Projet de Computational statistics: Online Em pour les données fonctionnelles

Romero Raphael, Philbert Alexandre

alexandre.philbert@telecom-paritech.fr raphael.romero@telecom-paristech.fr

1 Introduction

Notre étude se penche sur les travaux de [1] dans lequel l'Online Em, tel qu'introduit dans [2] est utilisé pour une tâche de classification non supervisée sur des courbes. Le travail sur des données fonctionelle nécessite l'utilisation de modèles déformables qui facilite leur comparaison. Pour cela un modèle hierarchique est défini. Le but est d'apprendre les paramètres de ce modèle via l'online EM pour trouver une fonction type pour chaque classe.

2 Online EM

L'algorithme Online EM est une variante de l'algorithme Expectation-Maximization conçue pour avoir une empreinte mémoire constante, contrairement à l'EM classique où l'on doit avoir accès à toutes les observations à chaque étape. Dans le cadre de l'EM classique on dispose de toute les observations $Y_1, ..., Y_n$ stockées en mémoire, et on suppose qu'elles ont été générées de manière indépendantes et identiquement distribuées selon une certaine loi π. Les étapes de l'algorithme sont alors les suivantes :

Etape E :

$$Q_n(\theta, \theta^k) = \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[logL_{\theta^k}(Y_t, Z_t)|Y_t, \theta_n^k]$$

Etape M :

$$\theta_n^{k+1} = argmax_\theta(Q_n(\theta, \theta^k))$$

Dans le cas des modèles exponentiels, la log vraisemblance prend la forme suivante :

$$logL_\theta(Y, Z) = \langle S(Y, Z), \psi(\theta) \rangle - A(\theta)$$

Donc l'étape E revient simplement à mettre à jour la statistique exhaustive S^k :

$$S_n^{k+1} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[s(Y_t, Z_t)|Y_t, \theta_n^k]$$

et l'étape M s'en déduit en introduisant l'opérateur $\bar{\theta} : S \mapsto argmax_\theta(\langle S, \psi(\theta)\rangle - A(\theta))$:

$$\theta_n^{k+1} = \bar{\theta}(S_n^{k+1})$$

En utilisant ces notations, on remarque que par le théorème central limite, lorsque le nombre d'observations devients grand les étapes deviennent :

E :

$$S_n^{k+1} = \mathbb{E}_\pi \mathbb{E}[s(Y_1, Z_1)|Y_1, \theta_n^k]$$

M :

$$\theta_n^{k+1} = \bar{\theta}(S_n^k)$$

Dans ce contexte l'online EM vise à trouver les solutions de l'équation

$$S = \mathbb{E}_\pi \mathbb{E}[s(Y_1, Z_1)|Y_1, \bar{\theta}(S)]$$

Pour cela une procédure itérative est utilisée. A chaque nouvelle observation, la quantité $\mathbb{E}[s(Y_n, Z_n)|Y_n, \bar{\theta}(S)]$ est vue comme une version bruitée de la cible $\mathbb{E}_\pi \mathbb{E}[s(Y_1, Z_1)|Y_1, \bar{\theta}(S)$, donc on met à jour la statistique suffisante à l'aide d'une approximation stochastique : Soit (γ_n) une suite décroissante dont la série diverge et dont la série des carrés converge. On met d'abord à jour la statistique suffisante :

$$S_{n+1} = (1 - \gamma_n)S_n + \gamma_n \mathbb{E}[s(X_n, Y_n)|Y_n]$$

Puis on met à jour le paramètre : $\theta_{n+1} = \theta(\bar{S}_n)$

3 Modèle déformable pour les données fonctionnelles

On parle de données fonctionnelles lorsque l'objet que l'on cherche à estimer est une fonctionnelle, par exemple une courbe ou une forme en deux dimensions. Mathématiquement un tel objet prend la forme d'une fonction f allant d'un espace $\mathbb{U}$ vers un corps, $\mathbb{R}$ en général. Dans ce cadre les observations auxquelles ont a accès sont modélisées par le processus stochastique suivant :

$$Y(u) = \lambda f(D(u, \beta)) + \sigma W(u)$$

où λ est un paramètre d'échelle , $D(u, \beta)$ est une fonction de déformation de l'espace de départ, paramétrée par un vecteur β.

Pour simplifier le modèle, on fait l'hypothèse que la fonction d'intéret f se décompose sur une base de fonctions $(\phi_1, ..., \phi_m)$:

$$f_\alpha = \sum_{i=1}^{m} \alpha_l \phi_l$$

De plus dans la réalité on n'observe pas la totalité du processus Y, mais on n'a accès qu'à des échantillons observés à des points $u_1, ..., u_{|\Omega|}$: On obtient en notation vectorielle :

$$Y = (Y(u_1), ..., Y(u_{|\Omega|}))^T = \lambda \Phi_\beta \alpha + \sigma W$$

où $(\Phi_\beta)_{ij} = \phi_j(u_i), \alpha = (\alpha_1, ..., \alpha_m)^T, W = W(u_1), ..., W(u_\Omega)$

On voit que dans ce modèle on cherche à estimer la fonctionnelle f et ce au moyen du paramètre α, alors que les variables λ et β sont des variables cachées par rapport auxquelle

la vraisemblance complète doit être intégrée pour obtenir la vraisemblance de la variable observée.

A partir de ce modèle déformable il est possible de définir un modèle de mixture de modèles déformables. Cela est notemment utile lorsque l'on souhaite réaliser un clustering des données fonctionnelles. Pour cela une variable de classe I (cachée) à valeurs dans $1, ..., C$ est introduite. Pour chaque $j = 1, ..., C$, conditionnellement à $Y \in \mathcal{C}_j$, le modèle s'écrit $Y = \lambda \Phi_\beta \alpha_j + \sigma W$.

On se place dans un cadre bayésien où chaque observation Y est générée de la manière suivante :

-$I \sim \mathcal{M}ulti(1, (w_1, ..., w_C))$

-$\lambda \sim \mathcal{G}amma(a, \frac{1}{a})$

-$\beta | I = j \sim \mathcal{N}(0, \Gamma_j)$

Si l'on note $X = (\beta, \lambda)$ la vraisemblance complète est donc

$$L_\theta(I, Y, X) = g_\theta(Y|I, X) p_\theta(X|I) w_I$$

avec

$$g_\theta(Y|I, X) \propto exp(-\frac{1}{2\sigma^2} ||Y - \lambda \phi_\beta \alpha_I||^2)$$

et

$$p_\theta(X|I) \propto exp(-\frac{1}{2} \beta^T \Gamma_I^{-1} \beta) \lambda^{a-1} exp(-a\lambda)$$

On voit que ce modèle est un modèle exponentiel. En effet la log-vraisemblance peut s'écrire sous la forme :

$$L_\theta(I, X, Y) = t(\theta) + \langle r(\theta), S(I, X, Y) \rangle$$

La fonction de log-normalisation est :

$$t(\theta) = log \frac{a^a}{\mathcal{G}(a)} - \frac{|\Omega|}{2} log 2\pi\sigma^2 - d_\beta log 2\pi$$

, La statistique exhaustive est :

$$S(I, X, Y) = (S_j(I, X, Y))_{j=1,...,C}$$

où

$$S_j(I, X, Y) = \delta_{I,j} [1, \lambda \phi_\beta^T Y, \lambda^2 \phi_\beta^T \phi_\beta, \beta\beta^T, ||Y||^2, \lambda, log\lambda]$$

et finalement le paramètre canonique est :

$$r(\theta) = (r_1(\theta), ..., r_C(\theta))$$

avec

$$r_j(\theta) = \frac{1}{2} [2log(w_j) - logdet(\Gamma_j), \frac{2\alpha_j}{\sigma^2}, \frac{-\alpha_j \alpha_j^T}{\sigma^2}, (-\Gamma_j^{-1})^T, -\frac{1}{\sigma^2}, -2a, -2(a-1)]$$

3.1 Forme close pour la maximisation

Une fois les étapes de simulation des variables manquantes et d'approximation stochastiques effectuées, il reste à mettre à jour le paramètre $\hat{\theta}_n = (\hat{\alpha}_{1,n}, ..., \hat{\alpha}_{C,n}, \hat{\Gamma}_{1,n}, ..., \hat{\Gamma}_{C,n}, \hat{w}_{1,n}, ..., \hat{w}_{C,n}, \hat{\sigma}_n^2)$. Pour simplifier, dans les exemples que l'on traite on fait l'hypothèse que les variances Γ_j sont sphériques : $\hat{\Gamma}_j = \hat{\gamma}_j^2 I_{d_\beta}$ La maximisation consiste à résoudre le problème

$$argmax_{\theta \in \Theta}\{t(\theta) + \sum_{j=1}^{C} \langle r_j(\theta), \tilde{s}_{n,j} \rangle\}$$

sous la contrainte $\sum_{i=1}^{C} w_i = 1$, où $\tilde{s}_{n,j} = (\tilde{s}_{n,j}^{(1)}, ..., \tilde{s}_{n,j}^{(7)})$ est la statistique exhaustive mise à jour à l'étape d'approximation stochastique. Par conséquent l'étape de maximisation s'écrit : Pour $j = 1, ..., C$:

$$\hat{w}_j = \frac{\tilde{s}_{n,j}^{(1)}}{\sum_{j'=1}^{C} \tilde{s}_{n,j'}^{(1)}}$$

$$\hat{\gamma}_j^2 = \frac{\tilde{s}_{n,j}^{(4)}}{d_\beta \tilde{s}_{n,j}^{(1)}}$$

$$\hat{\alpha}_j = (\tilde{s}_{n,j}^{(3)}))^{-1} \tilde{s}_{n,j}^{(2)}$$

$$\hat{\sigma}^2 = \frac{1}{|\Omega|} \sum_{j=1}^{C} (-2\hat{\alpha}_j^T \tilde{s}_{n,j}^{(2)} + tr(\hat{\alpha}_j \hat{\alpha}_j^T \tilde{s}_{n,j}^{(3)}) + \tilde{s}_{n,j}^{(5)})$$

4 Exemple numérique

Nous avons tenté de reproduire l'exemple réalisé dans l'article de F.Maire et al.(2017), basé sur la base de données Berkeley Growth Study. Celle-ci contient les enregistrements des vitesses de croissances de 39 garçons et 54 filles entre 2 et 18 ans.

On considère dans ce problème deux clusters, un pour les garçons et un pour les filles. Le but du problème est de déterminer la fonction liant l'age à la vitesse de croissance pour chaque sexe. L'espace de départ est donc inclus dans $\mathbb{R}$. La base de fonction considérée est de la forme $\phi_l(u) = exp(\frac{(u-r_l)^2}{v_l^2})$ où les r_l sont des points de repère régulièrement espacés, et v_l sont les largeurs de bandes associées aux fonctions de la base, calculées par

$$v_l = -min_{i, u_i \neq r_l} \frac{||r_l - u_i||^2}{log\epsilon}$$

.

La fonction de déformation temporelle est modélisée par $D(u, \beta) = u_i + (u_f - u_i)H(u, \beta)$ avec

$$H(u, \beta) = \frac{\int_{u_i'}^{u} exp(\sum_{k=1}^{d_\beta} \beta_k \Psi_k(\nu))}{\int_{u_i'}^{u_f'} exp(\sum_{k=1}^{d_\beta} \beta_k \Psi_k(\nu))}$$

où l'on a pris $u_i' \leq u_i$, $u_f \geq u_f$ et les $\Psi_k(\nu) = exp(-\frac{1}{2}(\nu - q_k)^2)$ sont des noyaux gaussiens centrés en des points de repères q_k régulièrement espacés. Cependant en pratique lors de

l'étape de simulation des variables manquantes, la mise à jour des pseudo-prior suppose le calcul approché des intégrales définissant H. Par conséquent cela allonge fortement le temps nécessaire à la simulation des données manquantes. Pour remédier à cela une possibilité est de remplacer les noyaux gaussiens par des splines d'ordre 1 (fonctions "triangles" approximant les noyaux gaussiens), centrées en les mêmes points d'intérets et de même largeur de bande. Cela permet d'obtenir une expression de H en forme close et donc de mettre à jour les pseudo-priors sans avoir à approximer numériquement des intégrales.

Références

[1] Florian Maire, Eric Moulines, and Sidonie Lefebvre. Online em for functional data. *Comput. Stat. Data Anal.*, 111(C) :27–47, July 2017.

[2] Olivier Cappé and Eric Moulines. On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society Series B*, 71(3) :593–613, 2009.