvées. On a, pour l'étape de maximisation de l'algorithme EM, les formules suivantes :

$$\alpha_j^{t+1} = \frac{1}{n} \sum_{i=1}^{n} p_{i,j}^t \qquad \mu_j^{t+1} = \frac{\sum_{i=1}^{n} p_{i,j}^t x_i}{\sum_{i=1}^{n} p_{i,j}^t} \qquad \Sigma_j^{t+1} = \frac{\sum_{i=1}^{n} p_{i,j}^t (x_i - \mu_j^{t+1})(x_i - \mu_j^{t+1})^T}{\sum_{i=1}^{n} p_{i,j}^t}$$

4.

D'après les calculs effectués sur le notebook, les erreurs commises sont très faibles. L'algorithme EM arrive très bien à estimer les paramètres dans ce cas. Il faut tout de fois mentionner que les 3 gaussiennes choisies lors du calcul sont bien distinctes les unes des autres, elles ne se superposent pas. Quand les gaussiennes choisies à la base sont plus proches les unes des autres, l'erreur commise est plus élevée.

5.

Le nuage de point ne ressemble pas à une seule gaussienne. En effet, il nuage semble "tordu". Cependant, on arrive à imaginer que ce nuage a été généré par au moins deux gaussiennes, une pour la partie plus à gauche, avec une plus petite variance, et une pour la droite, avec une variance plus élevée.

6.

Après avoir fait tourner l'algorithme EM sur ce jeu de données pour différentes valeurs de m (le nombre de gaussiennes à estimer), on calcul le BIC et on voit que le nombre de gaussiennes qui minimise ce critère est 3. Lorsque que l'on regarde les résultats visuels, on se rend compte qu'effectivement, 3 gaussiennes semble bien rendre compte de notre jeu de données. Lorsque l'on simule une seule gaussienne, l'algorithme essaye de couvrir tout le nuage de point. Pour 2 et 3 gaussiennes, on voit qu'il distingue les différentes parties du nuage de points et les couvre par des gaussiennes différentes de manière assez cohérente. A partir de 4 gaussiennes, l'interprétation est plus difficile, puisque plusieurs gaussiennes se superposent et certaines ont une variance très étirée dans une direction. On peut raisonnablement penser que m est trop grand dans ce cas.

Exercice 3

1.

Sur le notebook.

2.

On calcul l'espérance et la variance de notre estimateur d'échantillonnage préférentiel de $\mathbb{E}_p[f(X)]$ que l'on note IS_n :

$$IS_n = \frac{1}{n}\sum_{i=1}^{n} \frac{p(X_i)}{q(X_i)} f(X_i)$$

Alors, pour l'espérance on a, l'échantillon $(X_1, ..., X_n)$ étant i.i.d. et de loi q :

$$\mathbb{E}_q[IS_n] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_q\left[\frac{p(X_i)}{q(X_i)} f(X_i)\right] = \mathbb{E}_q\left[\frac{p(X)}{q(X)} f(X)\right]$$

où X suit la loi q. Ainsi, on a

$$\mathbb{E}_q[IS_n] = \mathbb{E}_p[f(X)]$$

De plus, pour la variance, on a

$$\mathrm{Var}(IS_n) = \frac{1}{n}\mathrm{Var}\left(\frac{p(X)}{q(X)} f(X)\right) = \mathbb{E}_q\left[\left(\frac{p(X)}{q(X)} f(X)\right)^2\right] - \mathbb{E}_q\left[\frac{p(X)}{q(X)} f(X)\right]^2$$

Ainsi, on a

$$\mathrm{Var}(IS_n) = \mathbb{E}_p\left[\frac{p(X)}{q(X)} f(X)^2\right] - \mathbb{E}_p\left[f(X)\right]^2$$

Numériquement l'espérance de notre échantillon a l'air de converger vers une valeur (~ 0.68) et on voit que la variance diminue fortement, ce qui est bon signe. On peut donc estimer que la valeur de l'intégrale cherchée est environ 0.68.

3.

Les résultats numériques ne sont pas du tout les mêmes que lorsque $\mu = 0.8$. Cela est du au fait que le support de q et celui de p ne coïncident pas, comme on peut le voir que le graphique précédent. Or, pour que l'importance sampling fonctionne, on veut que $Supp(f \times p) \subset Supp(q)$ ce qui n'est pas le cas ici. On voit alors que les poids calculés sont très mauvais (quasi tous nuls, sauf pour les derniers qui sont très grands), car les échantillons générés vont être dans une zone où p est quasi nulle.

4.

L'étape (iii) de l'algorithme de Population Monte Carlo consiste à trouver les nouveaux paramètres $\theta_{t+1} = (\alpha^{t+1}, \mu^{t+1}, \Sigma^{t+1})$ en maximisant

$$\sum_{i=1}^{n} \tilde{\omega}_i^{(t)} \log \left(\sum_{j=1}^{m} \alpha_j \varphi(X_i^{(t)}; \theta_j^t) \right)$$

Cette quantité est une approximation de

$$\int \log \left(\sum_{j=1}^{m} \alpha_j \varphi(x; \theta_j^t) \right) \nu(x) \mathrm{d}x = \mathbb{E}_{X \sim \nu} \left[\log \left(\sum_{j=1}^{m} \alpha_j \varphi(X; \theta_j^t) \right) \right]$$

Cette quantité ressemble beaucoup à la quantité $B(\theta, \theta_t)$ de l'exercice précédent. On est donc tenté de modifier l'algorithme EM afin de calculer cette quantité. La quantité inconnue ici étant ν, on l'approche par un importance sampling et donc les formules pour les paramètres de θ_{t+1} deviennent :

$$\alpha_j^{t+1} = \sum_{i=1}^{n} \tilde{\omega}_i^{(t)} p_{i,j}^t \qquad \mu_j^{t+1} = \frac{\sum_{i=1}^{n} \tilde{\omega}_i^{(t)} p_{i,j}^t x_i^{(t)}}{\sum_{i=1}^{n} \tilde{\omega}_i^{(t)} p_{i,j}^t} \qquad \Sigma_j^{t+1} = \frac{\sum_{i=1}^{n} \tilde{\omega}_i^{(t)} p_{i,j}^t (x_i^{(t)} - \mu_j^{t+1})(x_i^{(t)} - \mu_j^{t+1})^T}{\sum_{i=1}^{n} \tilde{\omega}_i^{(t)} p_{i,j}^t}$$

où $p_{i,j}^t = \frac{\varphi(x_i^{(t)} | \mu_j^t, \Sigma_j^t) \alpha_j^t}{\sum_{j=1}^{m} \varphi(x_i^{(t)} | \mu_j^t, \Sigma_j^t) \alpha_j^t}$ ne change pas.

5.

Sur le notebook.