ELSEVIER

Computational Statistics & Data Analysis 39 (2002) 137–152

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Bayesian analysis of the Logit model and comparison of two Metropolis–Hastings strategies

Anas Altaleb[a], Didier Chauveau[b,*]

[a] *Department of Mathematics and Statistics, Faculty of Engineering, University of Damascus, Damascus, Syria*

[b] *Equipe d'Analyse et Mathématiques Appliquées, Université de Marne-la-Vallée, 5 Bd. Descartes, Champs sur Marne, 77454 Marne la Vallee, Cedex 2, France*

Received 1 November 2000; received in revised form 1 June 2001

Abstract

We examine some Markov chain Monte Carlo (MCMC) methods for a generalized non-linear regression model, the Logit model. It is first shown that MCMC algorithms may be used since the posterior is proper under the choice of non-informative priors. Then two non-standard MCMC methods are compared: a Metropolis–Hastings algorithm with a bivariate normal proposal resulting from an approximation, and a Metropolis–Hastings algorithm with an adaptive proposal. The results presented here are illustrated by simulations, and show the good behavior of both methods, and superior performances of the method with an adaptive proposal in terms of convergence to the stationary distribution and exploration of the posterior distribution surface.

Keywords: Bayesian statistic; Markov chain Monte Carlo; Metropolis–Hastings algorithm; Adaptive algorithm; Stationarity; Convergence assessment

1. Introduction

From a Bayesian point of view, there is no fundamental difference between the observation and the parameter of a statistical model, both considered as a random variable. Thus, if we note D the data and θ the parameter of the model considered plus the latent data, then a formal inference requires the update of the joint distribution

* Corresponding author.
E-mail address: chauveau@math.univ-mlv.fr (D. Chauveau).

0167-9473/02/$ - see front matter
PII: S 0167-9473(01)00055-X

$f(D,\theta)$ on all the variables. Therefore, the determination of $\pi(\theta)$ and $f(D|\theta)$ gives $f(D,\theta)=f(D|\theta)\pi(\theta)$. Having observed D, we can use the Bayes's Theorem to determine the distribution of θ conditionally to the data (i.e. the posterior law)

$$\pi(\theta|D)=\frac{f(D|\theta)\pi(\theta)}{\int f(D|\theta)\pi(\theta)\,\mathrm{d}\theta}. \tag{1}$$

For Bayesians, the characteristics of the posterior law are significant for the inference, and quantities of interest (e.g., moments) can be expressed in terms of conditional expectation of a function of θ with respect to the posterior law, $E[h(\theta)|D]=\int h(\theta)\pi(\theta|D)\,\mathrm{d}\theta$. Moreover, it is also rare to have a posterior law $\pi(\theta|D)$ which is explicit (more exactly, which can be simulated directly) and it is necessary to be able to simulate a sample $(\theta_1,\ldots,\theta_n)$ which is approximately i.i.d. from $\pi(\theta|D)$, in order to determine the confidence regions or the general structure of the law (detection of modes, etc.).

A Markov chain Monte Carlo (MCMC) algorithm generates an ergodic Markov chain $\theta^{(t)}$ with the posterior $\pi(\theta|D)$ as the stationary distribution. An MCMC method then uses the fact that, for t large enough, $\theta^{(t)}$ is approximately $\pi(\cdot|D)$ distributed, and $E[h(\theta)|D]$ can be approximated by ergodic averages from the chain. The most commonly used MCMC methods are the Metropolis–Hastings (M–H) algorithm (Hastings, 1970), and the Gibbs sampler (first introduced by Geman and Geman, 1984).

In the next section, we present a generalized non-linear regression model: the Logit model. Then, we detail (in Lemma 1) the necessary conditions for the use of MCMC algorithms and we expose two methods that can be used for this model. We present a random walk Metropolis–Hastings algorithm with a bivariate normal approximation as the proposal, and we compare it with a recent MCMC method introduced by Chauveau and Vandekerkhove (1999, 2001): a Metropolis–Hastings algorithm with an adaptive proposal. In the last section, we apply some diagnostic methods to control convergence of these MCMC algorithms, computed with the "convergence diagnosis and output analysis software for Gibbs sampling output" (CODA, see Best et al., 1995). Practically we conclude that, for this model, the M–H algorithm with an adaptive proposal is more effective, in terms of convergence speed towards the stationary distribution and in terms of surface exploration speed of the posterior law, than the M–H algorithm with a normal approximation for the proposal. However, the adaptive method requires a larger computing time, since it needs some kind of exploratory stage before delivering its effective algorithm.

Note that these MCMC methods could be applied for the Probit, or other related models. However, the analog of Lemma 1 for these models needs to be worked out specifically.

2. The Logit model

A standard qualitative regression model is the Logit model, where the distribution of y conditionally to the explanatory variables $X\in\mathbb{R}^p$ is, for $\gamma\in\mathbb{R}^p$:

$$P(y=1)=1-P(y=0)=\frac{\exp(X^t\gamma)}{1+\exp(X^t\gamma)}.$$

We consider the logistic dependence between the explanatory variables and the observation. Consider the particular case where $X=(1,x)$ and $\gamma=(\alpha,\beta)$. The binary variables y_i's in $\{0,1\}$ are associated with the explanatory variables x_i's and modelled following a Bernoulli law of conditional probability:

$$y_i|x_i \sim B\left(\frac{\exp(\alpha+\beta x_i)}{1+\exp(\alpha+\beta x_i)}\right). \tag{2}$$

Suppose that our parameters follow an improper prior $\pi(\alpha,\beta)=1$. The likelihood of our model, for a sample $(y_1,x_1),\ldots,(y_n,x_n)$ is equal to

$$f(y_1,\ldots,y_n|x_1,\ldots,x_n,\alpha,\beta)=\prod_{i=1}^{n}\frac{\exp[(\alpha+\beta x_i)y_i]}{1+\exp(\alpha+\beta x_i)}.$$

The posterior law of (α,β) results then by a formal application of the Bayes's theorem:

$$\begin{aligned}\pi(\alpha,\beta\,|\,D) &\propto f(y_1,\ldots,y_n|x_1,\ldots,x_n,\alpha,\beta)\pi(\alpha,\beta)\\ &\propto \prod_{i=1}^{n}\frac{\exp[(\alpha+\beta x_i)y_i]}{[1+\exp(\alpha+\beta x_i)]}=\frac{\exp[\sum_{i=1}^{n}(\alpha+\beta x_i)y_i]}{\prod_{i=1}^{n}[1+\exp(\alpha+\beta x_i)]}.\end{aligned} \tag{3}$$

3. Condition to use MCMC algorithms

The use of non-informative prior laws, i.e. a σ-finite measure of infinite mass on the parameters' space, implies that the posterior law derivation using proportionality relation, as $\pi(\theta|x) \propto f(x|\theta)\pi(\theta)$, is not necessarily acceptable to implement an M–H algorithm on $f(x|\theta)\pi(\theta)$, because the corresponding law may not exist, i.e. $f(x|\theta)\pi(\theta)$ is not necessarily integrable. We meet the same difficulty for the Gibbs sampling which, contrary to the M–H algorithm, uses conditional laws extracted from $\pi(\theta_1,\ldots,\theta_q)$ which itself is represented by the above proportionality relation. It may happen that these laws are clearly definite and simulable, but do not correspond to a joint law π, i.e. π is not integrable (see Robert 1996a for examples). This fact, rather frequent in a generalized Bayesian approach does not represent a defect for the MCMC algorithms, nor even a simulation problem. We should not however omit the π-existence verification, that we prove in the lemma below. The following hypothesis is introduced.

Hypothesis [H]

Given a sample $(x_1,y_1),\ldots,(x_n,y_n)$ with $n\geqslant 4$, we suppose there exists positive x_i and negative x_i associated to both of $y_i=1$ and 0.

Lemma 1. *The posterior distribution of the Logit model* $\pi(\alpha,\beta\,|\,D)$, *where D represents the observed data, is a true law under* [H], *i.e.*

$$\int\int \pi(\alpha,\beta\,|\,D)\,\mathrm{d}\alpha\,\mathrm{d}\beta < +\infty.$$

Proof. Define

$$I = \int\int \prod_{i=1}^{n} \frac{\exp[(\alpha+\beta x_i)y_i]}{[1+\exp(\alpha+\beta x_i)]}\,\mathrm{d}\alpha\,\mathrm{d}\beta.$$

Given the i index set $\{i_1,\ldots,i_p\}$ for which $y_i=1$, then the integral is written as

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{\exp(\alpha p)\exp(\beta x_0)}{\prod_{i=1}^{n}[1+\exp(\alpha+\beta x_i)]}\,\mathrm{d}\alpha\,\mathrm{d}\beta = I_1+I_2+I_3+I_4,$$

with $x_0=\sum_{i=1}^{n} x_i y_i$, and

$$I_1 = \int_{-\infty}^{0}\int_{-\infty}^{0} \frac{\exp(\alpha p)\exp(\beta x_0)}{\prod_{i=1}^{n}(1+\exp(\alpha+\beta x_i))}\,\mathrm{d}\alpha\,\mathrm{d}\beta,$$

$$I_2 = \int_{-\infty}^{0}\int_{0}^{+\infty} \frac{\exp(\alpha p)\exp(\beta x_0)}{\prod_{i=1}^{n}(1+\exp(\alpha+\beta x_i))}\,\mathrm{d}\alpha\,\mathrm{d}\beta,$$

$$I_3 = \int_{0}^{+\infty}\int_{-\infty}^{0} \frac{\exp(\alpha p)\exp(\beta x_0)}{\prod_{i=1}^{n}(1+\exp(\alpha+\beta x_i))}\,\mathrm{d}\alpha\,\mathrm{d}\beta,$$

$$I_4 = \int_{0}^{+\infty}\int_{0}^{+\infty} \frac{\exp(\alpha p)\exp(\beta x_0)}{\prod_{i=1}^{n}(1+\exp(\alpha+\beta x_i))}\,\mathrm{d}\alpha\,\mathrm{d}\beta.$$

The denominator is equal to

$$\begin{aligned}
&\prod_{i=1}^{n}(1+\exp(\alpha+\beta x_i))\\
&\quad=\exp(n\alpha)\exp\left(\beta\sum_{i=1}^{n}x_i\right)+\cdots+\exp((p+1)\alpha)\left[\sum_{\sigma\in W_{p+1}}\exp\left(\beta\sum_{j=1}^{p+1}x_{\sigma(j)}\right)\right]\\
&\qquad+\exp(p\alpha)\left[\sum_{\sigma\in W_p}\exp\left(\beta\sum_{j=1}^{p}x_{\sigma(j)}\right)\right]\\
&\qquad+\exp((p-1)\alpha)\left[\sum_{\sigma\in W_{p-1}}\exp\left(\beta\sum_{j=1}^{p-1}x_{\sigma(j)}\right)\right]+\cdots+1,
\end{aligned}$$

where $W_q=\{\text{all the injections } \sigma:\{1,\ldots,q\}\to\{1,\ldots,n\}\}$. For the first and third integrals, we keep only in the denominator the terms which have $\exp(\alpha(p-1))$ as common factor, hence for I_1 we have

$$\begin{aligned}
I_1 &\leqslant \int_{-\infty}^{0}\int_{-\infty}^{0} \frac{\exp(\alpha p)\exp(\beta x_0)}{\exp(\alpha(p-1))[\sum_{\sigma\in W_{p-1}}\exp(\beta\sum_{j=1}^{p-1}x_{\sigma(j)})]}\,\mathrm{d}\alpha\,\mathrm{d}\beta\\
&= \int_{-\infty}^{0} \frac{\exp(\beta x_0)}{[\sum_{\sigma\in W_{p-1}}\exp(\beta\sum_{j=1}^{p-1}x_{\sigma(j)})]}\,\mathrm{d}\beta.
\end{aligned}$$

However, $\sum_{i=1}^{n} x_i y_i = \sum_{k=1}^{p} x_{i_k}$, where $(i_1, i_2, \ldots, i_p)$ are the indices of y_{i_k} for which $y_{i_k} = 1$. By [H], there exists $\tilde{k} \in (i_1, i_2, \ldots, i_p)$ such as $y_{\tilde{k}} = 1$ and $x_{\tilde{k}} > 0$. Given, without loss of generality, i_p such as $y_{i_p} = 1$ and $x_{i_p} > 0$, and $\tilde{\sigma}$ such as: $\tilde{\sigma}(k) = i_k$ with $k = 1, 2, \ldots, p-1$, we have $\tilde{\sigma} \in W_{p-1}$, therefore we can bound the integral by

$$I_1 \leqslant \int_{-\infty}^{0} \frac{\exp(\beta \sum_{k=1}^{p} x_{i_k})}{[\exp(\beta \sum_{k=1}^{p-1} x_{\tilde{\sigma}(k)})]} \, \mathrm{d}\beta = \int_{-\infty}^{0} \frac{\exp(\beta \sum_{k=1}^{p} x_{i_k})}{[\exp(\beta \sum_{k=1}^{p-1} x_{i_k})]} \, \mathrm{d}\beta$$

$$= \int_{-\infty}^{0} \exp(\beta x_{i_p}) \, \mathrm{d}\beta < \infty.$$

For I_2 and I_4, we keep the terms which have common factor $\exp(\alpha(p+1))$ and the proof goes in the same way. □

4. The Metropolis–Hastings algorithm

The generic M–H algorithm is based on the use of a conditional *proposal density* $q(y|x)$ with respect to the dominant measure for the model. It can be put in practice only if q can be quickly simulated and is, either analytically available up to a constant independent of x, or symmetrical, i.e. such as $q(y|x) = q(x|y)$. The M–H algorithm associated with the target π produces a Markov chain $(x^{(t)})$ based on the following transition: Given $x^{(t)}$,

1. `Generate` $y_t \sim q(y|x^{(t)})$,
2. `Take` $x^{(t+1)} = \begin{cases} y_t & \texttt{with probability } \rho(x^{(t)}, y_t) \\ x^{(t)} & \texttt{with probability } 1 - \rho(x^{(t)}, y_t) \end{cases}$ [A1]

`where`

$$\rho(x^{(t)}, y_t) = \min\left\{ \frac{\pi(y_t)}{\pi(x^{(t)})} \frac{q(x^{(t)}|y_t)}{q(y_t|x^{(t)})}, 1 \right\}.$$

This algorithm systematically accepts simulations y_t such that the ratio $\pi(y_t)/q(y_t|x^{(t)})$ is higher than the preceding value $\pi(x^{(t)})/q(x^{(t)}|y_t)$. It is only in the symmetrical case that the acceptance is controlled by the ratio $\pi(y_t)/\pi(x^{(t)})$.

4.1. Random walk Metropolis–Hastings algorithm

A particular case of the M–H algorithm is the random walk algorithm, for which $q(y|x) = g(|y-x|)$. For example, when x is continuous, $q(\cdot|x)$ can be a multivariate normal distribution of mean x and constant variance–covariance matrix Σ. A careful choice of the proposal distribution results in generating a small step $y - x^{(t)}$ which generally gives a high rate of acceptance and a slowly mixing chain. A bad choice of the proposal distribution leads to an excessive step which provides movements of the center to the tail of the distribution, and in general produces small values of $\pi(y)/\pi(x^{(t)})$ and a small rate of acceptance. Such a chain also leads to a slow mixing. The ideal solution to avoid these two cases is to use a scale parameter for the proposal law. The algorithm [A1] authorizing this dependence, $q(y|x)$ can thus

be of the form $g_\tau(|y-x|)$, i.e. $y_t = x^{(t)} + \tau\varepsilon_t$, ε_t being a random perturbation of distribution g, independent of $x^{(t)}$, and τ being a scale parameter. The Markov chain associated to q_τ is then a random walk. Note that the choice of g as a symmetrical function $g(-t) = g(t)$ gives $\rho(x^{(t)}, y_t) = \min\{1, \pi(y_t)/\pi(x^{(t)})\}$ in [A1].

Compared to other algorithm, the random walk M–H algorithm requires a specific rate of acceptance analysis, because of the dependence of the proposal law on the value previously accepted. A high rate of acceptance does not state that the algorithm evolves correctly. Conversely, if the average rate of acceptance is weak, the successive values of $\pi(y_t)$ are frequently small compared to $\pi(x^{(t)})$, i.e. the random walk moves quickly on the surface of π (but can visit too much the tails of π). An automatic parameterization method cannot guarantee optimal performances for the random walk M–H algorithm, and the choices of rates operated here do not inevitably lead to optimality. Gelman et al. (1996), recommend rates of acceptance close to 50% for one or two-dimensional models.

4.2. Independent Metropolis–Hastings algorithm

The particular situation where $q(y|x) = q(y)$ gives the independent M–H algorithm usually called the *independence sampler*. Conditions for its geometric convergence have been studied in recent literature. In particular, Mengersen and Tweedie (1996) proved geometric convergence in the total variation norm of the independence sampler when the proposal satisfies $q(y|x) = q(y) \geqslant a\pi(y)$ for some $a \in (0,1)$, a property which has been stated more precisely by Holden (1998), who proves (under some assumptions for π) geometric convergence with rate $(1-a)^t$ in the relative supremum norm. This functional result highlights the strong link between the convergence rate and the proximity of q to the target π, and is an essential ingredient of the adaptive M–H (Chauveau and Vandekerkhove, 1999, 2001) briefly presented in Section 5.2.

5. Two Metropolis–Hastings strategies

5.1. A normal-based approximation for the proposal density

Let $\hat{\theta}$ be the maximum likelihood estimator (MLE) of θ computed from the data y. We have the following asymptotic distribution:

$$\sqrt{n}(\theta - \hat{\theta}) \approx \mathcal{N}(0, C), \tag{4}$$

where $C = I^{-1}(\theta)$ is the Fisher information for an observation (in the i.i.d. case),

$$I(\theta|y) = E\left[-\frac{\partial^2 I(\theta|y)}{\partial\theta\partial\theta^t}\right].$$

For Bayesians, $\hat{\theta}$ is fixed conditionally on the data y and θ is the variable. Knowing the model and the data, (4) implies that the posterior density of θ is asymptotically normal of average $\hat{\theta}$, and of variance–covariance matrix C. The Bayesian justification

of (4) comes from the Taylor expansion of the loglikelihood of the posterior density about the fixed value $\hat{\theta}$:

$$l(\theta|y) = l(\hat{\theta}|y) + (\theta - \hat{\theta})^t S(\hat{\theta}|y) - \tfrac{1}{2}(\theta - \hat{\theta}) I(\hat{\theta}|y)(\theta - \hat{\theta})^t + r(\theta|y),$$

with

$$I(\hat{\theta}|y) = - \left. \frac{\partial^2 l(\theta|y)}{\partial\theta\partial\theta^t} \right|_{\hat{\theta}}.$$

Note that $E[I(\theta|y)]$ is the Fisher information matrix. Since $S(\hat{\theta}|y) = 0$, by assuming that $r(\theta|y)$ can be neglected, the posterior density is proportional to the multivariate normal density of average $\hat{\theta}$ and of variance–covariance matrix $C = I^{-1}(\hat{\theta}|y)$ (see Tanner and Wong, 1987).

5.1.1. Application to the Logit model: bivariate normal approximation

We reconsider the Logit model in order to approach the posterior law of (α, β) by implementing the random walk M–H algorithm with a bivariate normal proposal:

$$\begin{pmatrix} \alpha^{(t+1)} \\ \beta^{(t+1)} \end{pmatrix} = \mathcal{N}_2 \left(\begin{pmatrix} \alpha^{(t)} \\ \beta^{(t)} \end{pmatrix}, \Sigma^{(t)} \right).$$

To calculate $\Sigma^{(t)}$, we take the Taylor expansion of the logarithm of the target law about the value $(\hat{\alpha}, \hat{\beta})$, which is given by

$$\log \pi(\alpha, \beta) = \log \pi(\hat{\alpha}, \hat{\beta}) + \tfrac{1}{2}(\alpha - \hat{\alpha}, \beta - \hat{\beta}) \nabla\nabla^t \log \pi(\hat{\alpha}, \hat{\beta})(\alpha - \hat{\alpha}, \beta - \hat{\beta})^t.$$

Then we replace $E[\nabla\nabla^{(t)} \log \pi(\alpha, \beta)]$ by its observation. That implies the calculation of

$$\nabla \log \pi(\alpha, \beta) = \begin{pmatrix} \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \frac{\exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))} \\ \sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} \frac{x_i \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))} \end{pmatrix},$$

$$\nabla\nabla^t \log \pi(\alpha, \beta) = - \begin{pmatrix} \sum_{i=1}^{n} \frac{\exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} & \sum_{i=1}^{n} \frac{x_i \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} \\ \sum_{i=1}^{n} \frac{x_i \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} & \sum_{i=1}^{n} \frac{x_i^2 \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} \end{pmatrix}.$$

Moreover, we consider the matrix $\Sigma^{(t)}$ adjusted using a scale factor τ:

$$\Sigma^{(t)} = \tau^2 \begin{pmatrix} \sum_{i=1}^{n} \frac{\exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} & \sum_{i=1}^{n} \frac{x_i \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} \\ \sum_{i=1}^{n} \frac{x_i \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} & \sum_{i=1}^{n} \frac{x_i^2 \exp(\alpha+\beta x_i)}{(1+\exp(\alpha+\beta x_i))^2} \end{pmatrix}^{-1}.$$

The subjacent idea is to calibrate the factor τ according to the rate of acceptance of the algorithm (see Robert, 1996b). Lastly by substituting (α, β) with (α^t, β^t), maximum likelihood, we obtain $\Sigma^{(t)}$. Since the proposal law in this case is the bivariate normal $\mathcal{N}(0, \Sigma)$, then the algorithm application can respectively pass by the generation of the marginal and the conditional laws:

$$X_1 \sim \mathcal{N}(0, \sigma_1^2), \qquad X_2|X_1 \sim \mathcal{N}\left(\rho\frac{\sigma_2}{\sigma_1}x_1, \sigma_2^2(1-\rho^2)\right).$$

5.2. A Metropolis–Hastings algorithm with an adaptive proposal

This method is described in full details and studied theoretically in Chauveau and Vandekerkhove (2001). It has been successfully tested on simulations for recovering synthetic multimodal target densities consisting of mixtures of univariate or bivariate Gaussian distributions. However, this application for the Logit model is the first use of this method in actual models for doing Bayesian inference. Since the first strategy developed in Section 5.1 is in some sense an "ad hoc" method because of its data-driven approximation for the proposal density, we found interesting to compare it with this adaptive, but "blind" strategy.

We just give here the flavour of this method for brevity. Let p^t be the density of the M–H algorithm at time t, and assume that an arbitrary proposal q_0 such that $q_0(y) \geqslant a_0\pi(y)$ for all y is available. This proposal insures geometric convergence with rate $(1 - a_0)^t$ in the relative supremum norm $||(p^t - \pi)/\pi||_\infty$ (see Holden, 1998). A natural way to improve the convergence rate is to find a minoration constant greater than a_0, i.e. to use a proposal which approximates π in a better way than q_0 does. Since p^t converges geometrically to π, it provides some information that may be used at selected instants to build proposal densities approximating π. The authors define a M–H algorithm using a sequence of proposals q_t based on histogram density estimates of p^t constructed from i.i.d. copies of the algorithm, and suitably modified to fulfill the minoration condition $q_t \geqslant a_t\pi$. This results in an inhomogeneous version of the independent M–H algorithm. The theoretical convergence study requires appropriate technical conditions regarding the target density π, the number of i.i.d. chains required at the updating times, and the way these parallel chains are used (the chains used at time t have to be discarded to preserve independence and Markov property). It is shown that a single chain issued from this strategy has almost surely a rate of convergence better than $(1 - a_0)^t$.

This asymptotic result has to be moderated when actual application is concerned. The support of π is generally $\mathbb{R}^d$, but the histogram proposals can only be constructed on compact sets that need to be defined. Also, the algorithm can run only for a finite number of iterations, with a finite number of proposal updating. Indeed, the drawback of this method is clearly its implementation cost and its computing time since it uses parallel chains for proposal updating. The authors suggest to start with a dispersed proposal q_0 to obtain a good exploration of the support of π. A suitable compact can be quickly found by short trial-and-error runs, or some prior information concerning the target. Then, they suggest to apply an exploration scheme using some sets of

parallel chains and a small number of proposal mutations, in such a way to end up with a single chain from an homogeneous M–H algorithm using a good proposal (the last update) resulting from this adaptive stage. In this way, the computing time is reduced. Also, the implementation cost is actually not a real drawback since an interesting feature of the adaptive algorithm is that it is *generic*, in the sense that it only needs some tuning parameters (e.g., the definition of the adaptive scheme in terms of number of parallel chains and mutation times, and the definition of the compact). After that, it builds its sequence of proposal in an automated way. Hence a "black-box" type computer program has been written, into which the user only have to plug the definition of its target density.[1] This code implements the exploratory stage with the parallel simulation and the histograms construction in a multi-dimensional setting. It has been used in this paper to run the adaptive M–H algorithm for the Logit model. The only specific implementation task was to write the definition of the target density (3) for the model into this black-box routine. Note that the trial-and-error procedure needed to find an appropriate region can be done easily since the program also delivers the information that some chains have "escaped" from the selected compact if it is too small.

6. Comparison of the algorithms

The comparison between the generic M–H algorithm and the M–H algorithm with an adaptive proposal a priori seems to give the advantage to the first method, since this method draws its proposal distributions from an asymtotic approximation of π, while the adaptive version is founded on an arbitrary initial proposal which will converge sequentially to π. Indeed, the first method is, by construction, more direct than the adaptive one because it avoids parallel simulations.

The defects of the M–H algorithm more often come from a bad adjustment between the target π and the proposal q than from a too strong approximation between the two laws. However, the M–H methods allows to remedy these defects by increasing certain variation parameters. The essential disadvantage of the M–H algorithm is rather to not always seize the details of the distribution π because of a not very accurate simulation scale. Sometimes this considerably increases the computing time and reduces the convergence rate. In contrast, the adaptive M–H algorithm uses many parallel jumps started from a dispersed initial proposal, to quickly detect the locations of interest for π (this is a known advantage of any parallel chain method). The exploration stage allows hopefully for good updates of the initial proposal, these updates reflecting the complexity of π (modes, etc.). In this sense, it is a "mode hunting method".

6.1. How to compare the two methods?

We need to precise how we ran and compared both algorithms, since there is a difficulty here coming from the fact that the random walk with bivariate normal

[1] This computer C program is available upon request to the second author.

approximation is a "single chain" method, and the adaptive version is a "parallel chain" method in essence. In addition, the adaptive version requires a reasonably large number of jumps for its exploratory stage to perform well. We chose to allow 50,000 iterations for this stage, even if we could have build a good proposal in less iterations (note that it took only few minutes on a standard desktop computer, and that we also tried 10,000 iterations with good results). We thus decided to run a burn-in of the same number of iterations for the two methods, and then to compare both methods on the basis of one single chain of length 10,000 issued from each algorithm. This is "fair" from the duration of the resulting chains point of view, but may be unfair if we consider how the methods use the burn-in iterations.

The random walk M–H essentially uses these burn-in iterations to escape from the starting position (i.e. to obtain values that do not depend on the starting point), and (presumably) to obtain a chain approximately π distributed (this is what burn-in is usually for). On the other way, the adaptive M–H uses these iterations to *improve its kernel*, by building proposals that approximate π on the basis of what it has already discovered. One could claim that the random walk M–H does not require a burn-in length of 50,000 iterations, and that some of these burn-in iterations are useless for it and thus favoured the adaptive method, which is true in some sense. In another way, if we would want to correct the comparison from this "bias", then we would have to compare a single chain running for the total number of iterations including its burn-in, against the adaptive chain including its exploration stage. Unfortunately, we cannot compare using classical convergence diagnostic software (like CODA), a single chain against the adaptive version which uses specific parallel simulation schemes (like non-rectangular arrays of chains, as explained in Chauveau and Vandekerkhove, 2001).

This is actually the reason why we have chosen to compare two single chains issued from each methods, keeping in mind that the two methods do not take the same benefit of their burn-in stage. In short, what we show in the next section is that after its exploration stage, the adaptive version produces a really better chain, as shown by the simulation and convergence diagnosis.

7. Convergence diagnosis

Under relatively general conditions, the Markov chains produced by the MCMC methods are ergodic. These conditions are necessary, but however insufficient for the implementation, since they suffer from a significant problem in the applicability: how to determine the moment when we can conclude their convergence, in other words, when one should stop the chain and use the observations in order to estimate the distribution characteristics considering that the sample is sufficiently representative of the stationary distribution. There exists some so-called *convergence assessment* methods which give practical results to handle this problem (see Cowles and Carlin, 1996; Brooks and Roberts, 1998). Note that the available diagnosis try to check necessary conditions which are not sufficient to ensure the convergence.

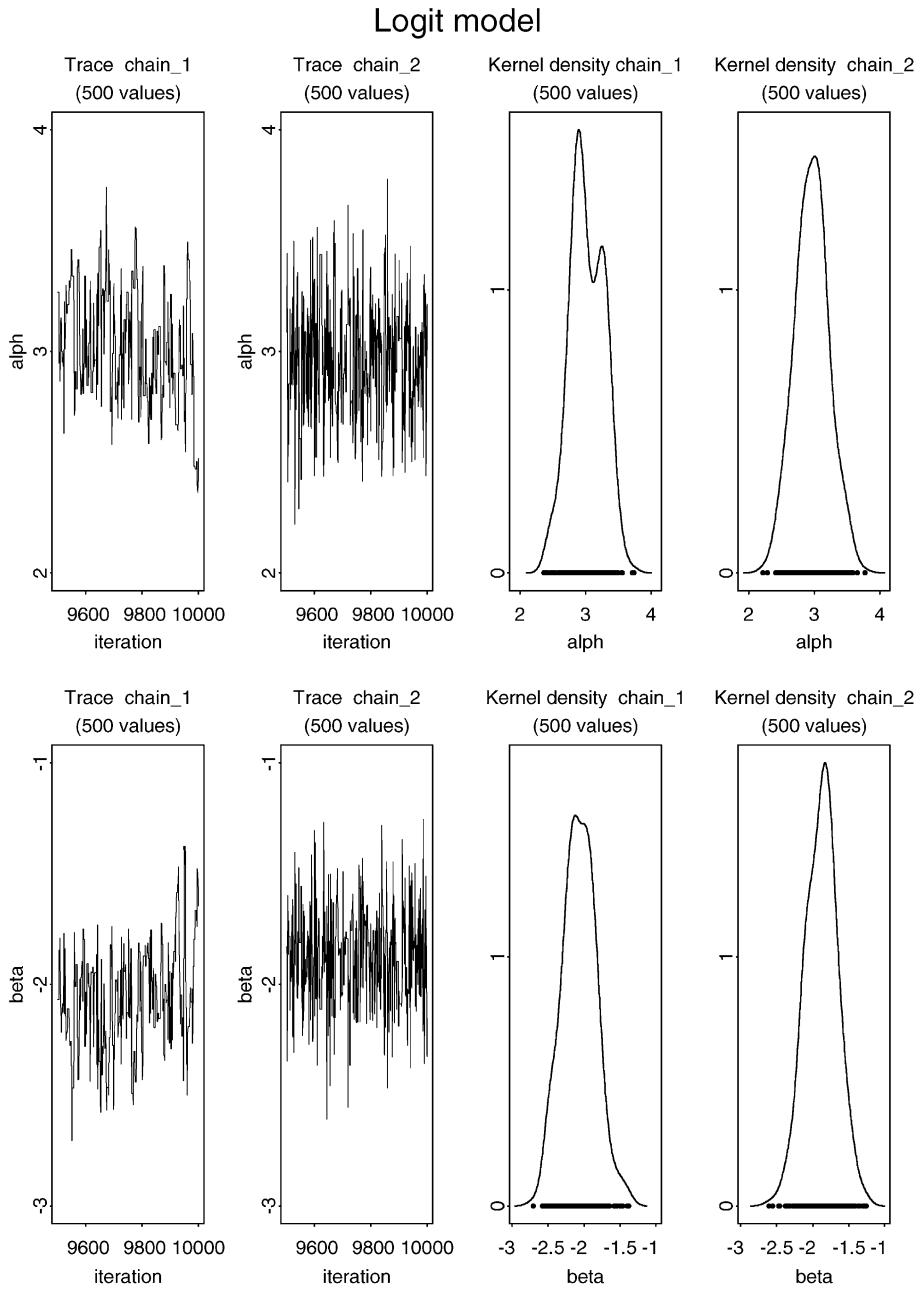

Fig. 1. Trace and density comparison for α and β coming from the M–H algorithm with a bivariate normal approximation ($chain_1$), and an adaptive M–H ($chain_2$). The illustrations are for the 500 last iterations of the chains.

According to Cowles and Carlin (1996) and Brooks and Roberts (1998), we can distinguish three convergence degrees for which a control is necessary. The first one decides if the variables $\theta^{(t)}$ are approximately π-distributed. The second essentially

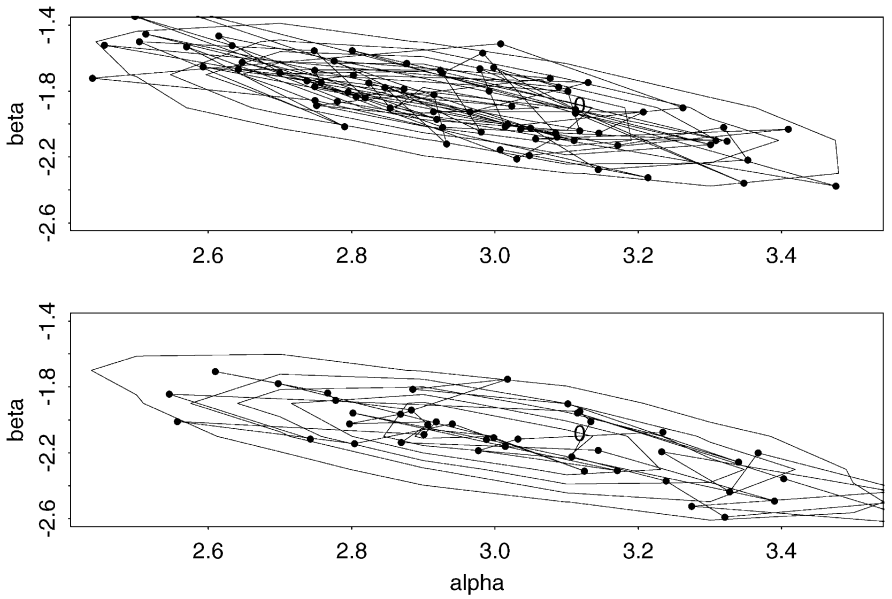

Fig. 2. Displacement comparison of the chain $(\alpha^{(t)}, \beta^{(t)})$ on the contour of the posterior law for the 100 last iterations. The chains come from the adaptive M–H (*top*), and the M–H algorithm with a bivariate normal (*bottom*).

determines the minimal value of T authorizing the approximation of $E_\pi[h(\theta)]$ by the classical Monte Carlo estimator $\sum_{t=1}^{T} h(\theta^{(t)})/T$. It is also related to the precision of this approximation by means of a Central Limit Theorem, which is related to the mixing of the chain. The methods of parallel chains and "batch sampling" are of the third convergence type, to guarantee the independence or quasi-independence of the simulated variables (see Mengersen et al., 1999).

7.1. Application to the Logit model

For the Logit model, we have simulated the explanatory variable $x_i \sim \mathcal{N}(0,1)$ and we observed this model for $\alpha = 3$, $\beta = -2$ with the data size $n = 500$. As said before, the illustrations are based on 10,000 iterations for each method, and are obtained by the CODA software (see Best et al., 1995).

In Fig. 1, the trace of each chain shows that the adaptive method moves more rapidly, essentially due to its independent proposal density. The density comparison also shows clearly that the density is smoother for the adaptive method over 500 iterations, essentially since this method results in less dependence between successive observations. This is in accordance with Fig. 2, which shows intuitively that the adaptive M–H is more effective in term of surface exploration of the posterior distribution (much more moves are observed for the same duration). For more precise diagnosis and comparisons, Fig. 3 gives the autocorrelations for the parameters α, β of the Logit model, for both methods. We see that the first method provides a

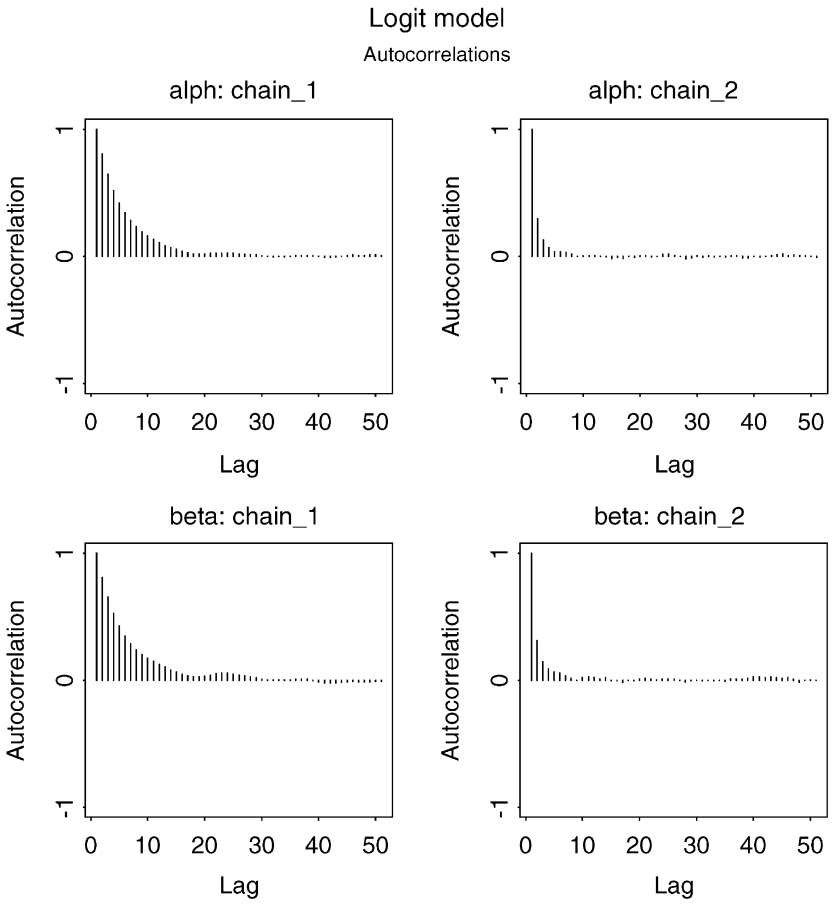

Fig. 3. Autocorrelation comparison for the parameters α and β coming from the M–H with a bivariate normal ($chain_1$), and the adaptive M–H ($chain_2$).

high autocorrelation in this example. This indicates a very weak mixing and induces a slow convergence for this algorithm, characterized in Fig. 3 by a slow topdown oscillation. On the contrary, the M–H algorithm with an adaptive proposal induces weak autocorrelations but approximately of the same order, i.e. a quick convergence speed, indicated in Fig. 3 by a rapid topdown oscillation. It seems obvious that the very strong autocorrelation for the M–H algorithm method considerably limits the displacement of the chain on the contour of the posterior law, and this undoubtedly explains the need for a large iteration number to achieve convergence.

From the Geweke (1992) convergence control tool, we observe in Fig. 4 that much of the values of its statistics Z_n for the M–H algorithm are out of the interval ± 1.96 of the reduced centered normal distribution $\mathcal{N}(0,1)$, meaning the failure of the convergence test (they must be inside this interval in the case of convergence). In contrast, for the M–H algorithm with an adaptive proposal, the majority of the values are in the interval ± 1.96, which supposes a strong possibility of convergence.

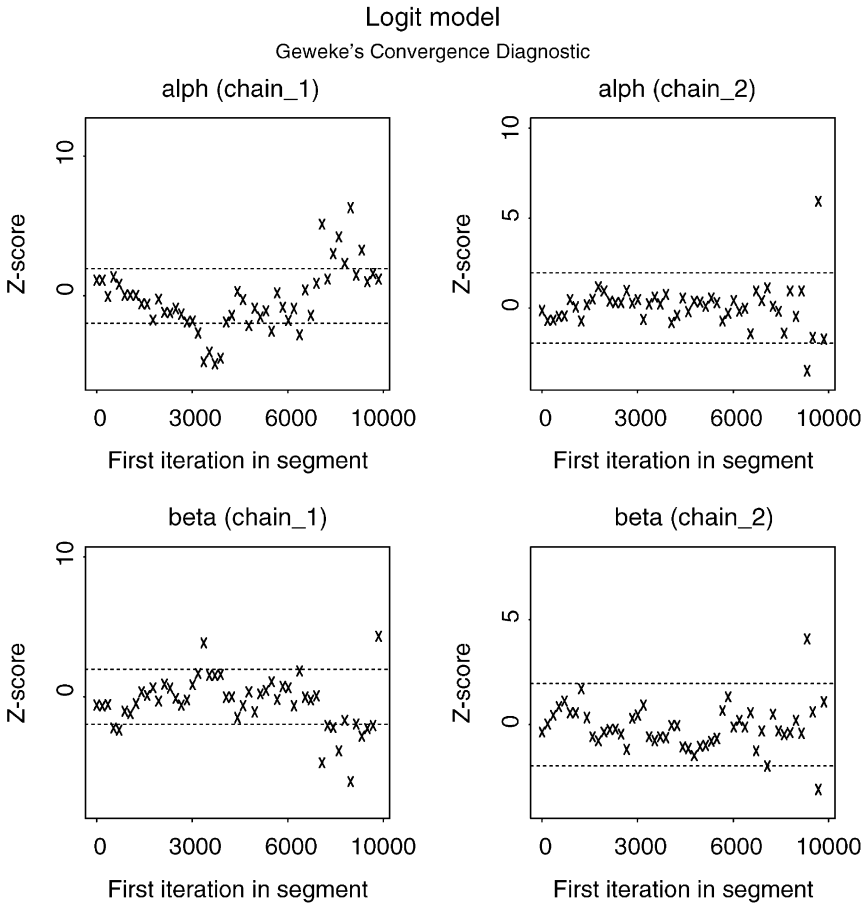

Fig. 4. Geweke convergence diagnosis comparison for the Logit model according to the M–H with a bivariate normal ($chain_1$), and the adaptive M–H ($chain_2$).

In addition to the two preceding methods of diagnosis (autocorrelation and Geweke convergence control), which encourage the rejection of the random walk M–H method, the effectiveness of the adaptive M–H algorithm is confirmed by the diagnostic of Raftery and Lewis (1992a, b). Table 1 gives the evolution of k, minimum sampling step, t_0, number of minimum iterations necessary to obtain the stationarity and T, total number of iterations ensuring convergence. Indeed, Table 1 indicates that for the M–H algorithm, with a test chain of 10,000 iterations, we need 15,094 (23,910) as the total number of iterations, in which 14 (22) initial iterations have to be rejected, and a step batch size of 1 (2), for the parameters α (β), respectively. While the adaptive method requires a step of 1 (3) with a rejection proportion of 35.7% (54.5%) compared to that of the M–H algorithm and requires a total number of iterations about 36.3% (50.6%) of that of the M–H algorithm with an identical minimum numbers of iterations respectively for the parameter α and β.

Table 1
Table of Raftery and Lewis diagnosis for this experiment, for the M–H with a bivariate normal approximation (chain 1), and the adaptive M–H (chain 2)

Chain	Variable	T in (k)	Burn-in (t_0)	Total (T)	Lower bound ($Tmin$)
Chain-1	α	1	14	15,094	3746
	β	2	22	23,910	3746
Chain-2	α	1	5	5482	3746
	β	3	12	12,114	3746

Acknowledgements

The authors wish to express their gratitude to the Referee for his insightful comments that help improving the paper.

References

Best, N.G., Cowles, M.K., Vines, K., 1995. CODA: convergence diagnosis and output analysis software for Gibbs sampling output, Version 0.30. Technical Report, MRC Biostatistics Unit, University of Cambridge.

Brooks, S.P., Roberts, G., 1998. Assessing convergence of Markov chain Monte Carlo algorithms. Statist. Comput. 8, 319–335.

Chauveau, D., Vandekerkhove, P., 1999. Un Algorithme de Hastings–Metropolis avec apprentissage séquentiel. C. R. Acad. Sci. Paris, Série I t 329, 173–176.

Chauveau, D., Vandekerkhove, P., 2001. Improving convergence of the Hastings–Metropolis algorithm with an adaptive proposal, Scand. J. Statist., to appear.

Cowles, M.K., Carlin, B.P., 1996. Markov Chain Monte-Carlo convergence diagnostics: a comparative study. J. Amer. Statist. Assoc. 91, 883–904.

Gelman, A., Gilks, W.R., Roberts, G.O., 1996. Efficient Metropolis jumping rules. In: Berger, J.O., Bernardo, J.M., Dawid, A.P., Lindley, D.V., Smith, A.F.M. (Eds.), Bayesian Statistics, Vol. 5. Oxford University Press, Oxford, pp. 599–608.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. 6, 721–741.

Geweke, J., 1992. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments (with discussion). In: Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (Eds.), Bayesian Statistics, Vol. 4. Oxford University Press, Oxford, pp. 169–193.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their application. Biometrika 57, 97–109.

Holden, L., 1998. Geometric convergence of the Metropolis–Hastings simulation algorithm. Statist. Probab. Lett. 39 (4), 371–377.

Mengersen, K.L., Robert, C.P., Guihenneuc-Jouyaux, C., 1999. MCMC convergence diagnostics: a "reviewww". In: Berger, J.O., Bernardo, J.M., Dawid, A.P., Lindley, D.V., Smith, A.F.M. (Eds.), Bayesian Statistics, Vol. 6. Oxford University Press, Oxford, pp. 415–441.

Mengersen, K.L., Tweedie, R.L., 1996. Rates of convergence of the Hastings and Metropolis algorithms. Ann. Statist. 24, 101–121.

Raftery, A.E., Lewis, S., 1992a. How many iterations in the Gibbs sampler?. In: Berger, J.O., Bernardo, J.M., Dawid, A.P., Smith, A.F.M. (Eds.), Bayesian Statistics, Vol. 4. Oxford University Press, Oxford, pp. 763–773.

Raftery, A.E., Lewis, S., 1992b. The number of iterations, convergence diagnostics and generic Metropolis algorithms. Technical Report, Department of Statistics, University of Washington, Seattle.
Robert, C.P., 1996a. Méthodes de Monte Carlo par Chaînes de Markov. Economica, Paris.
Robert, C.P., 1996b. Inference in mixture models. In: Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (Eds.), Markov Chain Monte Carlo in Practice. Chapman & Hall, London, pp. 441–464.
Tanner, M., Wong, W., 1987. The calculation of posterior distributions by data augmentation. J. Amer. Statist. Assoc. 82, 528–550.