

# Modèle à processus latent et algorithme EM pour la régression non linéaire

Allou Samé, Faicel Chamroukhi, Gérard Govaert

► **To cite this version:**

Allou Samé, Faicel Chamroukhi, Gérard Govaert. Modèle à processus latent et algorithme EM pour la régression non linéaire. 41èmes Journées de Statistique, SFdS, Bordeaux, 2009, Bordeaux, France, France. 2009. <inria-00386702>

**HAL Id: inria-00386702**

**<https://hal.inria.fr/inria-00386702>**

Submitted on 22 May 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# MODÈLE À PROCESSUS LATENT ET ALGORITHME EM POUR LA RÉGRESSION NON LINÉAIRE

Allou Samé<sup>1</sup>, Faicel Chamroukhi<sup>1,2</sup>, Gérard Govaert<sup>2</sup>

<sup>1</sup>*Institut National de Recherche sur les Transports et leur Sécurité*

*Laboratoire des Technologies Nouvelles*

*2 rue de la Butte Verte, 93166 Noisy-le-Grand Cedex*

<sup>2</sup>*Université de Technologie de Compiègne*

*Laboratoire HEUDIASYC, UMR CNRS 6599*

*BP 20529, 60205 Compiègne Cedex*

**Résumé** Cet article propose une méthode de régression non linéaire qui s'appuie sur un modèle intégrant un processus latent qui permet d'activer préférentiellement, et de manière souple, des sous-modèles de régression polynomiaux. L'estimation des paramètres du modèle est effectuée par un algorithme EM dédié.

**Mots clés** : régression non linéaire, processus latent, modèle de mélange, algorithme EM

**Abstract** A non linear regression approach which consists of a specific regression model incorporating a latent process is introduced in this paper. The parameters estimation is performed by a dedicated EM algorithm.

**Keywords** : non linear regression, latent process, mixture model, EM algorithm

## 1 Introduction

La régression non linéaire est un problème central dans de nombreux domaines qui concernent la prédiction, le débruitage de signaux et leur paramétrisation. Son but est de caractériser au mieux la relation (non linéaire) existant entre une variable dépendante (grandeur physique), que nous supposons scalaire dans cet article, et une variable indépendante qui, comme c'est souvent le cas, est liée au temps. Cette relation non linéaire peut être due au fait que les données sont issues d'un modèle physique intrinsèquement non linéaire par rapport au temps, ou que le processus de génération des données comporte différents régimes linéaires (voire polynomiaux) qui se succèdent au cours du temps.

Plusieurs modèles ont déjà été proposés dans le cadre de l'apprentissage statistique pour résoudre ce type de problème. Parmi ces approches, on peut citer les modèles polynomiaux par morceaux [6][7], les méthodes à base de B-splines [2], le perceptron multi-couche dans sa version régressive [2], les méthodes à fonctions de base radiale [2]. La plupart de ces méthodes ramènent généralement le problème de régression non linéaire à des problèmes de régression linéaire simples à résoudre.

Dans ce travail, nous proposons une méthode alternative qui consiste à remplacer le modèle de régression non linéaire habituel par un modèle de régression linéaire intégrant un processus caché, qui permet d'activer préférentiellement un modèle de régression polynomial parmi  $K$  modèles. L'utilisation d'une fonction logistique comme loi conditionnelle des

variables latentes assure une souplesse de transition (lente ou rapide) entre les différents polynômes, ce qui permet d’obtenir une modélisation correcte de non linéarité. La loi conditionnelle de la variable dépendante, sous ce modèle, est connue dans la littérature sous l’appellation de mélange d’experts [5]. Une estimation des paramètres du modèle par l’algorithme EM est ainsi proposée. Dans la section 2 nous décrivons comment le modèle de régression à processus latent peut être exploité dans le cadre de la régression non linéaire. La section 3 présente la méthode d’estimation des paramètres via l’algorithme EM et la section 4 montre les performances de la méthode proposée sur des données simulées.

## 2 Régression non linéaire et modèle à processus latent

### 2.1 Cadre général

On suppose disposer d’un échantillon  $((x_1, t_1), \dots, (x_n, t_n))$ , où  $x_i$  désigne la variable aléatoire scalaire dépendante et  $t_i$  la variable temporelle indépendante. Le problème de régression non linéaire, sous sa formulation générique [1], consiste à estimer une fonction  $g$  de paramètre  $\boldsymbol{\theta}$ , en considérant le modèle M1 :  $x_i = g(t_i; \boldsymbol{\theta}) + \varepsilon_i$ , où les  $\varepsilon_i$  sont des bruits gaussiens centrés *i.i.d* de variance  $\sigma^2$ . L’estimation s’effectue généralement par la maximisation de la vraisemblance ou de manière équivalente par la minimisation du critère des moindres-carrés  $C_1(\boldsymbol{\theta}) = \sum_{i=1}^n (x_i - g(t_i; \boldsymbol{\theta}))^2$ . Dans le modèle M1, la fonction  $g(t; \boldsymbol{\theta})$  représente l’espérance de  $x$  conditionnellement à  $t$ . La minimisation du critère  $C_1$ , sous certaines conditions de régularité sur  $g$  [1], fournit un estimateur asymptotiquement gaussien, efficace et sans biais du paramètre  $\boldsymbol{\theta}$ . En pratique, on a souvent recours à des algorithmes itératifs d’optimisation qui convergent localement.

Pour couvrir un panel assez large de fonctions non linéaires de régression qui soient facilement paramétrables, nous optons pour des fonctions qui peuvent s’écrire sous la forme d’une somme finie de polynômes pondérés par des fonctions logistiques :

$$g(t; \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k(t; \boldsymbol{w}) \boldsymbol{\beta}_k^T \mathbf{t} \quad \text{avec} \quad \pi_k(t; \boldsymbol{w}) = \frac{\exp(w_{k0} + w_{k1}t)}{\sum_{\ell=1}^K \exp(w_{\ell 0} + w_{\ell 1}t)}, \quad (1)$$

où le vecteur  $\boldsymbol{\beta}_k = (\beta_{k0}, \dots, \beta_{kp})^T$  de  $\mathbb{R}^{p+1}$  désigne l’ensemble des coefficients d’un polynôme de degré  $p$  et  $\mathbf{t} = (1, t, t^2, \dots, t^p)^T$  son vecteur de monômes associés, le vecteur  $\boldsymbol{w} = (w_{10}, w_{11}, \dots, w_{K0}, w_{K1})$  de  $\mathbb{R}^{2K}$  désigne l’ensemble des paramètres de la fonction logistique et  $\boldsymbol{\theta} = (\boldsymbol{w}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K)$  représente l’ensemble des paramètres du modèle.

Ce modèle particulier de régression peut s’interpréter comme étant un modèle formé de différents sous-modèles de régression activés de manière souple par des fonctions logistiques qui contrôlent également les vitesses de transition entre polynômes.

La minimisation du critère  $C_1$ , pour ce choix particulier de la fonction  $g$ , n’admet pas de solution analytique simple. Il faut s’appuyer sur une procédure numérique d’optimisation du type Gauss-Newton, Newton-Raphson ou quasi-Newton qui converge localement.

## 2.2 Modèle de régression à processus latent proposé

Pour atteindre l'objectif d'estimation de la fonction de régression  $g$  du modèle M1, nous proposons d'utiliser une méthode alternative s'appuyant sur un modèle génératif intégrant un processus latent discret  $(z_1, \dots, z_n)$  avec  $z_i \in \{1, \dots, K\}$ . Ce modèle est défini par :

$$x_i = \sum_{k=1}^K z_{ik} \boldsymbol{\beta}_k^T \mathbf{t}_i + \varepsilon_i \quad ; \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \quad (2)$$

où  $z_{ik}$  vaut 1 si  $z_i = k$  et vaut 0 sinon, et où les  $\varepsilon_i$  sont supposés être indépendants. Les variables  $z_i$  du processus latent, conditionnellement aux instants  $t_i$ , sont supposées être générées indépendamment suivant la loi multinomiale  $\mathcal{M}(1, \pi_1(t_i; \mathbf{w}), \dots, \pi_K(t_i; \mathbf{w}))$ , où  $\pi_k(t_i; \mathbf{w})$  est défini par l'équation (1).

On peut alors vérifier que le modèle de régression proposé conduit à la même espérance conditionnelle  $E[x|t] = \sum_{k=1}^K \pi_k(t; \mathbf{w}) \boldsymbol{\beta}_k^T \mathbf{t} = g(t; \boldsymbol{\theta})$  que celle du modèle M1. Ainsi, grâce aux propriétés asymptotiques classiques de normalité, d'absence de biais et d'efficacité de l'estimateur du maximum de vraisemblance de  $\boldsymbol{\theta}$ , la fonction de régression  $g$  estimée à partir du modèle (2) est asymptotiquement identique à celle obtenue à partir du modèle M1. Ce qui nous conforte sur le fait que le modèle proposé peut être une bonne alternative pour résoudre le problème de régression non linéaire, si on dispose d'un algorithme adapté pour l'estimation de ses paramètres.

On peut montrer que conditionnellement à  $t$ , la variable aléatoire  $x$  est distribuée suivant le mélange de densités normales  $p(x_i|t_i; \boldsymbol{\Phi}) = \sum_{k=1}^K \pi_k(t_i; \mathbf{w}) \mathcal{N}(x_i; \boldsymbol{\beta}_k^T \mathbf{t}_i, \sigma^2)$  appelé aussi mélange d'experts [5], où  $\boldsymbol{\Phi} = (\boldsymbol{\theta}, \sigma^2)$ ,  $\mathcal{N}(\cdot; \mu, \sigma^2)$  désigne la fonction de densité d'une loi normale d'espérance  $\mu$  et de variance  $\sigma^2$ . La section suivante montre comment les paramètres du modèle peuvent être estimés par la méthode du maximum de vraisemblance.

## 3 Estimation des paramètres via l'algorithme EM

Dans cette section, compte tenu du fait que la densité de la loi conditionnelle de  $x$  s'écrit sous la forme d'un modèle de mélange, nous exploitons le cadre élégant de l'algorithme EM [3] pour estimer ses paramètres.

Les hypothèses d'indépendance des  $\varepsilon_i$  et d'indépendance des  $z_i$  conditionnellement aux  $t_i$  entraînent l'indépendance des  $x_i$  conditionnellement aux  $t_i$ . La log-vraisemblance à maximiser s'écrit donc  $L(\boldsymbol{\Phi}) = \sum_{i=1}^n \log p(x_i|t_i; \boldsymbol{\Phi}) = \sum_{i=1}^n \log \sum_{k=1}^K \pi_k(t_i; \mathbf{w}) \mathcal{N}(x_i; \boldsymbol{\beta}_k^T \mathbf{t}_i, \sigma^2)$ . Cette maximisation ne pouvant pas être effectuée analytiquement, nous nous appuyons sur l'algorithme EM [3] pour l'effectuer. L'algorithme EM, dans cette situation, itère à partir d'un paramètre initial  $\boldsymbol{\Phi}^{(0)}$  les deux étapes suivantes jusqu'à la convergence.

**Étape E (Espérance)** Cette étape consiste à calculer l'espérance de la log-vraisemblance complétée  $\log p(\mathbf{x}, \mathbf{z}; \boldsymbol{\Phi})$ , conditionnellement aux données observées et au paramètre

courant  $\Phi^{(q)}$  ( $q$  étant l'itération courante). Dans notre situation, on obtient  $Q(\Phi, \Phi^{(q)}) = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(q)} \log[\pi_k(t_i; \mathbf{w}) \mathcal{N}(x_i; \beta_k^T \mathbf{t}_i, \sigma^2)]$ , où  $\tau_{ik}^{(q)} = \frac{\pi_k(t_i; \mathbf{w}^{(q)}) \mathcal{N}(x_i; \beta_k^T \mathbf{t}_i, \sigma^2)}{\sum_{\ell=1}^K \pi_\ell(t_i; \mathbf{w}^{(q)}) \mathcal{N}(x_i; \beta_\ell^T \mathbf{t}_i, \sigma^2)}$  est la probabilité a posteriori que  $x_i$  soit issu de la  $k^e$  composante du mélange. Cette étape nécessite simplement le calcul des  $\tau_{ik}^{(q)}$ .

**Étape M (Maximisation)** Cette étape (de mise à jour) consiste à calculer le paramètre  $\Phi^{(q+1)}$  qui maximise  $Q(\Phi, \Phi^{(q)})$  par rapport à  $\Phi$ . Il suffit pour cela de maximiser séparément les quantités  $Q_1(\mathbf{w}) = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(q)} \log \pi_k(t_i; \mathbf{w})$  et  $Q_2(\beta_1, \dots, \beta_K, \sigma^2) = \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(q)} \log \mathcal{N}(x_i; \beta_k^T \mathbf{t}_i, \sigma^2)$ . On obtient les  $\beta_k^{(q+1)}$  en résolvant analytiquement  $K$  problèmes de moindres-carrés ordinaires pondérés par les  $\tau_{ik}^{(q)}$ . La variance  $\sigma^{2(q+1)}$ , qui est identique pour toutes les composantes du mélange, est donnée par :  $\sigma^{2(q+1)} = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(q)} (x_i - \beta_k^T \mathbf{t}_i)^2$ . La maximisation de  $Q_1$  par rapport à  $\mathbf{w}$  est un problème convexe de régression logistique multinomial pondéré par les  $\tau_{ik}^{(q)}$ . Nous utilisons l'algorithme IRLS (Iterative Reweighted Least Squares) [4] adapté à ce type de problème, pour calculer  $\mathbf{w}^{(q+1)}$ . Cette dernière procédure est initialisée avec le paramètre  $\mathbf{w}^{(q)}$ .

Les valeurs optimales du nombre de composantes  $K$  du modèle et de l'ordre  $p$  des polynômes de régression peuvent être obtenues en maximisant le critère d'information bayésien (BIC) [8] défini comme étant le critère de vraisemblance obtenu à la convergence de l'algorithme, pénalisé par un terme qui dépend du nombre de paramètres libres du modèle.

## 4 Expérimentation sur des données simulées

L'objet de cette partie est d'évaluer l'approche de régression proposée en utilisant des données simulées. Pour ce faire, nous la comparons avec une méthode de régression polynomiale par morceaux [6][7]. La qualité des estimations fournies par chacune des deux méthodes est l'écart quadratique moyen entre la courbe de régression estimée et la courbe de régression simulée :  $\frac{1}{n} \sum_{i=1}^n (g_{sim}(t_i) - g_{est}(t_i))^2$ . Chaque jeu de données est généré en ajoutant un bruit gaussien centré à des points d'une courbe non linéaire, régulièrement échantillonnés sur l'intervalle temporel  $[0; 5]$ .

### 4.1 Paramètres de simulation et réglage des algorithmes

Trois fonctions non linéaires de régression non linéaires ont été considérées :

- la fonction non linéaire  $g_1(t, \theta) = \sum_{k=1}^4 \pi_k(t; \mathbf{w}) \beta_k^T \mathbf{t}$  avec  $\beta_1 = [34, -60, 30]$ ,  $\mathbf{w}_1 = [547, -154]$ ,  $\beta_2 = [-17, 29, -7]$ ,  $\mathbf{w}_2 = [526, -135]$ ,  $\beta_3 = [185, -104, 15]$ ,  $\mathbf{w}_3 = [464, -115]$ ,  $\beta_4 = [-804, 343, -35]$ ,  $\mathbf{w}_4 = [0, 0]$ ;
- la fonction polynomiale par morceaux  $g_2(t, \theta) = \beta_1^T \mathbf{t} \mathbf{I}_{[0;2.5]}(t) + \beta_2^T \mathbf{t} \mathbf{I}_{]2.5;5]}(t)$  avec  $\beta_1 = [33, -20, 4]$  et  $\beta_2 = [-78, 47, -5]$ ;
- la fonction non linéaire  $g_3(t) = 20 \sin(1.6\pi t) \exp(-0.7t)$ .

Les deux algorithmes testés ont été lancés avec  $(K = 4, p = 2)$  pour la situation 1,  $(K = 2, p = 2)$  pour la situation 2 et avec  $(K = 5, p = 3)$  pour la situation 3. Pour chaque jeu de données, l’algorithme EM a été lancé à partir de 10 initialisations aléatoires différentes et seule la solution ayant la plus grande vraisemblance a été retenue.

## 4.2 Résultats

La figure 1 montre pour chacune des situations, comment varie l’écart entre les courbes de régression simulées et les courbes estimées, en fonction de la taille d’échantillon et de la variance du bruit. Pour chaque taille d’échantillon et chaque valeur de la variance du bruit, l’erreur quadratique présentée correspond à une moyenne sur 20 jeux de données différents.

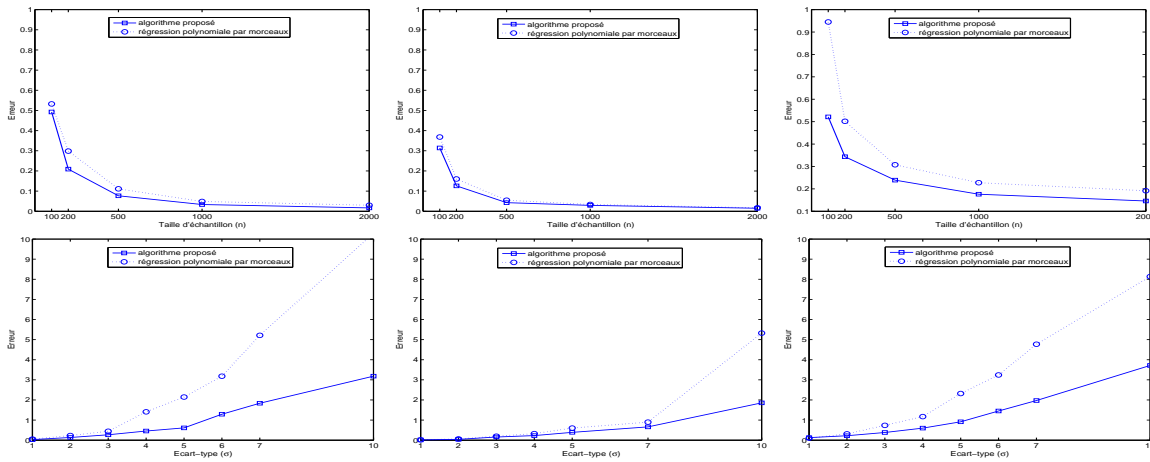


FIG. 1 – Erreur entre la courbe estimée et la courbe réelle simulée, en fonction de la taille d’échantillon (haut) et en fonction de la variance du bruit (bas), pour la situation 1 (gauche), la situation 2 (milieu) et la situation 3 (droite)

Ces graphiques montrent que la méthode proposée donne de meilleurs résultats que la méthode de régression polynomiale par morceaux. En outre, on peut observer que l’écart entre les courbes estimées et les courbes simulées décroît quand la taille d’échantillon augmente. L’augmentation de la variance du bruit entraîne quant à elle une augmentation de l’erreur qui est plus prononcée pour le modèle de régression polynomiale par morceaux. La figure 2 montre pour chacune des situations, un exemple de courbe de régression estimée avec l’algorithme proposé.

## 5 Conclusion

Une méthode de régression non linéaire a été proposée dans cet article. Celle-ci s’appuie sur un modèle de régression simple intégrant un processus latent qui permet d’activer

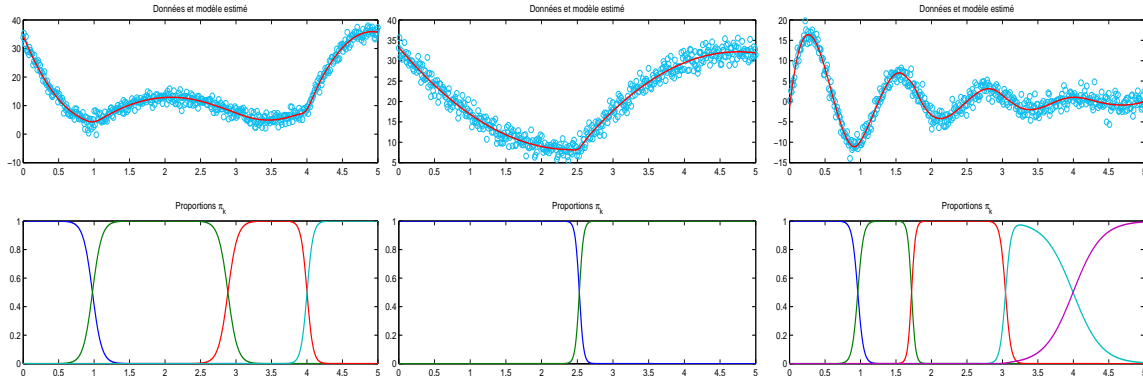


FIG. 2 – Fonctions de régression  $g(t; \theta)$  estimées par l’algorithme proposé et proportions  $\pi_k$  correspondantes

successivement, et manière souple, des sous-modèles de régression polynomiaux. Les premiers résultats obtenus sur des données simulées sont très encourageants et confortent l’intérêt de notre démarche. Les perspectives de ce travail seront de comparer l’algorithme EM proposé avec un autre algorithme itératif minimisant directement le critère des moindres-carrés, en termes de vitesse de convergence et de qualité des estimations.

## Bibliographie

- [1] A. Antoniadis, J. Berruyer et René Carmona (1992) *Régression non linéaire et applications*, Economica.
- [2] C. M. Bishop (2006) *Pattern recognition and machine learning*, Information Science and Statistics, Springer.
- [3] A. P. Dempster, N. M. Laird et D. B. Rubin (1977) *Maximum likelihood from incomplete data via the EM algorithm*, JRSS, B, 39(1) : 1–38.
- [4] P. Green (1984) *Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some robust and resistant alternatives*, JRSS, B, 46(2) : 149–192.
- [5] M. I. Jordan et R. A. Jacobs (1994) *Hierarchical Mixtures of Experts and the EM algorithm*, Neural Computation, 6 : 181–214.
- [6] V. E. McGee et W. T. Carleton (1970) *Piecewise regression*, Journal of the American Statistical Association, 65, 1109–1124.
- [7] A. Samé, P. Aknin et G. Govaert (2007) *Classification automatique pour la segmentation des signaux unidimensionnels*, Rencontres de la SFC, ENST, Paris.
- [8] G. Schwarz (1978) *Estimating the dimension of a model*, Annals of Statistics, 6 : 461–464.