Classification de données de régression de grande dimension.

Emilie Devijver [1]

[1] *Laboratoire de Mathématiques UMR 8628, Université Paris-Sud 11, F-91405 Orsay cedex, emilie.devijver@math.u-psud.fr*

Résumé. Les modèles de mélange en régression sont utilisés pour modéliser la relation qui existe entre la réponse et les prédicteurs, lorsque ces données proviennent de différentes classes. Avec l'augmentation des données de grande dimension, les modèles doivent aujourd'hui tenir compte des problèmes entrainés. Durant cet exposé, nous proposerons une procédure de classification non supervisée en grande dimension. Nous reprenons l'usage de la pénalisation ℓ_1 pour sélectionner les variables pertinentes, et pour construire une collection de modèles de mélange, obtenue en faisant varier le paramètre de régularisation. Nous estimons les paramètres de chaque modèle par maximum de vraisemblance, puis nous sélectionnons un modèle grâce à un critère pénalisé ℓ_0 non asymptotique, construit à partir des données, suivant l'heuristique de pente de Birgé et Massart. Nous validerons cette procédure sur des donnnées fonctionnelles, exemple typique de données de grande dimension. Notre procédure est appliquée à la projection sur une base d'ondelettes de ces fonctions, de façon à garder l'aspect fonctionnel.

Mots-clés. Modèle de mélange, régression, grande dimension, sélection de modèles, données fonctionnelles.

Abstract. Mixture models in régression are used to represent the relation between the response and the regressors, when these data belong to different clusters. With the increasing of high-dimensional data, models need to take into account high-dimensional issues. During this talk, we will propose a clustering procedure in high-dimension. We will use the ℓ_1-penalization to select relevant variables, and to construct a model collection of mixture models, get in varying the regularization parameter. We estimate the parameters by maximum likelihood approach, and select a model thanks to a data-driven criterion, non asymptotic, adapted from the slope heuristic of Birgé and Massart. We will apply this procedure on functional data, which is a good example of high-dimensional data. Denote that we apply our procedure to the projection of each function onto some wavelet basis, in order to keep the function structure.

Keywords. Model-based clustering, régression, high-dimension, model selection, functional data.

1 Introduction

Soit $(x_i, y_i)_{1\leq i\leq n} \in \mathbb{R}^p \times \mathbb{R}^q$ un échantillon issu de variables aléatoires notées (X, Y). On veut regrouper les observations pour lesquelles les variables $Y|X$ ont un comportement similaire. On choisit comme modèle les mélanges finis de gaussiennes en régression. Ce modèle a été développé dans un article de P. Bülhman et al (2010) dans le cas où Y est scalaire, et X est multidimensionnelle. On souhaite généraliser cette étude dans le cas où Y est multivariée, X étant toujours multidimensionnelle. On suppose de plus que les données sont de grande dimension ($p \times q > n$). Une idée naturelle serait d'utiliser l'estimateur du Lasso pour pallier à ce problème. Cependant, le choix du paramètre de régularisation est un problème compliqué. Caroline Meynet, dans sa thèse, propose une procédure dans le cas d'estimation de densité en grande dimension, en construisant une collection de modèles de parcimonies différentes. Nous utilisons la trame de cette procédure pour construire la nôtre, dans un cadre de régression.

2 Modèles de mélanges Gaussiens en régression

On observe n couples indépendants $(x_i, y_i)_{1\leq i\leq n} \in \mathbb{R}^p \times \mathbb{R}^q$, de densité conditionnelle inconnue s_0. On suppose que ces données proviennent d'un modèle de mélanges de modèles linéaires Gaussien : si le couple de variables aléatoires (X, Y) appartient au groupe r,

$$Y = \beta_r X + \epsilon \tag{1}$$

où ϵ est un bruit blanc. On obtient donc que $Y|X = x \sim s_\xi(y|x)dy$, avec

$$s_\xi(y|x) = \sum_{r=1}^{k} \frac{\pi_r}{(2\pi)^{\frac{q}{2}} \det(\Sigma_r)^{1/2}} \exp\left(-\frac{(y-\beta_r x)^t \Sigma_r^{-1}(y-\beta_r x)}{2}\right);$$

$$\xi = (\pi_1, \ldots, \pi_k, \beta_1, \ldots, \beta_k, \Sigma_1, \ldots, \Sigma_k) \in \left(\Pi_k \times (\mathbb{R}^{q\times p})^k \times (\mathbb{S}_{++}^q)^k\right);$$

$$\Pi_k = \left\{(\pi_1, \ldots, \pi_k); \pi_r > 0 \text{ pour } r \in \{1, \ldots, k\} \text{ et } \sum_{r=1}^{k} \pi_r = 1\right\};$$

$\mathbb{S}_{++}^q$ est l'ensemble des matrices symétriques définies positives sur $\mathbb{R}^q$.

Conformément à ce qui a été fait dans le cas de la réponse scalaire dans l'article de P. Bühlman et al, on reparamétrise notre modèle de façon à avoir un estimateur du maximum de vraisemblance invariant en échelle, et d'obtenir un problème d'optimisation convexe. On définit alors ${}^t P_r P_r = \Sigma_r^{-1}$ et $\Phi_r = P_r \beta_r$. On obtient $Y_i|X_i = x_i \sim h_\theta(y|x_i)dy$, pour

$i = 1 \dots n$, avec

$$h_\theta(y|x) = \sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{q/2}} \exp\left(-\frac{(P_r y - \Phi_r x)^t (P_r y - \Phi_r x)}{2}\right)$$

$$\theta = (\pi_1, \dots, \pi_k, \Phi_1, \dots, \Phi_k, P_1, \dots, P_k) \in \left(\Pi_k \times (\mathbb{R}^{q \times p})^k \times T_q^k\right)$$

$$\Pi_k = \left\{(\pi_1, \dots, \pi_k); \pi_r > 0 \text{ pour } r \in \{1, \dots, k\} \text{ et } \sum_{r=1}^{k} \pi_r = 1\right\}$$

T_q est l'ensemble des matrices triangulaires inférieures avec des coefficients diagonaux non nuls.

La log-vraisemblance associée à notre échantillon est

$$l(\theta) = \sum_{i=1}^{n} \log\left(\sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{q/2}} \exp\left(-\frac{(P_r Y_i - X_i \Phi_r)^t (P_r Y_i - X_i \Phi_r)}{2}\right)\right)$$

et l'estimateur du maximum de vraisemblance est défini par

$$\hat{\theta}_0 := \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{-\frac{1}{n} l(\theta)\right\}.$$

Nous définissons aussi un estimateur du maximum de vraisemblance pénalisé par une norme ℓ_1

$$\hat{\theta}_\lambda := \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{-\frac{1}{n} l(\theta) + \lambda \sum_{r=1}^{k} \pi_r ||\Phi_r||_1\right\}$$

où $||\Phi_r||_1 = \sum_{j=1}^{p} \sum_{z=1}^{q} |\Phi_{j,z,r}|$, et avec λ à spécifier. Cet estimateur ressemble à l'estimateur du Lasso, introduit par Tibshirani en 1996. La différence réside dans la pénalité. Initialement, le Lasso ne pénalisait que les moyennes conditionnelles. Pour permettre d'étudier un mélange de gaussiennes de variances différentes, il faut plutôt pénaliser par Φ.

Pour résoudre ces problèmes d'optimisation, nous généralisons l'algorithme EM décrit par P. Bühlman et al. L'algorithme EM, introduit par Dempster (1977), permet de calculer une estimation des paramètres d'un modèle de mélange. On alterne deux étapes, la première calculant la logvraisemblance complétée avec les valeurs courantes des paramètres, et la deuxième étape maximisant la logvraisemblance par rapport aux paramètres. L'algorithme décrit par P. Bühlman et al. généralise celui de Dempster pour l'estimateur du Lasso, et nous l'étendons pour X et Y multivariés. On obtient comme calculs, à l'itération ite+1, pour tout $j \in \{1, \dots, p\}, r \in \{1, \dots, k\}, z \in \{1, \dots, q\}$,

- à l'étape E:

$$\hat{\gamma}_{i,r} = \frac{\pi_r^{(\text{ite})} \left(\prod_{z=1}^{q} \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P_r^{(\text{ite})} y_i - x_i \Phi_r^{(\text{ite})}\right)^t \left(P_r^{(\text{ite})} y_i - x_i \Phi_r^{(\text{ite})}\right)\right)}{\sum_{l=1}^{k} \pi_l^{(\text{ite})} \left(\prod_{z=1}^{q} \rho_{r,z}^{(\text{ite})}\right) \exp\left(-\frac{1}{2}\left(P_r^{(\text{ite})} y_i - x_i \Phi_l^{(\text{ite})}\right)^t \left(P_r^{(\text{ite})} y_i - x_i \Phi_l^{(\text{ite})}\right)\right)}$$

- à l'étape M:

$$
\begin{aligned}
(\tilde{y}_{i,r}, \tilde{x}_{i,r}) &= \sqrt{\hat{\gamma}_{i,r}}(y_i, x_i) \\
n_r &= \sum_{i=1}^{n} \hat{\gamma}_{i,r} \\
\Delta &= (-n_r \langle \tilde{y}_{i,z,r}, \Phi_{r,z} \tilde{x}_{i,r} \rangle)^2 - 4||\tilde{y}_z||_2^2 \\
S_{r,j,z}^{(\text{ite})} &= -\sum_{i=1}^{n} \tilde{X}_{i,r,j} \rho_{r,z}^{(\text{ite})} \tilde{Y}_{i,r,z} + \sum_{j_2=1, j_2 \neq j}^{p} \tilde{x}_{i,r,j} \tilde{x}_{i,r,j_2} \Phi_{r,j_2,z}^{(\text{ite})}; \\
\pi_r^{(\text{ite}+1)} &= \pi_r^{(\text{ite})} + t^{(\text{ite})} \left(\frac{\sum_{i=1}^{n} \hat{\gamma}_{i,r}}{n} - \pi_r^{(\text{ite})} \right); \\
\rho_r^{(\text{ite}+1)} &= \frac{n_r \langle \tilde{y}_{i,z,r}, \Phi_{r,z}^{(\text{ite})} \tilde{x}_{i,r} \rangle + \sqrt{\Delta}}{2 n_r ||\tilde{y}_z||_2^2}; \\
\Phi_{r,j,z}^{(\text{ite}+1)} &= \begin{cases} \frac{-S_{r,j,z}^{(\text{ite})} + n\lambda(\pi_r^{(\text{ite})})}{||\tilde{x}_{r,j}||_2^2} & \text{si } S_{r,j,z}^{(\text{ite})} > n\lambda\pi_r^{(\text{ite})}; \\ -\frac{S_{r,j,z}^{(\text{ite})} + n\lambda(\pi_r^{(\text{ite})})}{||\tilde{x}_{r,j}||_2^2} & \text{si } S_{r,j,z}^{(\text{ite})} < -n\lambda\pi_r^{(\text{ite})}; \\ 0 & \text{sinon.} \end{cases}
\end{aligned}
$$

et $t^{(m)} \in (0, 1]$, la plus grande valeur dans la grille $\{\delta^k, k \in \mathbb{N}\}$, $0 < \delta < 1$, telle que la fonction soit décroissante.

Pour initialiser notre algorithme, nous initialisons les affectations par un algorithme type k-means sur les couples (x, y), puis nous calculons des estimations des paramètres dans chaque classe ainsi construite, le modèle étant linéaire. Puis nous faisons tourner notre algorithme EM 10 fois, pour observer le comportement de la vraisemblance. Nous répétons cette méthode 50 fois, et gardons les valeurs initiales qui maximise la vraisemblance. Pour arrêter notre algorithme, nous attendons la convergence de la vraisemblance et celle des paramètres à estimer. Pour assurer que ce ne soit pas un minimum local, nous faisons tourner notre procédure un minimum de fois. Pour assurer la fin de l'algorithme, nous imposons un nombre maximum d'itérations.

3 La procédure Lasso-EMV

Cette procédure peut être décomposée en trois étapes principales : on construit une collection de modèles, pour chaque modèle on calcule l'estimeur du maximum de vraisemblance, puis on choisit le meilleur parmi ces modèles.

- On construit une collection de modèles notée $\{\mathcal{H}_{(k,J)}\}_{(k,J)\in\mathcal{M}}$ dans laquelle $\mathcal{H}_{(k,J)}$ est défini par l'équation

$$\mathcal{H}_{(k,J)} = \{y \in \mathbb{R}^q | x \in \mathbb{R}^p \mapsto h_\theta(y|x)\} \tag{2}$$

où

$$h_\theta(y|x) = \sum_{r=1}^{k} \frac{\pi_r \det(P_r)}{(2\pi)^{q/2}} \exp\left(-\frac{(P_r y - \Phi_r x^{[J]})^t (P_r y - \Phi_r x^{[J]})}{2}\right),$$

et

$$\theta = (\pi_1, \ldots, \pi_k, \Phi_1, \ldots, \Phi_k, \rho_1, \ldots, \rho_k) \in \Pi_k \times \left(\mathbb{R}^{|J|}\right)^k \times \left(\mathbb{R}_+^q\right)^k.$$

La collection de modèles est indexée par $\mathcal{M} = K \times \mathcal{J}$. On note $K \subset \mathbb{N}^*$ l'ensemble des nombres de composantes possibles. Notons aussi $\mathcal{J}$ une collection de sous-ensembles de $\{1, \ldots, p\} \times \{1, \ldots, q\}$. Pour trouver les variables actives, et construire l'ensemble J, on pénalise le contraste empirique par une pénalité ℓ_1 sur les paramètres de moyennes $||\Phi_r||_1 = \sum_{j=1}^{p} \sum_{z=1}^{q} |\Phi_{r,j,z}|$ pour tout $r \in \{1, \ldots, k\}$. Dans les procédures de type ℓ_1, le choix du paramètre de régularisation est souvent difficile : si on fixe le nombre de composantes $k \in K$, on peut construire une grille G_k de paramètres de régularisation en utilisant les formules de mise à jour des paramètres dans l'algorithme EM:

$$\Phi_{r,j,z} = 0 \quad \Leftrightarrow \quad \lambda_{r,j,z} = \frac{|S_{r,j,z}|}{n\pi_r}.$$

On obtient ainsi une formule pour λ, le paramètre de régularisation, dépendant du coefficient que l'on veut annuler. On peut calculer les coefficients de cette grille grâce à l'estimateur du maximum de vraisemblance.

Alors, pour chaque $\lambda \in G_k$, on peut approcher l'estimateur du Lasso défini par

$$\hat{\theta}^L_{(k,J)} = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log(h_\theta(y_i|x_i)) + \lambda \sum_{r=1}^{k} \pi_r ||\Phi_r||_1 \right\}$$

à l'aide de l'algorithme EM. Ainsi, pour tout $k \in K$, et pour tout $\lambda \in G_k$, on a construit l'ensemble des variables actives J. On note $\mathcal{J}$ la collection de tous ces ensembles.

- La deuxième étape consiste à approcher l'estimateur du maximum de vraisemblance, restreint aux variables actives,

$$\hat{h}_{(k,J)} = \underset{t \in \mathcal{H}_{(k,J)}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log(t(y_i|x_i)) \right\}$$

en utilisant l'algorithme EM pour chaque $(k, J) \in \mathcal{M}$.

- La troisième étape consiste à selectionner un modèle. On utilise l'heuristique des pentes décrites dans l'article de Birgé et Massart (2007). D'abord, on regroupe les modèles par leur dimension D, pour obtenir une collection de modèles $\{\mathcal{H}_D\}_{D \in \mathcal{D}}$.

La dimension d'un modèle correspond au nombre de paramètres à estimer dans ce modèle. Pour chaque dimension D, notons $\hat{h}_D$ l'estimateur maximisant la vraisemblance parmi les estimeurs associés aux modèles de dimenion D. Alors, la fonction $D/n \mapsto \frac{1}{n}\sum_{i=1}^{n} \log(\hat{h}_D)$ a un comportement linéaire pour les grandes dimensions. On estime la pente, notée $\hat{\kappa}$. Alors, on sélectionne le minimiseur $\hat{D}$ du critère pénalisé $-\frac{1}{n}\sum_{i=1}^{n} \log(\hat{h}_D) + 2\hat{\kappa}D/n$, et l'estimateur sélectionné est $\hat{h}_{(k_{\hat{D}},J_{\hat{D}})}$.

4 Simulations

On voudrait appliquer notre procédure à des données fonctionnelles. Pour tirer partie de la structure fonctionnelle, on préfère projetter sur une base orthonormale. On cherche des coefficients parcimonieux, la méthode sous-jacente à notre procédure incite donc à choisir les ondelettes. De plus, grâce aux ondelettes, on pourra comprendre quel est l'élément dérminant pour différencier les observations dans les groupes.

Durant la présentation, on appliquera notre procédure sur des données simulées, pour justifier l'utilisation de chaque critère. Puis on l'appliquera sur un jeu de données fonctionnelles électriques. En posant X la courbe de charge d'un jour j, et Y la courbe de charge d'un jour $j+1$, on cherche à regrouper les couples de jours. La structure semaine / week-end doit resortir.

Bibliographie

[1] Birgé, Massart (2007), Minimal penalties for Gaussian model selection, *Probability Theory Related Fields.*
[2] Meynet, Maugis (2012), A sparse variable selection procedure in model-based clustering, *Rapport de recherche INRIA.*
[3] Van de Geer, Bühlman, Städler (2000), ℓ_1 penalization for mixture regression models, *Test*, 19(2):209-256.