



UNIVERSITÉ
CAEN
NORMANDIE

UNIVERSITÉ DE CAEN NORMANDIE



Régression à l'aide du modèle de Markov caché

*Auteurs : S. Blin A. Bourjal
C. Champarou
M2 Statistiques Appliquées et
Analyse Décisionnelle*

Tuteur projet : M. F. CHAMROUKHI

Année universitaire 2018-2019

Introduction

Les modèles de Markov cachés constituent un outil naturel pour traiter l'un des problèmes fondamentaux de la modélisation stochastique : de nombreux processus stochastiques générés naturellement présentent une hétérogénéité temporelle dictée par un facteur sous-jacent, (mais non observable) changement dans le système de génération de signal.

La force du HMM semble être due à sa capacité à reconnaître les relations entre les régimes changeants sur un court base de terme, on pourrait modéliser de manière adéquate les données observées par un processus.

Une autre source de force des HMM est leur capacité exceptionnelle à incorporé les caractéristiques structurelles du phénomène à l'étude dans les caractéristiques structurelles du modèle. processus que la science sous-jacente peut justifier. Le HMM a été appliqué avec succès dans de nombreux contextes scientifiques avec un niveau croissant de complexité du modèle qui suit l'augmentation des disponibilités puissance de calcul.

Dans cette section, nous introduisons une extension des HMM à l'analyse de régression que nous appelons la régression de modèle de Markov cachée (HMMR). Comme suggéré par son nom, le HMMR est un modèle qui relie la variable dépendante \mathbf{Y} à un ensemble de variables indépendantes $(\mathbf{X}_1; \dots; \mathbf{X}_p)$ à travers un certain nombre de plans de régression avec des valeurs de paramètre de régression distinctes.

Table des matières

1	Paramètre	3
2	Estimation	3
2.1	Estimation par Maximum de vraisemblance	3
2.2	Estimation par l'Algorithme EM	3
2.2.1	Etape E	4
2.2.2	Etape M	5
3	Application	7

Modèle

On considère une série temporelle $(\mathbf{Y}_1, \dots, \mathbf{Y}_n)$ régie par un processus discret latent $(\mathbf{Z}_1, \dots, \mathbf{Z}_n)$ ou \mathbf{Z}_t représente l'état (discret) à l'instant t ($t = 1, \dots, n$). On dispose d'une série observée $(\mathbf{y}_1, \dots, \mathbf{y}_n)$. L'objectif est de prédire et de segmenter la série sur la base du modèle de Markov Caché.

$$f(\mathbf{y}_i | t; \Psi) = \sum_{k=1}^K \alpha_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}\beta_k, \sigma_k^2 \mathbf{I}_m), \quad (1)$$

1 Paramètre

Soient :

1. $\mathbf{p} = (\mathbf{p}_1, \dots, \mathbf{p}_K)$ la distribution initiale, où $\mathbf{p}_k = \mathbf{p}(z_1 = k)$; k est dans $\{1, \dots, \mathbf{K}\}$.
2. \mathbf{A} la matrice des probabilités de transition, où $\mathbf{A}_k = \mathbf{p}(z_t = k | z_{t-1} = l)$ pour $t = 2, \dots, n$, satisfaisant $\sum_{k=1}^K (\mathbf{A}_{lk}) = 1$.
3. l'ensemble des paramètres (Ψ_1, \dots, Ψ_K) du paramétrique conditionnel densités de probabilité des données observées $\mathbf{p}(\mathbf{y}_t | \mathbf{z}_t = k; \Psi_K)$ pour $t = 1, \dots, n$ et $k = 1, \dots, \mathbf{K}$. Ces probabilités sont aussi appelées la probabilités d'émission.

2 Estimation

Soit $\Psi = (\pi, \mathbf{A}, \Psi_1, \dots, \Psi_K)$: le vecteur de paramètre du modèle à estimé.

2.1 Estimation par Maximum de vraisemblance

$$\mathcal{L}(\Psi) = \log \mathbf{p}(\mathbf{Y}; \Psi) = \log \sum_z \mathbf{p}(\mathbf{Y}, z; \Psi) \quad (2)$$

$$\mathcal{L}(\Psi) = \log \sum_{z_1, \dots, z_n} \mathbf{p}(z_1, \pi) \prod_{t=2}^n \mathbf{p}(z_t | z_{t-1}; \mathbf{A}) \prod_{t=1}^n \mathbf{p}(\mathbf{y}_t | z_t; \Psi). \quad (3)$$

Cette log-vraisemblance est difficile à maximiser directement, d'où l'utilisation d'un algorithme performant qui est nommé l'algorithme d'expectation maximisation (EM).

2.2 Estimation par l'Algorithme EM

L'algorithme espérance-maximisation (en anglais expectation-maximization algorithm, souvent abrégé EM) est un algorithme itératif qui permet de trouver les paramètres du maximum de vraisemblance d'un modèle probabiliste lorsque ce dernier dépend de variables latentes non observables.

De nombreuses variantes ont par la suite été proposées, formant une classe entière d'algorithmes.

Avant d'aborder la notion de cet algorithme, il est très important d'aborder la notion de la vraisemblance des données complètes :

$$\begin{aligned}
p(\mathbf{Y}, z; \Psi) &= p(z_1; \pi) \prod_{t=2}^n p(z_t | z_{t-1}; \mathbf{A}) \prod_{t=1}^n p(\mathbf{y}_t | z_t; \Psi) \\
&= \prod_{k=1}^K \pi_k^{z_{1k}} \prod_{t=2}^n \prod_{k=1}^K \prod_{l=1}^K \mathbf{A}_{lk}^{z_{t-1,l} z_{tk}} \prod_{t=1}^n \prod_{k=1}^K p(\mathbf{y}_t | z_t = k; \Psi_k)^{z_{tk}}
\end{aligned} \tag{4}$$

Ce qui nous donne ensuite la notion de la log-vraisemblance des données complètes :

$$\mathcal{L}(\Psi) = \sum_{k=1}^K z_{1k} \log \pi_k + \sum_{t=2}^n \sum_{k=1}^K \sum_{l=1}^K z_{tk} z_{t-1,l} \log \mathbf{A}_{lk} + \sum_{t=1}^n \sum_{k=1}^K z_{tk} \log p(\mathbf{y}_t | z_t = k; \Psi_k) \tag{5}$$

2.2.1 Etape E

L'étape E calcule une estimation de la log-vraisemblance des données complètes :

$$\begin{aligned}
Q(\Psi, \Psi^{(q)}) &= \mathbb{E}[\mathcal{L}_c(\Psi) | \mathbf{Y}; \Psi^{(q)}] \\
&= \sum_{k=1}^K \mathbb{E}[z_{1k} | \mathbf{Y}; \Psi^{(q)}] \log \pi_k + \sum_{t=2}^n \sum_{k=1}^K \sum_{l=1}^K \mathbb{E}[z_{tk} z_{t-1,l} | \mathbf{Y}; \Psi^{(q)}] \log \mathbf{A}_{lk} \\
&\quad + \sum_{t=1}^n \sum_{k=1}^K \mathbb{E}[z_{tk} | \mathbf{Y}; \Psi^{(q)}] \log p(\mathbf{y}_t | z_t = k; \Psi_k) \\
&= \sum_{k=1}^K p(z_1 = k | \mathbf{Y}; \Psi^{(q)}) \log \pi_k + \sum_{t=2}^n \sum_{k=1}^K \sum_{l=1}^K p(z_t = k, z_{t-1} = l | \mathbf{Y}; \Psi^{(q)}) \log \mathbf{A}_{lk} \\
&\quad + \sum_{t=1}^n \sum_{k=1}^K p(z_t = k | \mathbf{Y}; \Psi^{(q)}) \log p(\mathbf{y}_t | z_t = k; \Psi_k) \\
&= \sum_{k=1}^K \tau_{1k}^{(q)} \log \pi_k + \sum_{t=2}^n \sum_{k=1}^K \sum_{l=1}^K \epsilon_{tlk}^{(q)} \log \mathbf{A}_{lk} + \sum_{t=1}^n \sum_{k=1}^K \tau_{tk}^{(q)} \log p(\mathbf{y}_t | z_t = k; \Psi_k)
\end{aligned} \tag{6}$$

Ainsi, dans l'équation ci-dessus, il nous faut calculer les probabilités à postériori de $\epsilon_{tlk}^{(q)}$ et $\tau_{tk}^{(q)}$. Ces dernières se calculent par les récursions du forward-backward.

Forward-Backward

Le processus du forward calcule récursivement les probabilités :

$$\alpha_{tk} = \mathbf{p}(\mathbf{y}_1, \dots, \mathbf{y}_t, z_t = \mathbf{k}; \Psi) \quad (7)$$

Dans un autre temps, le processus du backward calcul les probabilités à partir des temps $(t+1, \dots, n)$

$$\beta_{tk} = \mathbf{p}(\mathbf{y}_{t+1}, \dots, \mathbf{y}_n, z_t = \mathbf{k}; \Psi) \quad (8)$$

De ce fait la probabilité à postériori de l'état \mathbf{k} à un temps t donne la totalité des observations \mathbf{Y} et les paramètres du modèle Ψ

$$\tau_{tk} = \mathbf{p}(z_t = \mathbf{k} | \mathbf{Y}) = \frac{\mathbf{p}(\mathbf{Y}, z_t = \mathbf{k})}{\mathbf{p}(\mathbf{Y})} = \frac{\alpha_{tk}\beta_{tk}}{\sum_{l=1}^K \alpha_{tl}\beta_{tl}} \quad (9)$$

Les probabilités à postériori jointes de l'état \mathbf{k} à un temps t et l'état l à l'état $t-1$ donne la sequence d'observations :

$$\begin{aligned} \epsilon_{tlk} &= \mathbf{p}(z_t = \mathbf{k}, z_{t-1} = l | \mathbf{Y}) \\ &= \frac{\mathbf{p}(z_t = \mathbf{k}, z_{t-1} = l, \mathbf{Y})}{\mathbf{p}(\mathbf{Y})} \\ &= \frac{\alpha_{(t-1)l} \mathbf{p}(\mathbf{y}_t | z_t = \mathbf{k}) \beta_{tk} \mathbf{A}_{lk}}{\sum_{l=1}^K \sum_{k=1}^K \alpha_{(t-1)l} \mathbf{p}(\mathbf{y}_t | z_t = \mathbf{k}) \beta_{tk} \mathbf{A}_{lk}} \end{aligned} \quad (10)$$

Les probabilités à postérieures sont ensuite implémenté dans la fonction forward backward sous la forme suivante :

$$\tau_{tk}^{(q)} = \frac{\alpha_{tk}^{(q)} \beta_{tk}^{(q)}}{\sum_{k=1}^K \alpha_{tk}^{(q)} \beta_{tk}^{(q)}} \quad (11)$$

et

$$\epsilon_{tlk}^{(q)} = \frac{\alpha_{(t-1)l}^{(q)} \mathbf{p}(\mathbf{y}_t | z_t = \mathbf{k}; \theta^{(q)}) \beta_{tk}^{(q)} \mathbf{A}_{lk}^{(q)}}{\sum_{l=1}^K \sum_{k=1}^K \alpha_{(t-1)l}^{(q)} \mathbf{p}(\mathbf{y}_t | z_t = \mathbf{k}; \Psi) \beta_{tk}^{(q)} \mathbf{A}_{lk}^{(q)}} \quad (12)$$

2.2.2 Etape M

L'étape M actualise le paramètre Ψ en évaluant le paramètre $\Psi^{(q+1)}$ maximisé par la Q-fonction. Cette dernière peut être décomposé comme :

$$Q(\Psi, \Psi^{(q)}) = Q_{\pi}(\pi, \Psi^{(q)}) + Q_{\mathbf{A}}(\mathbf{A}, \Psi^{(q)}) + \sum_{k=1}^K Q_{\Psi_k}(\Psi, \Psi^{(q)}) \quad (13)$$

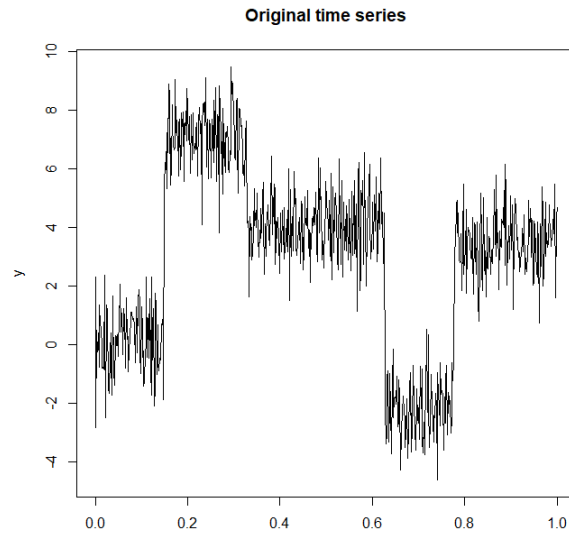
avec

$$\begin{aligned}
Q_{\pi}(\pi, \Psi^{(q)}) &= \sum_{k=1}^K \tau_{1k}^{(q)} \log \pi_k \\
Q_{\mathbf{A}}(\mathbf{A}, \Psi^{(q)}) &= \sum_{t=2}^n \sum_{k=1}^K \sum_{l=1}^K \epsilon_{tlk}^{(q)} \log \mathbf{A}_{lk} \\
Q_{\Psi_k}(\Psi, \Psi^{(q)}) &= \sum_{t=1}^n \tau_{tk}^{(q)} \log p(\mathbf{y}_t | z_t = \mathbf{k}; \Psi_k)
\end{aligned} \tag{14}$$

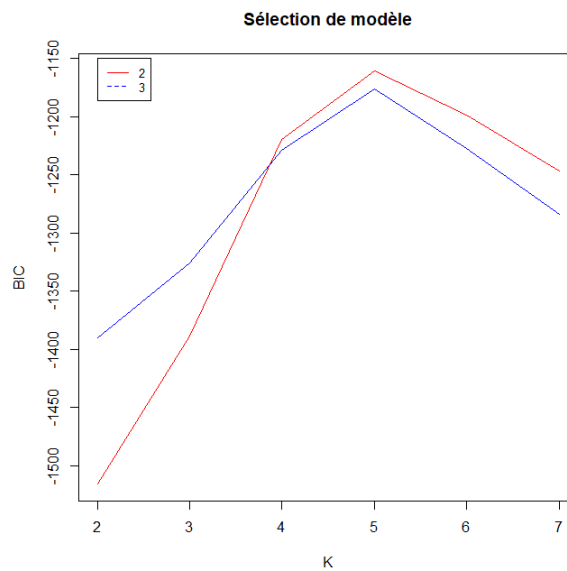
Ansi, la maximisation de $Q(\Psi, \Psi^{(q)})$ est réalisé par la maximisation (séparement) de $Q_{\pi}(\pi, \Psi^{(q)})$, $Q_{\mathbf{A}}(\mathbf{A}, \Psi^{(q)})$ et $Q_{\Psi_k}(\Psi, \Psi^{(q)})$ avec $(\mathbf{k} = 1, \dots, \mathbf{K})$. Ces maximisations sont réalisés and utilisant le Lagrangien.

3 Application

Dans cette section, nous présenterons l'utilisation de la régression à l'aide du modèle de Markov sur des données simulées. Celle-ci sont représenté sur le graphe suivant :

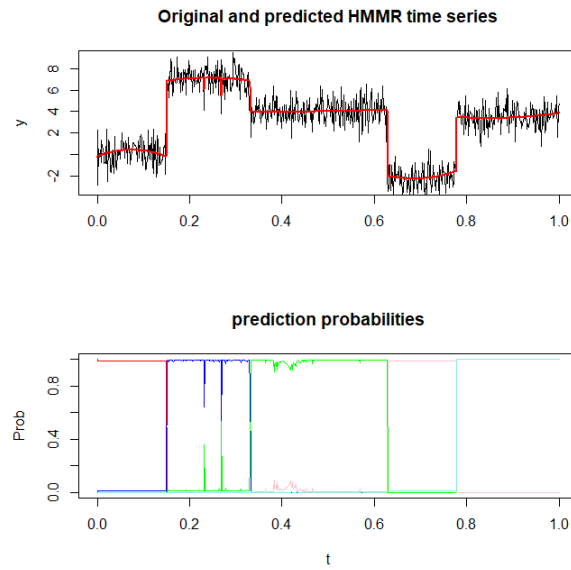


Nous pouvons constater que le graphe peut être séparé en 5 parties distinctes. Nous supposons donc que cela représentera 5 états différents dans l'exemple que nous étudions sur ces données simulées. Afin de confirmer nos conjectures, nous utilisons le Bayesian information criterion (BIC) afin de déterminer le nombre de classes et l'ordre de notre regression polynomiale.

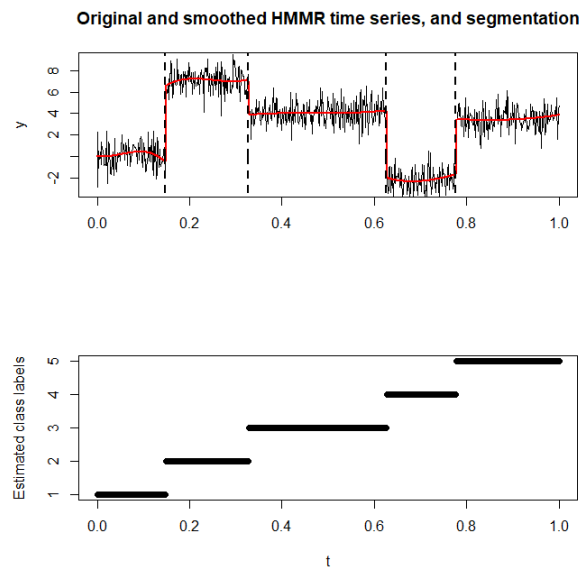


Dans ce graphique, on peut constater qu'il y a un pic en $K=5$ pour les 2 lignes. Ce pic confirme notre hypothèse faite précédemment sur le nombre de classe. On choisira l'ordre de notre regression polynomiale égale à 2 car c'est la valeur la plus élevée pour $K=5$.

Pour la suite de notre étude, on choisit $K=5$, $p=2$ et l'hétéroscédaticité pour nos données. On utilisera l'algorithme EM pour estimer nos paramètre.



La prédiction nous donne le graphique ci-dessus. Grâce à cela nous avons donc découpé le graphique en fonction du nombre de classe que nous disposions nous donnant le graphique suivant :



Enfin, nous avons estimé les différents segments de tel manière à ce que chaque segment aie une fonction polynomiale qui l'ajuste

