Stat Comput (2013) 23:123–134
DOI 10.1007/s11222-011-9296-2

On adaptive Metropolis–Hastings methods

Jim E. Griffin · Stephen G. Walker

Received: 13 August 2010 / Accepted: 18 October 2011 / Published online: 19 November 2011

Abstract This paper presents a method for adaptation in Metropolis–Hastings algorithms. A product of a proposal density and K copies of the target density is used to define a joint density which is sampled by a Gibbs sampler including a Metropolis step. This provides a framework for adaptation since the current value of all K copies of the target distribution can be used in the proposal distribution. The methodology is justified by standard Gibbs sampling theory and generalizes several previously proposed algorithms. It is particularly suited to Metropolis-within-Gibbs updating and we discuss the application of our methods in this context. The method is illustrated with both a Metropolis–Hastings independence sampler and a Metropolis-with-Gibbs independence sampler. Comparisons are made with standard adaptive Metropolis–Hastings methods.

Keywords MCMC · Metropolis–Hastings algorithm · Gibbs sampling · Metropolis-within-Gibbs · Adaptive Monte Carlo

1 Introduction

This paper is concerned with the sampling of a density function, which will be denoted by $f(x)$. The dimension of x is not relevant but is assumed to be finite. A particular choice of method to sample $f(x)$ is the Metropolis–Hastings algorithm, which is based on the construction of a Markov chain $(x^{(k)})$ with stationary density $f(x)$, and proceeds as follows: if the current state at iteration k is $x^{(k)}$, then a proposal x^* is sampled from some density $p(x^*|x^{(k)})$ and $x^{(k+1)}$ is taken to be x^* with probability

$$\alpha = \min\left\{1, \frac{f(x^*)\, p(x^{(k)}|x^*)}{f(x^{(k)})\, p(x^*|x^{(k)})}\right\}$$

or $x^{(k+1)} = x^{(k)}$ with probability $1-\alpha$. Full details are to be found in Smith and Roberts (1993) and Tierney (1994). A chain of N iterations from this algorithm can be used to estimate $\mathrm{E}[g(X)]$ with respect to $f(x)$ using the ergodic average $\frac{1}{N}\sum_{i=1}^{N} g(x^{(i)})$.

For a suitably chosen proposal $p(\cdot|x)$, the method is guaranteed to converge to the correct distribution; that is, as the chain proceeds the distribution of $x^{(k)}$ becomes closer to $f(x)$. However, the rate at which convergence happens is determined by the choice of the proposal density. Metropolis–Hastings independence samplers, i.e. simply a $p(x)$, can be extremely efficient if we can find a proposal which approximates $f(x)$ well, but it can be difficult to choose an approximation for complex distributions. One possible solution is the use of "pilot" runs to gather information about the distribution $f(x)$ which will be used to determine $p(x)$. A more elegant approach uses a proposal $p_j(x)$ at iteration j which is determined by $x^{(1)}, x^{(2)}, \ldots, x^{(j-1)}$ and so changes with the chain. Unfortunately, this destroys the Markov property of the chain and convergence to the correct distribution depends upon a correct choice of updating scheme for $p_j(x)$.

The idea of allowing proposals to depend on previous values of the chain is usually called adaptive Monte Carlo, and is developed from Haario et al. (2001), who provide

J.E. Griffin (✉) · S.G. Walker
School of Mathematics, Statistics & Actuarial Science, University of Kent, Canterbury, UK
e-mail: J.E.Griffin-28@kent.ac.uk

S.G. Walker
e-mail: S.G.Walker@kent.ac.uk

a scheme that is guaranteed to converge to the correct distribution. This approach has become a popular line of research and current ideas are reviewed by Andrieu and Thoms (2008). Also, a study of the ergodicity of the adaptive Metropolis algorithm of Haario et al. (2001) is given in Saksman and Vihola (2010). A general result on convergence is given by Roberts and Rosenthal (2007) and results for adaptive Metropolis–Hastings independence samplers are given by Andrieu and Moulines (2006). Adaptive methods using Metropolis–Hastings random walk samplers are discussed by, for example, Roberts and Rosenthal (2009), and Metropolis–Hastings independence samplers by Gasemyr (2003), who uses a normal proposal, and Giordani and Kohn (2008), who use mixtures of normals.

This paper considers an alternative type of adaptation using a joint distribution over K independent copies of $f(x)$. A Gibbs sampler with Metropolis-within-Gibbs steps is used to sample from this joint distribution. The idea of using multiple copies has appeared frequently (see e.g. Cappé et al. 2004, Gilks et al. 1994 and Mengersen and Robert 2003). If the chain is in equilibrium, the current values of the K copies will be drawn from the target distribution and can be used to define a proposal adapted to the target. Our method uses the K copies to calculate an approximation to the target which can be used as the proposal. This approximation approach was initially suggested by Warnes (2001) with the Normal Kernel Coupler, that uses a kernel density estimate of the K copies as a proposal. He proves that his algorithm converges to the target distribution. More recently, Cai et al. (2008) update the ith copy using an approximation based on the other $K-1$ copies as a proposal. This avoids the need to derive a specific proof of convergence as in Warnes (2001).

Our approach uses a pseudo-distribution that combines the proposal distribution with K copies of $f(x)$ and this provides a formal justification for a proposal which depends on past states of the chain without the development of fundamentally new theory. This allows the Normal Kernel Coupler to be justified using standard results and the sampler of Cai et al. (2008) to be extended to include the ith copy in the proposal when that copy is updated. The idea is closely related to the Bayesian Adaptive Independence Sampler (BAIS) proposed by Keith et al. (2008). In their algorithm, a model with parameters θ is fitted to the copies $x_1, x_2, \ldots, x_K$ with prior $p(\theta)$. The posterior predictive distribution of x_{K+1} is then used as a proposal. In contrast, we suggest using a proposal that arises from any estimation procedure (Bayesian or non-Bayesian) applied to the copies $x_1, x_2, \ldots, x_K$.

The number of extra computations will depend on the exact choice of proposal distribution. Simple approximations, such as the normal distribution, will involve negligible extra computations since sufficient statistics can typically be stored. To illustrate this, in a standard Metropolis random walk algorithm we would potentially replace the old value of x with the new value of x. In our algorithm, we would also update the sufficient statistics $\sum_{k=1}^{K} x_k$ and $\sum_{k=1}^{K} x_k^2$. A negligible difference in the number of operations. On the other hand, more computationally expensive approximations such as a kernel density estimate may be needed in more challenging examples (such as a bi-modal example that we explore later).

In practice, Metropolis–Hastings algorithms are often not directly applied to multidimensional target distributions and a Gibbs sampler is preferred since it allows one to take advantage of any conditional independence structure in $f(x)$. This is the case even if the full conditional distributions of some parameters do not have standard form. In this situation, it is common to use Metropolis–Hastings steps to update those parameters, which is often referred to as Metropolis-within-Gibbs. The target distribution is now $f(x_j|x_{-j})$, where x_{-j} refers to x without the jth element, which depends on the values of the other elements of the chain. Finding an appropriate proposal can be difficult and "pilot" runs can be hard to use. If a suitable approximation cannot be found, a Metropolis–Hastings random walk can be used but this will generally mix more slowly than an independence sampler. This problem is well-suited to adaptive Monte Carlo ideas but there has been little work developing such methods. Haario et al. (2005) update each dimension of a multivariate distribution using a Metropolis–Hastings random walk sampler where the variance of the steps at iteration j is chosen to be $(2.44)^2$ times the sample variance of the previous samples for that dimension. Roberts and Rosenthal (2009) use a random walk in each dimension with diminishing adaptation. Latuszynski et al. (2011) extend this idea to also allow adaptation of the probability of updating each dimension in a random-scan Gibbs sampler. Our framework can be simply extended to Metropolis-within-Gibbs sampling and allows the development of effective adaptive strategies.

We need to mention at this point the paper by Laskey and Myers (2003). Our multiple copies idea to be described in Sect. 2.1 is essentially the population MCMC algorithm defined in *Definition 2* of their paper. However, as these authors readily acknowledge, there is no theoretical analysis of the sampler undertaken in the paper and hence no theory for convergence. So, one view of our paper is that we are providing a new framework for the population MCMC of Laskey and Myers (2003), in which it is possible to confirm convergence in the traditional way.

The paper is organised as follows: Sect. 2 describes the new framework and general algorithms; Sect. 3 illustrates the use of these methods on some examples, discusses issues of convergence and mixing of the chain and compares the method to previously proposed adaptive Metropolis–Hastings algorithms. Finally, Sect. 4 contains a discussion.

2 The framework

Let us consider the joint density based on a target density $f(x)$ and a conditional density $p(\cdot|\cdot)$;

$$f(z,x) = p(z|x)\, f(x).$$

Clearly the marginal density of x is $f(x)$. A sample from $f(x)$ can be generated by first sampling from the joint distribution $f(z,x)$ and then retaining only the x samples. This idea underlies all auxiliary variable methods (see e.g. Besag and Green 1993). We can sample from the joint distribution $f(z,x)$ by constructing a sampler using the following steps at each iteration:

1. Sample z from $p(z|x)$.
2. Propose the deterministic move $x \to z$ and $z \to x$ using a Metropolis–Hastings step. This move is accepted with probability

$$\alpha = \min\left\{1, \frac{f(x,z)}{f(z,x)}\right\} = \min\left\{1, \frac{f(z)\, p(x|z)}{f(x)\, p(z|x)}\right\}.$$

It is easy to observe that this is the acceptance probability in a Metropolis–Hastings algorithm for sampling $f(x)$ with the proposal $p(x^\star|x)$ if $z = x^\star$. So using this Metropolis-within-Gibbs sampler and retaining only the x samples is identical to a Metropolis–Hastings algorithm. However, we now have a method where the proposal density sits in the joint density and which can be easily generalized to allow more complicated proposals.

2.1 Multiple copies idea

Let us consider introducing an extra copy of X into our joint distribution to define the more elaborate density function for (z, x_1, x_2):

$$f(z,x_1,x_2) = p(z|x_1,x_2)\, f(x_1)\, f(x_2).$$

The marginal distributions of x_1 and x_2 are $f(x_1)$ and $f(x_2)$. A chain from this joint distribution can again be Gibbs sampled using the following steps at each iteration:

1. Sample z from $p(z|x_1,x_2)$.
2. Propose the switch of z with x_1 with probability $1/2$; otherwise propose the switch z with x_2. If we propose to switch z with x_1, the acceptance probability for this move is

$$\alpha = \min\left\{1, \frac{f(x_1,z,x_2)}{f(z,x_1,x_2)}\right\} = \min\left\{1, \frac{p(x_1|z,x_2)\, f(z)}{p(z|x_1,x_2)\, f(x_1)}\right\}.$$

Similarly, if we propose to switch z with x_2, the acceptance probability will be

$$\alpha = \min\left\{1, \frac{f(x_2,x_1,z)}{f(z,x_1,x_2)}\right\} = \min\left\{1, \frac{p(x_2|x_1,z)\, f(z)}{p(z|x_1,x_2)\, f(x_2)}\right\}.$$

The output at each iteration is now the values of x_1 and x_2 which are both draws from the target distribution if the chain is in equilibrium. The expectation of a function $g(x)$ with respect to $f(x)$ can be approximated, using a sample of N values, by

$$\hat{E}_N[g(X)] = \frac{1}{2N}\sum_{j=1}^{N}\sum_{i=1}^{2} g\left(x_i^{(j)}\right)$$

where $x_1^{(j)}$ and $x_2^{(j)}$ represent the values of x_1 and x_2 at the jth iteration. This is the ergodic average over both copies of x. The chain only updates one copy at each iteration and so the Monte Carlo error decreases at a rate $1/\sqrt{N}$ (it is straightforward to show that $\mathrm{Var}(\hat{E}_N[(g(X)]) = \mathrm{Var}(g(X))/N$ if $p(z|x_1,x_2) = f(z)$).

Importantly, the proposal $p(\cdot|\cdot)$ can depend on the current values of the two copies: the values of x_1 and x_2. Unlike Cai et al. (2008) we do not directly apply the Metropolis–Hastings algorithm to $f(x_1,x_2) = f(x_1)f(x_2)$ which allows us to condition on both variables. Once the chain is in equilibrium, x_1 and x_2 are both draws from the target distribution, which provides more information about the target distribution than the single value x in a standard Metropolis–Hastings algorithm. However, two values do not provide enough information to accurately approximate the target distribution. An accurate approximation would be possible if we consider more copies of the target distribution.

A "proposal" depending on the current value of an arbitrary number of copies of the target distribution can be constructed by considering the joint density

$$f(z,x_1,\ldots,x_K) = p(z|x_1,\ldots,x_K)\prod_{i=1}^{K} f(x_i).$$

This idea is similar to the approach that Keith et al. (2008) use to derive their Bayesian Adaptive Independence Sampler. However, they restrict their attention to deriving $p(z|x_1,\ldots,x_K)$ from a Bayesian model. Our sampling algorithm updates the parameters taking the current set $(x_1,\ldots,x_K)$ and applying the following steps:

1. Sample z from $p(z|x_1,x_2,\ldots,x_K)$.
2. Propose the switch z with x_i where i is chosen uniformly from $(1,2,\ldots,K)$. The move is accepted with probabil-

ity

$$\alpha = \min\left\{1, \frac{f(z)\, q_i(x_i)}{f(x_i)\, q_z(z)}\right\},$$

where $q_i(x_i) = p(x_i|x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_K)$ and $q_z(z) = p(z|x_1, x_2, \ldots, x_K)$.

3. Record $(x_1, \ldots, x_K)$ as output.

In this case, the expectation of $g(X)$ can be approximated by

$$\hat{E}_N[g(X)] = \frac{1}{NK} \sum_{j=1}^{N} \sum_{i=1}^{K} g\left(x_i^{(j)}\right)$$

and again the Monte Carlo error decreases at a rate $1/\sqrt{N}$. The method is different to fully adaptive methodology since proposals are only based on K samples, and we will see in Sect. 3 that we do not need a large value of K to achieve good mixing performance.

The proposals can be chosen in any way. Once the chain is in equilibrium, the values of $x_1, x_2, \ldots, x_K$ are all draws from the target distribution. These provide an increasing amount of information about the target distribution as K becomes larger. A generic approach that we have found useful treats $x_1, x_2, \ldots, x_K$ as a random sample. Standard statistical methods can be used to estimate $f(x)$ and this estimate used as the proposal $p(\cdot|\cdot)$. For example, we could construct a normal approximation to $f(x)$ by estimating the mean and variance from $x_1, x_2, \ldots, x_K$, by taking their sample mean and variance, or we can use nonparametric techniques such as kernel density estimation (which defines the "Normal Kernel Coupler" of Warnes (2001) if a normal kernel is used). Better mixing rates will be achieved if we can closely approximate $f(x)$. We have found that simple methods, such as normal approximations, can be effective in practice.

The method involves two likelihood evaluations per iteration like a standard Metropolis–Hastings sampler (which can be reduced to one evaluation if the likelihood value for all current copies are stored). Evaluation of the transition probabilities, $q_i(x_i)$ and $q_z(z)$, will depend on details of the implementation. A normal approximation can be evaluated using a few operations by storing the appropriate sufficient statistics. Kernel density estimation will be more time-consuming and typically have number of operations that is $O(K)$. These consideration will become less important as the evaluation of the likelihood becomes more time-consuming (which occur in many problems since K will typically be relatively small).

2.2 Single copy idea

Section 2.1 discusses an algorithm based on the notion of constructing a joint distribution which contains K copies of the target distribution. Here, with a restriction on the $p(z|x_1, \ldots, x_K)$, we consider the notion of a "rolling" chain containing draws from a single copy of the target distribution. This is most conveniently discussed using a bivariate pair (x_1, x_2). The Gibbs and Metropolis switch update, described in Sect. 2.1, takes (x_1, x_2) to one of (x_1, z), (z, x_2) or (x_1, x_2). If we preserve symmetry in the conditional density for $p(z|x_1, x_2)$, that is $p(z|x_1, x_2) = p(z|x_2, x_1)$, then with this new state we can perform any permutation of the x's and preserve the same density. Thus, we can, if arriving at (z, x_2), switch these two around to leave us with (x_2, z). Now we can see we are taking (x_1, x_2) to one of (x_1, z), (x_2, z) or (x_1, x_2). In this way we can think of a single chain the transition of which has just been described. Hence, at any iteration, either the same pair are retained or one of the pair drops out to be replaced by z. The output of the chain will be x_2 at each iteration. The proposal now depends on the current and previous state. In summary, x_2 is the current state and x_1 a previous state. A proposal is made from $p(z|x_1, x_2)$ and either (x_1, x_2) remain, in which case x_2 stays as the current state and x_1 a previous state, or else z becomes the new current state and either x_1 or x_2, become a previous state.

This extends naturally to a K version by taking a z as in Sect. 2.1, where now the $(x_1, \ldots, x_K)$ are viewed as the current and $K-1$ previous states of a single chain, and proposing switching this z with one of these x. We assume x_K is the current state of the chain. So if the switch occurs z replaces one of the x_j and this x_j drops off. If there is symmetry, i.e.

$$p(z|x_1, \ldots, x_K) = p(z|x_{\sigma(1)}, \ldots, x_{\sigma(K)})$$

for any permutation σ on $\{1, \ldots, K\}$, then we can put z as the new current state. Hence, we have a "rolling" single chain where the proposal depends on the current and previous $K-1$ states of the chain. If the chain moves by accepting the proposal then this is the new state and is the object reported at that iteration. In this way it is seen that the traditional notion of a Metropolis step being allowed only to depend on the current state has been extended to include proposals based on the current and an arbitrary but fixed number of previous states. This is true provided the proposal satisfies the symmetry condition.

2.3 Metropolis-within-Gibbs updating

Metropolis–Hastings methods are often used to update parameters in a Gibbs sampling scheme. There has been little work applying adaptive MCMC methods in this context. Our approach is well-suited to extension to this more complicated situation since it is based on a Gibbs sampler and we illustrate the use of the multiple copies idea in this context. We now wish to sample $f(x)$ where x is a p-dimensional vector using the full conditional distributions $f_j(x_j|x_{-j})$

for $j = 1, 2, \ldots, p$. We suppose that the first m full conditional distributions will be sampled using a Metropolis–Hastings step and that the other full conditional distributions $f_{m+1}, f_{m+2}, \ldots, f_p$ can be sampled directly. We again use K copies of x where the jth element of the ith copy is $x_{i,j}$. This context is more complicated than directly sampling $f(x)$ using Metropolis–Hastings sampling since we have m separate Metropolis–Hastings algorithms (one for each of the first m parameters) and the copies of the jth element of x $(x_{1,j}, x_{2,j}, \ldots, x_{K,j})$, are no longer identically distributed conditional on $(x_{1,-j}, x_{2,-j}, \ldots, x_{K,-j})$ where $x_{i,-j}$ represents the ith copy with the jth element removed. It is natural to define a proposal for each element of each copy so that we have $p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})$ for all $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, K$ and adopt the joint density

$$\prod_{i=1}^{K} f(x_i) \prod_{j=1}^{m} p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p}).$$

Once again, marginalizing over all z_j returns us to the correct target distribution. The algorithm works as follows:

1. Perform the following steps for $j = 1, 2, \ldots, m$
   (i) Choose i uniformly from $1, 2, \ldots, K$.
   (ii) Sample z_j from the conditional density $p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})$.
   (iii) Propose the switch z_j with $x_{i,j}$ and accept with probability

$$\alpha = \min\left\{1, \frac{f(z_j|x_{i,-j})\, p(x_{i,j}|\{x'_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})}{f(x_{i,j}|x_{i,-j})\, p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})}\right\}$$

   where x' is defined by $x'_{i,j} = z_j$ and $x'_{k,l} = x_{k,l}$ if $k \neq i$ or $j \neq l$.
2. Perform the following steps for $j = m+1, m+2, \ldots, p$
   (i) Choose i uniformly from $1, 2, \ldots, K$.
   (ii) Sample $x_{i,j}$ from its full conditional distribution $f_j(x_{i,j}|x_{i,-j})$.
3. Record $\{x_{i,j}\}_{i=1,2,\ldots,K;\, j=1,2,\ldots,p}$ as output.

Note that we only need to simulate one z_j for every j at each iteration and that step 2 follows from marginalizing over z when updating these parameters.

The Metropolis–Hastings updating will be efficient if the proposal distribution

$$p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})$$

is chosen to approximate $f_j(x_{i,j}|x_{i,-j})$. We extend the idea, discussed in Sect. 2.1, of choosing the proposal by making inference about the target distribution (in this case the full conditional distribution) using the copies as a random sample. In this case, there are responses $x_{1,j}, x_{2,j}, \ldots, x_{K,j}$ and conditioning variables $x_{1,-j}, \ldots, x_{K,-j}$ and inference is naturally addressed by regression techniques. For example, linear regression or nonparametric regression methods, such as splines or kernel regression techniques could be used. It is important that the approximation of $f_j(x_{i,j}|x_{i,-j})$ is good. If we choose a parametric model then the relationship between the conditioning variables and the response must be well-represented (in our illustration we make a case that linear regression may be justified in some hierarchical models). If a nonparametric method is used then we will only be able to use a small number of the conditioning variables (to avoid the curse of dimensionality). This will often be the case in Bayesian inference. Firstly, assuming conditional independence will imply that $f_j(x_j|x_{-j})$ only depends on a subset of x_{-j} and, in many cases, we will not need to regress on those parameters whose posterior distribution is relatively concentrated (see Sect. 3.2).

The method has the same number of likelihood evaluations and samples from known distributions as a Gibbs sampler where $x_1, x_2, \ldots, x_m$ are updated using a Metropolis–Hastings algorithms. As with the sampler in Sect. 2.1, the number of additional operations depends on the choice of approximation. The sampler will only involve a few extra operations if the approximation involves a small number of sufficient statistics but will scale as $O(K)$ (or worse) for more complicated methods such as kernel density estimation or classical nonparametric regression methods. In practice, the operations needed for evaluation of the likelihood and transition probabilities will often dominate the operations needed in the steps where we can sample directly from the full conditional distribution and a better algorithm would update all copies of $x_{m+1}, x_{m+2}, \ldots, x_p$ from their full conditional distributions at every iteration. This algorithm works as follows:

1. Perform the following steps for $j = 1, 2, \ldots, m$
   (i) Choose i uniformly from $1, 2, \ldots, K$.
   (ii) Sample z_j from the conditional density $p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})$.
   (iii) Propose the switch z_j with $x_{i,j}$ and accept with probability

$$\alpha = \min\left\{1, \frac{f(z_j|x_{i,-j})\, p(x_{i,j}|\{x'_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})}{f(x_{i,j}|x_{i,-j})\, p(z_j|\{x_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p})}\right\}$$

   where x' is defined by $x'_{i,j} = z_j$ and $x'_{k,l} = x_{k,l}$ if $k \neq i$ or $j \neq l$.
2. For $j = m+1, m+2, \ldots, p$, $i = 1, 2, \ldots, K$, sample $x_{i,j}$ from its full conditional distribution $f_j(x_{i,j}|x_{i,-j})$.
3. Record $\{x_{i,j}\}_{i=1,2,\ldots,K;\, j=1,2,\ldots,p}$ as output.

The algorithms above use Gibbs sampling applied to univariate full conditional distributions. However, commonly-used blocking schemes where some variables are updated

jointly would be appropriate if some variables are highly correlated under the target distribution.

The following example illustrates the application of these methods to Bayesian inference. Suppose we wish to fit the hierarchical model

$$p(y_j|\mu_j), \quad p(\mu_j|\theta), \quad p(\theta), \quad j = 1, 2, \ldots, n$$

and so are interested in the posterior distribution of θ and $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ which has the joint density $f(\mu, \theta) \propto \prod_{j=1}^{n} p(y_j|\mu_j)\, p(\mu_j|\theta)\, p(\theta)$. We assume that θ can be directly sampled from its full conditional distribution $f(\theta|\mu) \propto \prod_{j=1}^{n} p(\mu_j|\theta)\, p(\theta)$ but that the full conditional of μ_j, $f(\mu_j|\theta) \propto p(y_j|\mu_j)p(\mu_j|\theta)$ cannot be easily sampled directly and will be updated using a Metropolis–Hastings step. Our approach defines the joint density of target and proposal to be

$$\prod_{j=1}^{n} p(z_j|\{\mu_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,n}, \theta_1, \ldots, \theta_K) \times \prod_{i=1}^{K} p(y_j|\mu_{i,j})\, p(\mu_{i,j}|\theta_i)\, p(\theta_i).$$

The steps for a single iteration of the algorithm are:

1. Perform the following steps for $j = 1, 2, \ldots, n$
   (i) Choose i uniformly from $1, 2, \ldots, K$
   (ii) Sample z_j from the conditional density
   $$p(z_j|\{\mu_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,n}, \theta_1, \theta_2, \ldots, \theta_K).$$
   (iii) Propose the switch z_j with $\mu_{i,j}$ and accept with probability
   $$\alpha = \min\left\{1, \frac{f(z_j|\theta_i)\, p(\mu_{i,j}|\{\mu'_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p}, \theta_1, \theta_2, \ldots, \theta_K)}{f(\mu_{i,j}|\theta_i)\, p(z_j|\{\mu_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p}, \theta_1, \theta_2, \ldots, \theta_K)}\right\}$$
   where μ' is defined by $\mu'_{i,j} = z_{i,j}$ and $\mu'_{k,l} = \mu_{k,l}$ if $k \neq i$ or $j \neq l$.
2. Sample θ_i from its full conditional distribution $f(\theta_i|\{\mu_{i,j}\}_{j=1,2,\ldots,n})$, for $i = 1, \ldots, K$.
3. Record $\theta_1, \theta_2, \ldots, \theta_K, \{\mu_{i,j}\}_{i=1,2,\ldots,K;\, j=1,2,\ldots,n}$ as output.

The full conditional distribution of $\mu_{i,j}$ simplifies in this model and only depends on θ_i. The choice of proposal is left to the user but it seems sensible to restrict the conditioning set in the proposal for $\mu_{i,j}$ to $\theta_1, \theta_2, \ldots, \theta_K$ and $\mu_{1,j}, \mu_{2,j}, \ldots, \mu_{K,j}$; i.e.,

$$p(z_j|\{\mu_{k,l}\}_{k=1,2,\ldots,K;l=1,2,\ldots,p}, \theta_1, \theta_2, \ldots, \theta_K) = p(z_j|\{\mu_{k,j}\}_{k=1,2,\ldots,K}, \theta_1, \theta_2, \ldots, \theta_K).$$

One simple choice would be for the proposal distribution to be the predictive distribution of $\mu_{i,j}$ at θ_i from a linear regression of $\mu_{1,j}, \mu_{2,j}, \ldots, \mu_{K,j}$ on $\theta_1, \theta_2, \ldots, \theta_K$, where the errors are assumed to be normally distributed. This is a univariate regression and so, potentially, more flexible nonparametric regression methods could be used.

Other standard MCMC techniques such as updating blocks of variables jointly or collapsing samplers can also be easily incorporated into this framework.

3 Illustrations

We consider three examples to illustrate the properties of the method. In the first example, a Metropolis–Hastings sampler is applied to a univariate target distribution $f(x)$. The performance of our method is compared to a random walk proposal and a standard adaptive Metropolis–Hastings independence sampler. We also demonstrate the new algorithm on a bi-modal target distribution where the separation of the modes is wide and hence standard Metropolis random walk algorithms will run into trouble.

The other example involves a Gibbs sampler with Metropolis-within-Gibbs steps. This involves posterior simulation of a generalized linear mixed model. The Monte Carlo error of estimating posterior means from the sampler output are used to measure the efficiency of the methods. The Monte Carlo errors are estimated by taking sample variances of repeated runs of the algorithms.

The effectiveness of MCMC methods is usually measured in terms of the number of iterations needed for the chain to reach equilibrium from a given point. In practice, our MCMC samplers are often started from a random point chosen from an initial distribution which is more dispersed than the target. The number of iterations needed for the chain to reach equilibrium from a point drawn from the initial distribution will be called the convergence time. This measure is important when determining a necessary burn-in time. Once the Markov chain is in equilibrium, mixing describes how quickly the Markov chain moves around the equilibrium distribution. This determines how long the chain must be run after the burn-in period to achieve some level of accuracy for Monte Carlo estimates.

3.1 Metropolis–Hastings samplers

3.1.1 Normal distribution

In this example, we compare different Metropolis–Hastings algorithms to simulate from a normal distribution with mean 0 and variance $1/100$. The samplers are: a Metropolis–Hastings random walk sampler with a normal proposal tuned to give an acceptance rate at the optimal level of 0.234, the

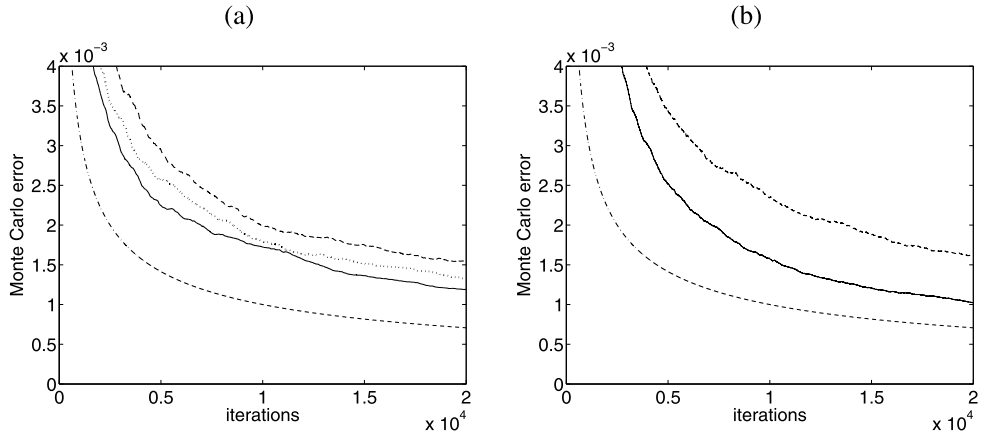

Fig. 1 Normal example: the Monte Carlo error for various samplers as a function of the number of iterations. The samplers shown in panel (**a**) are: the Adaptive scheme proposed in this paper with K copies for $K = 10$ (*dashed line*), $K = 20$ (*dotted line*), $K = 40$ (*solid line*) and the Monte Carlo estimate (*dot-dashed line*). The samplers shown in panel (**b**) are: Metropolis–Hastings random walk (*dashed line*), the Adaptive Monte Carlo scheme of Roberts and Rosenthal (*solid line*) and the Monte Carlo estimate (*dot-dashed line*)

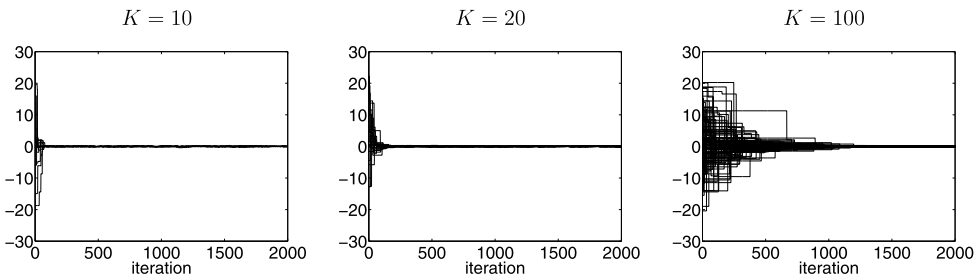

Fig. 2 Normal example: trace plots of sampled values for all copies of the target distribution with $K = 10$, $K = 20$ and $K = 100$

Adaptive Metropolis–Hastings scheme described in this paper with a normal approximation and different numbers of copies ($K = 10, 20, 40$) and the Adaptive scheme described in Roberts and Rosenthal (2009) who, after k iterations, use the proposal

$$p(x^\star|x) = \lambda \mathrm{N}(\mu_0, \sigma_0^2) + (1-\lambda)\mathrm{N}(\bar{x}_k, \hat{\sigma}_k^2)$$

where $\bar{x}_k = \sum_{i=1}^k x^{(i)}$ and $\hat{\sigma}_k^2 = \sum_{i=1}^k (x^{(i)} - \bar{x}_k)^2$. This is a mixture distribution where the second component is updated from the previous sample output and the first component is chosen to be over–dispersed relative to the target distribution (in our case we used $\mu_0 = 0$ and $\sigma_0^2 = 1$). The first component is introduced to stabilize the algorithm and λ is chosen to be relatively small (in this example, we choose $\lambda = 0.1$). In this example, the target distribution is normal and a normal approximation can perfectly fit the target distribution.

The Monte Carlo errors for the mean, starting the chain from the equilibrium distribution, are shown in Fig. 1, with the Monte Carlo error of a Monte Carlo estimate generating directly from the normal distribution shown as a dot-dashed line in both panels for comparison. The Adaptive Metropolis–Hastings sampler is more efficient than the random walk and bridges the gap between random walk samplers and a Monte Carlo sampler as the value of K increases, which indicates that the proposal distribution is providing an increasingly accurate approximation to the posterior distribution. The Adaptive Monte Carlo scheme of Roberts and Rosenthal (2009) is shown as the solid line in Panel (b) and ultimately outperforms the other adaptive samplers (as we would expect since it uses increasing amounts of information). However, the adaptive scheme with $K = 40$ is more efficient for small run sizes (until about 8000 iterations) and remains competitive for larger run sizes.

Clearly, the mixing of the chain improves with the number of copies used. However, the convergence of the sampler will slow as K increases. This effect is illustrated in Fig. 2, which shows trace plots of the output from multiple runs of the sampler for different values of K. The starting points were chosen from a normal distribution with mean 0 and variance 100 which is extremely over-dispersed relative to target distribution. The number of iterations before convergence strongly depends on the value of K. Convergence of all chains only seems to occur after about $10K$ iterations in this example.

In this method, there is a trade-off between better mixing after convergence (which improves with K) and time to convergence (which decreases with K). We suggest that values of K between 30 and 70 would be good benchmark choices which balance these two factors. However, if we could be

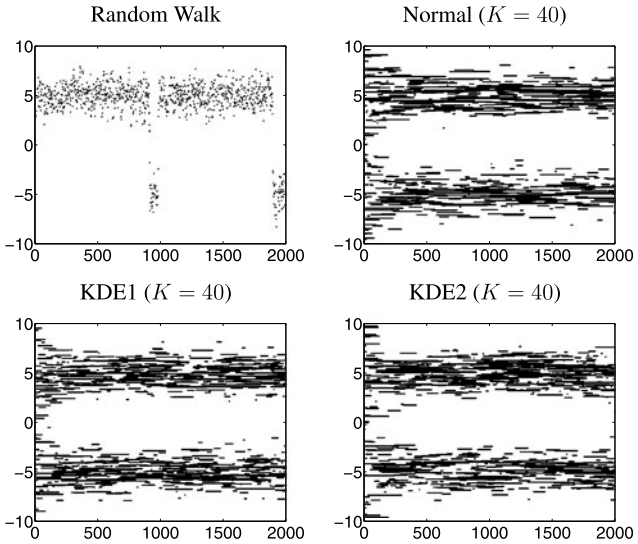

Fig. 3 Bi-modal example ($\mu = 5$): Trace plots from the samplers using the random walk sampler and multiple chain samplers with $K = 40$ and normal, KDE1 and KDE2 proposals

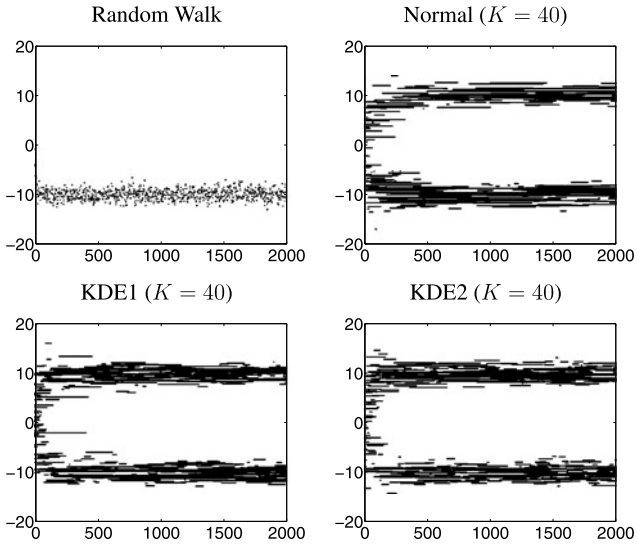

Fig. 4 Bi-modal example ($\mu = 10$): trace plots from the samplers using the random walk sampler and multiple chain samplers with $K = 40$ and normal, KDE1 and KDE2 proposals

certain that the starting values were drawn from a distribution close to the target distribution (and so remove the issue of convergence) then larger values of K would give better performance.

3.1.2 Bimodal example

Simulation from a bi-modal distribution is notoriously difficult. So we now consider sampling from a mixture of two normal distributions given by

$$p(x) = \frac{1}{2}\mathrm{N}(-\mu, 1) + \frac{1}{2}\mathrm{N}(\mu, 1).$$

Metropolis–Hastings random walk algorithms will often become stuck in one of the two modes as the probability of moving between the two modes decreases as μ increases. We consider several algorithms for the values $\mu = 5$ and $\mu = 10$. These are:

1. A Metropolis–Hastings random walk with normal proposal with a standard deviation of 2.
2. Our adaptive Metropolis–Hastings algorithm with the marginal distribution approximated using a normal distribution.
3. Our adaptive Metropolis–Hastings algorithm with the marginal distribution approximated by a kernel density estimate with the plug-in bandwidth of Silverman (1986) (KDE1).
4. Our adaptive Metropolis–Hastings algorithm with the marginal distribution approximated by a kernel density estimation with the bandwidth selection method of Botev et al. (2010) (KDE2) (Matlab code for this method is downloadable from MATLAB Central). This estimator has been shown to perform well in problems with bi-modal distributions.

The samplers were run for 20 000 iterations with the first 10 000 iterations discarded as a burn-in, which guarantees that the samplers have converged. Figure 3 shows trace plots for the different samplers when $\mu = 5$. The random walk sampler is able to move between the two modes but tends to spend long periods around one of the modes. In contrast, the Adaptive Monte Carlo scheme produces output around each mode due to the use of multiple chains. This is true for both kernel density estimate proposals and the normal proposal. Figure 4 shows trace plots when $\mu = 10$. The random walk samplers become trapped in a single mode and so produce a poor sample from the distribution. Once again the Adaptive Monte Carlo sampler generates samples from both modes throughout the run.

Figure 5 shows the Monte Carlo error for the mean of the target distribution with different numbers of iterations for the Adaptive samplers. The results when $\mu = 5$ are similar for the different proposals and the different numbers of chains. This suggests that the improvements in the approximation to the target distribution from having extra copies does not lead to greatly improved mixing. In fact, the normal approximation performs similarly to the kernel density estimates. The results for the more challenging example where $\mu = 10$ show greater differences. The normal approximation tends to perform poorly. The samplers with smaller numbers of chains seem to decay slowly suggesting that the sampler may not converge for some runs. The performance with KDE1 with $K = 40$ and $K = 80$ and KDE2 is better with an exponential decay in the Monte Carlo error. This suggests that the choice of the proposal can be critical to successful sampling in these challenging examples.

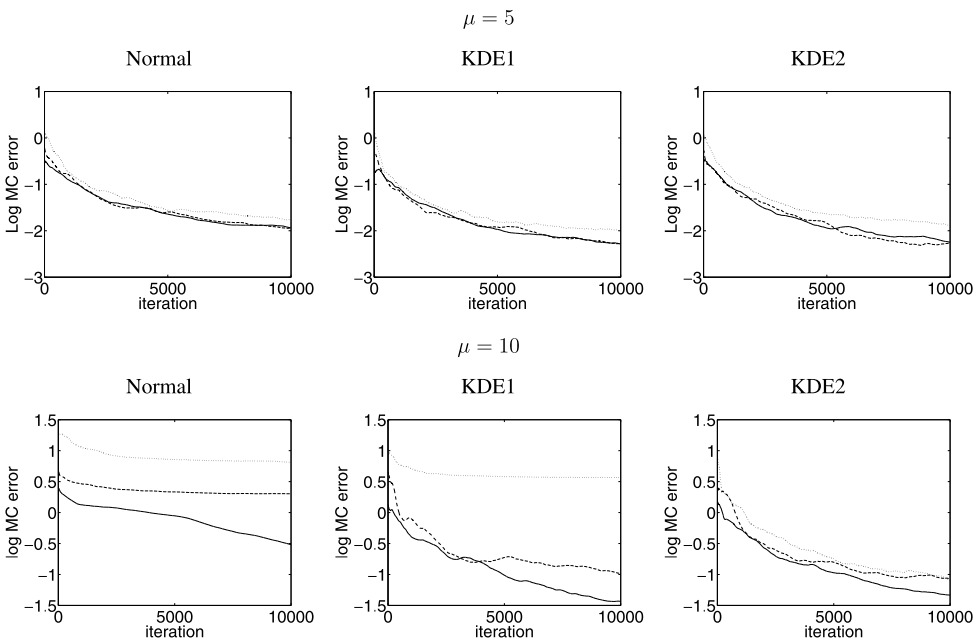

Fig. 5 Bi-modal example: the Monte Carlo error of approximating the mean with the sample mean using the first k iterations for three proposals: Normal, KDE1 and KDE2. The number of chains are $K = 20$ (*dotted line*), $K = 40$ (*dashed line*) and $K = 80$ (*solid line*)

3.2 Metropolis-within-Gibbs sampler: generalized linear mixed models

Metropolis–Hastings step are often used in Gibbs sampling schemes when some full conditional distributions are non-standard. In this example, we will consider using the adaptive sampler developed in this paper to simulate from these full conditionals. These problems can be considered potentially more challenging than the examples in Sect. 3.1 since the value of the conditioning variables (and so the distributions) will be different across the copies.

Here we consider posterior simulation from the following hierarchical model for the sample $((t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n))$,

$$y_j \sim \mathrm{Bi}(t_j, p_j), \qquad p_j = \frac{\exp\{x_j'\theta_j\}}{1+\exp\{x_j'\theta_j\}},$$

$$\theta_j \sim \mathrm{N}(\theta_0, \Sigma), \quad \theta_0 \sim \mathrm{N}(\mu, \Sigma_0).$$

where $\mathrm{Bi}(t, p)$ represents a Binomial distribution with t trials and success probability p; x_j is a p-dimensional vector of regressors, and θ is a p-dimensional vector of regression coefficients. The $(n+1)$-dimensional posterior distribution is defined on $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$. The parameter θ_0 can be updated directly from its full conditional distribution but the full conditional distribution of θ_j $(j = 1, 2, \ldots, n)$ cannot be sampled directly. The full conditional distributions for θ_j will typically be unimodal but may well be highly skewed with potentially heavy tails. The structure of the problem is the same as the example in Sect. 2.3.

We concentrate initially on $p = 1$ with Metropolis–Hastings updates and look at several proposal distributions. It is important to note that the Gibbs sampler framework for adaptation allows us to condition on θ_0 in the proposal for θ_j (in fact, we could condition on any θ_k for $k \neq j$). Let $\theta_{i,j}$ represent the value of θ_j in the ith copy. We consider the following proposals for updating $\theta_{i,j}$:

1. Normal approximation: $p(z_{i,j}) = \mathrm{N}(\hat{\theta}_j, \hat{\sigma}_j^2)$ where $\hat{\theta}_j = \frac{1}{K}\sum_{i=1}^K \theta_{i,j}$ and $\hat{\sigma}_j^2 = \frac{1}{K-1}\sum_{i=1}^K (\theta_{i,j} - \hat{\theta}_j)^2$.
2. Linear regression: $p(z_{i,j}) = \mathrm{N}(\hat{\theta}_{i,j}, \hat{\sigma}_j^2(1 + 1/n + (\theta_{0,i} - \bar{\theta}_0)^2/S_{xx}))$ where $\hat{\theta}_{i,j} = \hat{a}_j + \hat{b}_j(\theta_{0,i} - \bar{\theta}_0)$, $\hat{a}_j = \bar{\theta}_j = \frac{1}{K}\sum_{i=1}^K \theta_{i,j}$, $S_{xx} = \sum_{i=1}^K (\theta_{0,i} - \bar{\theta}_0)^2$, $\hat{b}_j = \sum_{i=1}^K (\theta_{0,i} - \bar{\theta}_0)(\theta_{i,j} - \bar{\theta}_j)/S_{xx}$ and $\hat{\sigma}_j^2 = \frac{1}{K-2}\sum_{j=1}^K (\theta_{i,j} - \hat{a}_j - \hat{b}_j(\theta_{0,i} - \bar{\theta}_0))^2$.

The first method proposes from an approximation of the marginal distribution of θ_j, which is over-dispersed relative to the target distribution $\theta_j|\theta_0$. The second method proposes from a linear regression of θ_j on θ_0, using the K copies.

The performance of the different samplers is compared by the Monte Carlo errors of the posterior mean of θ_0, that is, $\hat{\theta}_{0,N}$ with N iterations, for four different simulated samples from the model. In each case, $\theta_0 = 0.5$ and $\Sigma = 1/3$, but the distribution of the (t_i) and the value of n differ. The values of (t_i) are simulated from two distributions: (1) $t_i = \mathrm{round}(15 + 2\epsilon_i)$, where ϵ_i is standard normal; and (2) $t_i = 2 + \phi_i$, where ϕ_i is Gamma distributed with shape parameter 2 and mean 2. The first distribution has a small mean value of t_i, whilst the second distribution has a larger mean. The values of n are 40 and 100. Taking all combinations of the distributions of the (t_i), with the values of n, defines 4 different simulation scenarios.

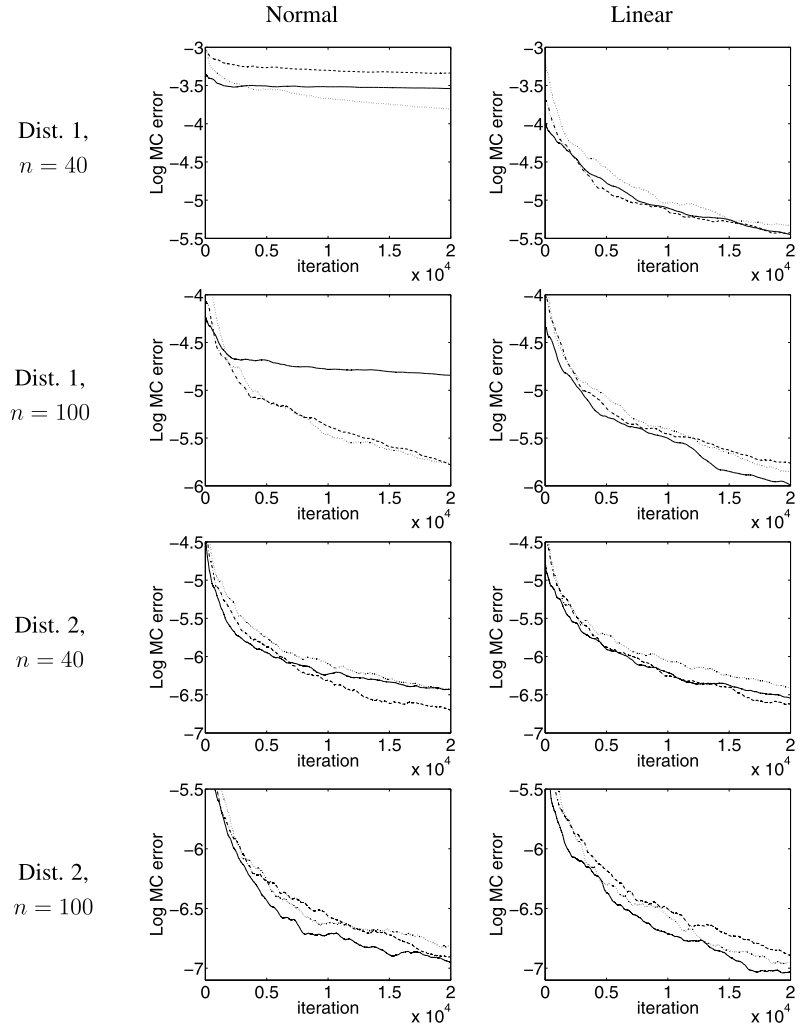

Fig. 6 GLMM: The logarithm of $\mathrm{SD}(\bar{\theta}_N)$ as a function of N with normal approximation and linear regression with different number of copies: with $K = 20$ (*dotted line*), $K = 40$ (*dot-dashed line*) and $K = 80$ (*solid line*) with univariate random effects

The results are presented in Fig. 6. There are some clear differences in performance. Firstly, the efficiency of all algorithms improves as n and the mean value of the (t_i) increase. This is because the posterior distribution of θ_0 is becoming increasingly concentrated. Secondly, for any given proposal, the performance improves as K increases. The linear regression proposal is better than the normal approximation in only one case; when n and the mean of the (t_i) are both small. In this case the posterior distribution of θ_0 is more dispersed than the other cases. To understand why the linear regression proposal works badly in other situations, consider the two properties of the posterior distribution of θ_0. Firstly, let $y_i^\star = \mathrm{logit}(y_i/n_i)$, then, if n_i is large, we can use the approximation $y_i^\star \sim \mathrm{N}(\theta_i, \sigma^{\star 2})$ and

$$\mathrm{E}(\theta_i|y_i^\star) = \frac{\sigma^2}{\sigma^2+\sigma^{\star 2}} y_i^\star + \frac{\sigma^{\star 2}}{\sigma^2+\sigma^{\star 2}}\theta_0.$$

Clearly, the regression will only be important if $\sigma^{\star 2}/(\sigma^2 + \sigma^{\star 2})$ is large, or when $\sigma^{\star 2}/\sigma^2$ is large. This will happen when the mean of n_i is small. Secondly, the conditional distribution of $\theta_i|\theta_0$ will be close to the marginal distribution of θ_i if the posterior variance of θ_0 is small. This occurs when n is large. Therefore, the regression is only useful when the mean of n_i is small and n is small.

We now consider the case where $p = 2$ and concentrate on $n = 40$ with Distribution 1, which is a case where the regression can significantly outperform the normal approximation. Increasing the dimension of random effect θ should decrease the amount of information from the data about each element of θ. The parameters were

$$\theta_0 = \begin{pmatrix} 0.5 \\ -0.2 \end{pmatrix} \quad \text{and} \quad \Sigma = 3\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Once again the proposals are a normal approximation to the marginal distribution of θ_i and a linear regression of θ_i on θ_0 (where now we have a multivariate linear regression extended in a standard way).

The results for $\rho = 0$ and $\rho = 0.5$ are presented in Fig. 7. They show that the linear regression approximation outperforms the normal approximation for both elements of θ_0. In

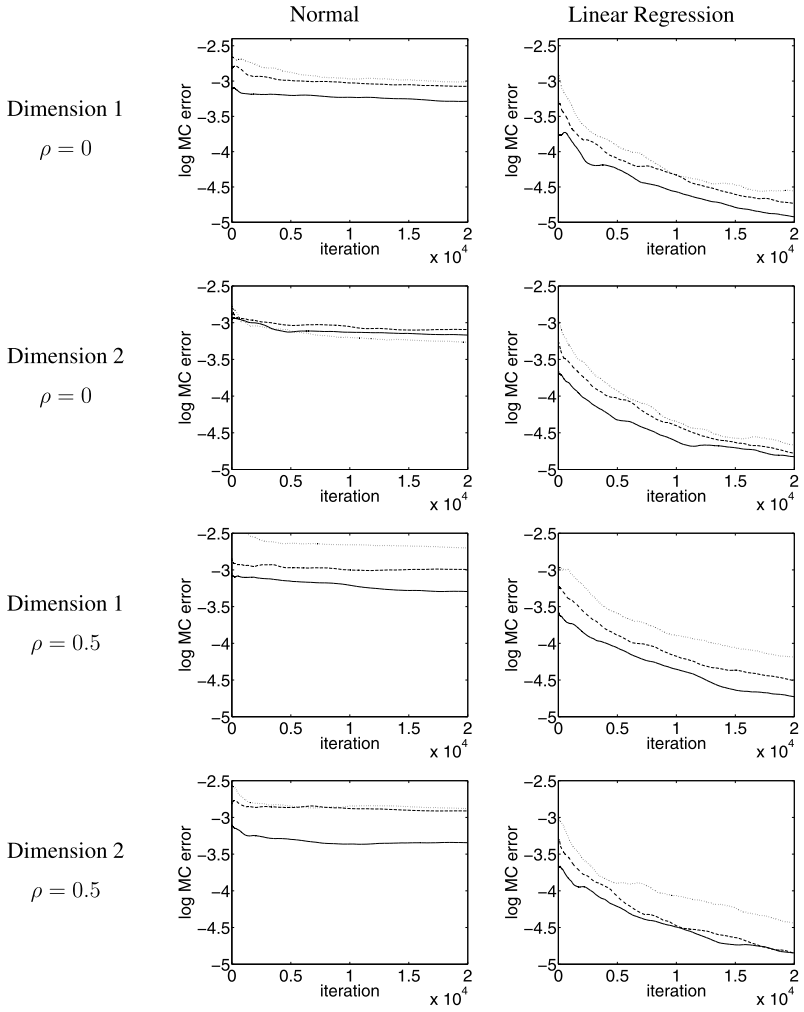

Fig. 7 GLMM: The SD($\bar{\theta}_N$) as a function of N with normal approximation (*dashed lines*) and linear regression (*solid line*) with different number of copies: with $K = 20$ (*dotted line*), $K = 40$ (*dashed line*) and $K = 80$ (*solid line*) with bivariate random effects for $\rho = 0$ and $\rho = 0.5$ with distribution 1 and $n = 40$

fact, increasing the number of copies K leads to improved performance for the linear regression proposal but can lead to inferior performance for the normal approximation (due to slower convergence).

Overall, the adaptive method described in this paper can be included in a Gibbs sampler to update parameters with non-standard full conditional distribution. The method works best when the dependence of the full conditional distribution on other parameters is included in the proposal. This can be effectively achieved using standard linear regression techniques.

4 Discussion

We have presented a framework in which it is possible to propose moves for a Metropolis–Hastings step which can depend on an arbitrary number of previous observations from the chain output. The Markov chain can be understood as a moving set of K pieces or as a sampler on a joint density constructed by combining the proposal distribution with K copies of the target distribution. The proposal is then defined as an approximation to the target distribution derived from the K samples. The convergence of this algorithm follows from standard theory and allows us to show that several previously proposed methods converge. Any current MCMC sampler with a Metropolis step can be easily modified to accommodate the more general proposal density.

The choice of K depends on a trade-off. Larger values of K lead to better mixing (due to higher acceptance rates associated with better mixing) but slower convergence (due to the larger number of K that have to be updated). The problems of convergence are reduced by using starting values drawn from a more dispersed distribution than the posterior.

The methods are also applied to Metropolis-with-Gibbs sampling. In this case, we can approximate the full conditional distribution by a normal distribution or using regression techniques. Often the full conditional distribution depends on a small number of parameters (or functions of parameters). Applications to different posterior distributions show that the method can lead to improved algorithms with improved mixing.

Acknowledgements The authors would like to thank two referees for constructive and helpful comments on an earlier version of the paper.

References

Andrieu, C., Moulines, D.: On the ergodicity properties of some adaptive MCMC algorithms. Ann. Appl. Probab. **16**, 1462–1505 (2006)

Andrieu, C., Thoms, J.: A tutorial on adaptive MCMC. Stat. Comput. **18**, 343–373 (2008)

Besag, J., Green, P.J.: Spatial statistics and Bayesian computation. J. R. Stat. Soc., Ser. B, Stat. Methodol. **55**, 25–37 (1993)

Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. Ann. Stat. **38**, 2916–2957 (2010)

Cai, B., Meyer, R., Perron, F.: Metropolis–Hastings algorithms with adaptive proposals. Stat. Comput. **18**, 421–433 (2008)

Cappé, O., Guillin, A., Marin, J.M., Robert, C.P.: Population Monte Carlo. J. Comput. Graph. Stat. **13**, 907–929 (2004)

Gasemyr, J.: On an adaptive version of the Metropolis–Hastings algorithm with independent proposal distribution. Scand. J. Stat. **30**, 159–173 (2003)

Gilks, W.R., Roberts, G.O., George, E.I.: Adapative direction sampling. The Statistician **43**, 179–189 (1994)

Giordani, P., Kohn, R.: Adaptive Independent Metropolis–Hastings by Fast Estimation of Mixtures of Normals. Technical Report. Available at SSRN: http://ssrn.com/abstract=1082955 (2008)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**, 223–242 (2001)

Haario, H., Saksman, E., Tamminen, J.: Componentwise adaptation for high dimensional MCMC. Comput. Stat. **20**, 265–273 (2005)

Jasra, A., Stephens, D.A., Holmes, C.C.: On population-based simulation for static inference. Stat. Comput. **17**, 263–279 (2007)

Keith, J.M., Kroese, D.P., Sofronov, G.Y.: Adaptive independence samplers. Stat. Comput. **18**, 409–420 (2008)

Laskey, B.M., Myers, J.W.: Population Markov chain Monte Carlo. Mach. Learn. **50**, 175–196 (2003)

Latuszynski, K., Roberts, G.O., Rosenthal, J.S.: Adaptive Gibbs samplers and related MCMC methods. Technical Report. University of Toronto (2011)

Mengersen, K.L., Robert, C.P.: Iid sampling with self-avoiding particle filters: the pinball sampler. In: Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., West, M. (eds.) Bayesian Statistics 7, pp. 277–292. Oxford University Press, Oxford (2003)

Roberts, G.O., Rosenthal, J.S.: Coupling and ergodicity of adaptive MCMC. J. Appl. Probab. **44**, 458–475 (2007)

Roberts, G.O., Rosenthal, J.S.: Examples of adaptive MCMC. J. Comput. Graph. Stat. **18**, 349–367 (2009)

Saksman, E., Vihola, M.: On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. Ann. Appl. Probab. **20**, 2178–2203 (2010)

Silverman, B.W.: Density Estimation. Chapman & Hall, London (1986)

Smith, A.F.M., Roberts, G.O.: Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion). J. R. Stat. Soc., Ser. B, Stat. Methodol. **55**, 3–24 (1993)

Tierney, L.: Markov chains for exploring posterior distributions. Ann. Stat. **22**, 1701–1786 (1994)

Warnes, G.R.: The normal kernel Coupler: an adaptive Markov chain Monte Carlo method for efficiently sampling form multi-modal distributions. Technical Report (2001)