



HAL
open science

Variables latentes dans les modèles linaires généralisés

Djeneba Beldohore Thiam, Gregory Nuel

► **To cite this version:**

Djeneba Beldohore Thiam, Gregory Nuel. Variables latentes dans les modèles linaires généralisés. 1ères Rencontres R, Jul 2012, Bordeaux, France. hal-00717533

HAL Id: hal-00717533

<https://hal.science/hal-00717533>

Submitted on 13 Jul 2012

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.

Variables latentes dans les modèles linaires généralisés

D. Thiam^a et G. Nuel^{a,b}

^aLabo de Maths Appliquées (MAP5, CNRS 8145)
Université Paris Descartes
djenaba.thiam@gmail.com

^b Institut des Maths et Interactions (INSMI)
CNRS Paris
gregory.nuel@parisdescartes.fr

Mots clefs : Variables Latentes, modèles linaires généralisés, algorithme EM.

En sciences sociale comme en biologie, de nombreux phénomènes d'intérêt ne sont pas observés directement et sont modélisés par des variables latentes [1]. Un exemple est celui de données non observées en raison d'un seuil de détection au niveau de l'appareil de mesure [2]. Une autre illustration est celle des données hétérogènes qui peuvent être traitées en introduisant une classe latente [3].

Parmi les packages disponibles sous R qui permettent la gestion des variables latentes dans les modèles linaires généralisés (GLMs) on peut citer `flexmix` [4] dans le cadre de modèles de régression et `lcm` [5] dans le cadre de modèles à effets mixtes. Si ces packages sont puissants et indéniablement utiles, ils ont le défaut notable de limiter les possibilités de modélisation aux cas implémentés par les développeurs. Dans le cas de `flexmix` par exemple, on ne peut considérer qu'un simple modèle de mélange (avec variables concomitantes pour la classe). Que faire si cette classe latente intervient de manière hiérarchique? Comment gérer des paramètres partagés à travers différentes classes? Que faire si un GLM particulier n'est pas implémenté (ex: régression multinomiale)? Et comment peut-on introduire des variables latentes continues (autres que des effets mixtes)?

L'objectif de ce travail est de répondre à ces questions en présentant une approche simple et généraliste qui permet de gérer sous R tout type de variables latentes dans les modèles linaires généralisés sans recourir pour cela à une implémentation spécifique. Notre approche consiste à utiliser un classique algorithme Expectation-Maximization (EM) [6] sans avoir à entrer au cœur des méthodes d'estimations. La clé est une utilisation astucieuse de l'option `weights` des procédures d'estimation classiques (ex: `lm`, `glm`, `lmer` et `glmer`), ces poids étant mis à jour impérativement à l'extérieur de la procédure d'estimation.

Pour illustrer cette approche, considérons le modèle suivant:

$$\mathbf{y} \sim \mathbf{x} + \mathbf{z}$$

où le \mathbf{y} est une variable de réponse ($\in \mathbb{R}^n$), \mathbf{x} est une covariable ($\in \mathbb{R}^n$), et \mathbf{z} une variable latente binaire ($\in \{0, 1\}^n$). Si \mathbf{z} était connu, un simple `fit = lm(y ~ x + z)` permettrait d'ajuster ce modèle. La valeur de \mathbf{z} étant manquante, on se tourne vers l'algorithme EM. Supposons qu'à une itération donnée de l'algorithme on dispose d'un paramètre θ (contient les paramètres du modèle linéaire ainsi que la proportion *a priori* ρ de $\mathbf{z}[i] = 1$), il nous suffit alors de remplacer

le paramètre courant par:

$$M(\theta) = \arg \max_{\theta'} \underbrace{\sum_z \mathbb{P}(z|\mathbf{x}, \mathbf{z}; \theta) \log \mathbb{P}(y|\mathbf{x}, \mathbf{z}; \theta')}_{Q(\theta'|\theta)}$$

Or cette étape est en fait équivalente à l'ajustement du modèle:

$$\begin{pmatrix} y \\ y \end{pmatrix} \sim \begin{pmatrix} \mathbf{x} \\ \mathbf{x} \end{pmatrix} + \begin{pmatrix} z = 1 \\ z = 0 \end{pmatrix} \quad \text{avec} \quad \text{weights} = \begin{pmatrix} \mathbf{w} \\ 1 - \mathbf{w} \end{pmatrix}$$

où $\mathbf{w} = \mathbb{P}(z = 1|\mathbf{x}, \mathbf{z}; \theta)$. On peut ainsi facilement mettre à jour les paramètres de la régression à l'aide de la commande: `fit = lm(c(y, y) ~ c(x, x) + c(z = 1, z = 0), weights = c(w, 1 - w))`. Le paramètre ρ de la variable latente peut quant à lui facilement être mis à jour directement à partir de \mathbf{w} : $\rho = \text{mean}(\mathbf{w})$.

Dans le cas d'une variable latente discrète, l'approche proposée est totalement équivalente à un algorithme EM classique (y compris en termes de complexité). Pour les variable latentes continues, notre approche se ramène à une approximation (on se contente de répliquer \mathbf{z} pour un nombre donné de valeurs qui sont spécifiques à chaque individus et à chaque itération). La méthode est évidemment généralisable aux modèles linéaires généralisés plus complexes, seul le calcul et la mise à jours des poids restant à la charge de l'utilisateur.

Avec la méthodologie proposée, nous montrons qu'une exploitation astucieuse de l'option `weights` d'une procédure de R permet d'introduire de manière très souple des variables latentes dans cette procédure (ici l'ajustement de modèles linéaires généralisés). Au delà de cet exemple particulier, l'approche que nous suggérons devrait inciter le développeur de toute procédure statistique sous R à se poser la question de la prise en compte d'observations pondérées par des poids avec la perspective d'une future exploitation de cette fonctionnalités dans le cadre de l'algorithme EM et de ses variantes.

Références

- [1] Kenneth A. Bollen (2002). Latent variables in psychology and social sciences. *Annual Review of Psychology* , **53**, 605-634
- [2] Goodman L.A(1974). The analysis of systems of qualitative variables when some of the variables are unobservable *JAmerican Journal of Sociology* , **79**, 1179-1259
- [3] Bert F. Green (1951). A general solution for the latent class model of latent structure analysis *PSYCHOMETRIKA* , **16**, 151-166
- [4] Friedrich Leisch (2003). FlexMix: A general framework for finite mixture models and latent class regression in R. *Report* , **86**
- [5] Cecile Proust-Lima, Benoit Liqueur (2009). lcmm: an R package for estimation of latent class mixed models and joint latent class models, *R cran*.
- [6] A. P. Dempster; N. M. Laird; D. B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm *Journal of the Royal Statistical Society* , **39**, 1-38