

Original citation:

Tran , Minh-Ngoc, Pitt, Michael K. and Kohn, Robert V. Adaptive metropolis-Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*.

Permanent WRAP url:

<http://wrap.warwick.ac.uk/63212>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work of researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher statement:

© Springer. *Statistics and Computing*. The final publication is available at Springer via <http://dx.doi.org/10.1007/s11222-014-9509-6>

A note on versions:

The version presented here may differ from the published version or, version of record, if you wish to cite this item you are advised to consult the publisher's version. Please see the 'permanent WRAP url' above for details on accessing the published version and note that access may require a subscription.

For more information, please contact the WRAP Team at: publications@warwick.ac.uk

Adaptive Metropolis-Hastings Sampling using Reversible Dependent Mixture Proposals

Minh-Ngoc Tran · Michael K. Pitt · Robert Kohn

Abstract This article develops a general-purpose adaptive sampler that approximates the target density by a mixture of multivariate t densities. The adaptive sampler is based on reversible proposal distributions each of which has the mixture of multivariate t densities as its invariant density. The reversible proposals consist of a combination of independent and correlated steps that allow the sampler to traverse the parameter space efficiently as well as allowing the sampler to keep moving and locally exploring the parameter space. We employ a two-chain approach, in which a trial chain is used to adapt the proposal densities used in the main chain. Convergence of the main chain and a strong law of large numbers are proved under reasonable conditions, and without imposing a Diminishing Adaptation condition. The mixtures of multivariate t densities are fitted by an efficient Variational Approximation algorithm in which the number of components is determined automatically. The performance of the sampler is evaluated using simulated and real examples. Our autocorrelated framework is quite general and can handle mixtures other than multivariate t .

Keywords. Ergodic convergence; Markov Chain Monte Carlo; Metropolis-within Gibbs composite sampling; Mul-

tivariate t mixtures; Simulated annealing; Variational Approximation.

1 Introduction

Suppose that we wish to sample from a target distribution using a Metropolis-Hastings sampling method. For a traditional Metropolis-Hastings algorithm, where the proposal distribution is fixed in advance, it is well known that the success of the sampling method depends heavily on how the proposal distribution is selected. It is challenging to develop non-adaptive proposals in several types of problems. One example is when the target density is highly non-standard and/or multimodal. A second example is when the parameters are structural and deeply embedded in the likelihood so that it is difficult to differentiate the likelihood with respect to the likelihood; see for example Schmidl et al. (2013) who consider dynamic model with a regression function that is obtained as a solution to a differential equation. In such cases adaptive sampling, which sequentially updates the proposal distribution based on the previous iterates, has been shown useful. See, e.g., Haario et al. (1999, 2001); Roberts and Rosenthal (2007, 2009); Holden et al. (2009) and Giordani and Kohn (2010).

The chain generated from an adaptive sampler is no longer Markovian, and the convergence results obtained for traditional Markov chain Monte Carlo (MCMC) sampling no longer apply. Andrieu and Thoms (2008) warn that care must be taken in designing an adaptive sampler, as otherwise it may not converge to the correct distribution. They demonstrate this by constructing an adaptive chain that does not converge to the target. However, some theoretical results on the convergence of adaptive samplers are now available. A number of

M.-N. Tran
Australian School of Business
University of New South Wales, Australia
E-mail: minh-ngoc.tran@unsw.edu.au

M. K. Pitt
Department of Economics
University of Warwick, United Kingdom
E-mail: M.Pitt@warwick.ac.uk

R. Kohn
Australian School of Business
University of New South Wales, Australia
E-mail: r.kohn@unsw.edu.au

papers prove convergence by assuming that the adaptation eventually becomes negligible, which they call the Diminishing Adaptation condition. See, for example, Haario et al. (1999, 2001); Andrieu and Moulines (2006); Roberts and Rosenthal (2007) and Giordani and Kohn (2010). This condition is relatively easy to check if the adaptation is based on moment estimates as, for example, in an adaptive random walk, as in Haario et al. (1999, 2001) and Roberts and Rosenthal (2007) where the rate at which the adaptation diminishes is governed naturally by the sample size. It is more difficult to determine an optimal rate of adaptation when the adaptation is based on non-quadratic optimization as in Giordani and Kohn (2010) and in the Variational Approximation approach in our article.

Our article constructs a general-purpose adaptive sampler that we call the Adaptive Correlated Metropolis-Hastings (ACMH) sampler. The sampler is described in Section 4. The first contribution of the article is to propose a two-chain approach to construct the proposal densities, with the iterates of the first chain used to construct the proposal densities used in the second (main) chain. The ACMH sampler approximates the target density by a sequence of mixtures of multivariate t densities. The heavy tails of t and mixture of t distributions is a desirable property that a proposal distribution should have. Each mixture of t distribution is fitted by a Variational Approximation algorithm which automatically selects the number of components. Variational Approximation is now well known as a computationally efficient method for estimating complex density functions; see, e.g., McGrory and Titterton (2007) and Giordani et al. (2012). An attractive property of Variational Approximation that makes it suitable for constructing proposal distributions is that it can locate the modes quickly and efficiently.

The second, and main, contribution of the article is to introduce in Section 3 a method to construct reversible proposal densities, each of which has the mixture of t approximation as its invariant density. The proposal densities consist of both autocorrelated and independent Metropolis-Hastings steps. Independent steps allow the sampler to traverse the parameter space efficiently, while correlated steps allow the sampler to keep moving, (i.e., avoid getting stuck), while also exploring the parameter space locally. If the approximating t mixture is close to the target, then the reversible proposals introduced in our article will allow the sampler to move easily and overcome the low acceptance rates often encountered by purely independent proposals. Note that the reversible correlated proposals we introduce are quite general and it is only necessary to be able to generate from them but it is unnecessary to be able to

evaluate them. This is an important property for the correlated mixtures of t that we use.

The third contribution of the paper is to show in Section 2 that the ACMH sampler converges uniformly to the target density and to obtain a strong law of large numbers under reasonable conditions, without requiring that Diminishing Adaptation holds. As pointed out above, it is difficult to impose Diminishing Adaptation in a natural way for general proposals such as those in our paper.

Adaptive sampling algorithms can be categorized into two groups: exploitative and exploratory algorithms (Schmidler, 2011). Exploitative algorithms attempt to improve on features of the target distribution that have already been seen by the sampler, i.e. based on the past iterations to improve on what have been discovered by the past iterations. The adaptive samplers of Haario et al. (2001) and Giordani and Kohn (2010) belong to this group. The second group encourages exploring the whole support of the target, including tempering (Geyer and Thompson, 1995), simulated annealing (Neal, 2001) and the Wang-Landau algorithm (Wang and Landau, 2001a,b). It is therefore useful to develop a general-purpose adaptive sampler that can be both exploratory and exploitative. An important feature of the ACMH sampler is the use, in Section 5, of an exploratory stage to initialize the adaptive chain. In particular, we describe in this paper how to use simulated annealing (Neal, 2001) to initialize the chain. Giordani and Kohn (2010) suggest initializing the chain using either random walk steps or by using a Laplace approximation, neither of which work well for targets that are multimodal and/or have a non-standard support. Initializing by an exploratory algorithm helps the sampler initially explore efficiently the features of the target, and these features will be improved in the subsequent exploitative stage. Section 6 shows that such a combination makes the ACMH sampler work well for challenging targets where many other samplers may fail.

A second feature of the ACMH sampler is that it uses a small proportion of adaptive random walk steps in order to explore tail regions around local modes more effectively. A third feature of the ACMH sampler is that it uses Metropolis-within-Gibbs component-wise sampling to make the sampler move more efficiently in high dimensions, where it is often difficult to efficiently move the whole state vector as a single component because of the large differences in the values of the target and the proposal at the current and proposed states. See Johnson et al. (2011) and its references for some convergence results on such composite MCMC sampling.

Section 6 presents simulation studies and Section 7 applies the adaptive sampling scheme to estimate the

covariance matrix for a financial data set and analyze a spam email data set.

There are two important and immediate extensions of our work, which are discussed in Section 8. The first is to more general reversible mixture proposals. The second is to problems where the likelihood cannot be evaluated explicitly, but can be estimated unbiasedly.

Giordani and Kohn (2010) construct a general-purpose adaptive independent Metropolis-Hastings sampler that uses a mixture of normals as the proposal distribution. Their adaptive sampler works well in many cases because it is flexible and so helps the proposal approximate the target distribution better. They use the k -means algorithm to estimate the mixtures of normals. Although this method is fast, using independent Metropolis-Hastings steps only may result in a low acceptance rate that may not explore the local features of the state space as effectively. In addition, the automatic selection of components used in our article works appreciably better than using BIC, as done in Giordani and Kohn (2010).

de Freitas et al. (2001) use Variational Approximation to first estimate the target density and then use this approximation to form a fixed proposal density within an MCMC scheme. There are two problems with this approach, which are discussed in Section 4.1.

Holden et al. (2009) provide a framework for constructing adaptive samplers that ensures ergodicity without assuming Diminishing Adaptation. Not imposing Diminishing Adaptation is attractive because it means that the adaptation can continue indefinitely if new features of the target are learned. However, we believe that the Holden et al. (2009) framework is unnecessarily limited for two reasons. First, it does not use the information about the target obtained from dependent steps. Second, it augments the history on which the adaptation is based by using proposals that are rejected by the Metropolis-Hastings method; such inclusions typically lead to suboptimal adaptive proposals in our experience.

Hoogerheide et al. (2012) also use a multivariate mixture of t proposal densities which they fit using the EM algorithm. However, they stop adapting after a preliminary stage and do not have a principled way of choosing the number of components. In addition, their approach is harder to use when the likelihood cannot be computed, but can be estimated unbiasedly. Schmidl et al. (2013) propose an adaptive approach based on a vine copula but they stop adapting after a fixed number of adaptive steps. We note that it is straightforward to extend the multivariate mixture of t approach in Hoogerheide et al. (2012) and the copula approach

in Schmidl et al. (2013) to reversible proposals and to a two chain adaptive solution as in our article.

Craiu et al. (2009) also emphasize the importance of initial exploration and combine exploratory and exploitative stages using parallel an inter-chain adaptation algorithm to initially explore the sample space. They run many chains in parallel and let them interact in order to explore the modes, while we use annealed sampling with an SMC sampler. Their main contribution is the regional adaption algorithm, but they only discuss the case with two regions/two modes. It does not seem straightforward to extend the algorithm to general multimodal cases and it seems difficult to determine the number of regions/modes.

2 The adaptive sampling framework

2.1 Adaptive sampling algorithm

Let $\Pi(z)$ be the target distribution with corresponding density $\pi(z)$. We consider using a two-chain approach to adapt the proposal densities. The idea is to run simultaneously two chains, a trial chain X' and a main chain X , where the proposal densities used by the main chain X are estimated from the iterates of the trial chain X' , and are not based on the past iterates of chain X . We refer to the past iterates of X' used to estimate the proposal as the history vector, and denote by \mathcal{H}^{n-1} the history vector obtained after iteration $n-1$, which is used to compute the proposal density $q_{\mathcal{H}^{n-1}}(\cdot|\cdot)$ at iteration n . We consider the following general adaptive sampling algorithm.

Two-chain sampling algorithm.

1. Initialize the history \mathcal{H}^0 , the proposal $q_{\mathcal{H}^0}(\cdot|\cdot)$ and initialize x'_0, x_0 of the trial chain X' and main chain X , respectively.
2. For $n = 1, 2, \dots$

(a) Update the trial chain:

- Generate a proposal $z' \sim q_{\mathcal{H}^{n-1}}(z'|x'_{n-1})$.
- Compute

$$\alpha'_n(z', x'_{n-1}, \mathcal{H}^{n-1}) = \min\left(1, \frac{\pi(z')q_{\mathcal{H}^{n-1}}(x'_{n-1}|z')}{\pi(x'_{n-1})q_{\mathcal{H}^{n-1}}(z'|x'_{n-1})}\right).$$

- Accept z' with probability $\alpha'_n(z', x'_{n-1}, \mathcal{H}^{n-1})$ and set $x'_n = z'$, otherwise set $x'_n = x'_{n-1}$.
- Set $\mathcal{H}^n = (\mathcal{H}^{n-1}, x'_{n-1})$ if z' is accepted, otherwise set $\mathcal{H}^n = \mathcal{H}^{n-1}$.

(b) Update the main chain:

- Generate a proposal $z \sim q_{\mathcal{H}^{n-1}}(z|x_{n-1})$.

– Compute

$$\alpha_n(z, x_{n-1}) = \min \left(1, \frac{\pi(z)q_{\mathcal{H}^{n-1}}(x_{n-1}|z)}{\pi(x_{n-1})q_{\mathcal{H}^{n-1}}(z|x_{n-1})} \right).$$

– Set $x_n = z$ with probability $\alpha_n(z, x_{n-1})$, otherwise set $x_n = x_{n-1}$.

The ACMH sampler is based on this two-chain sampling framework and its convergence is justified by Corollary 3.

2.2 Convergence results

This section presents some general convergence results for adaptive MCMC. Suppose that E is a sample space with $x, z \in E$. \mathcal{E} is a σ -field on E . Suppose that $q_i(z|x)$ is the proposal density used at the i th iteration of the Metropolis-Hastings algorithm. In the two-chain algorithm above, q_i is the density $q_{\mathcal{H}^{i-1}}$ which is estimated based on the history \mathcal{H}^{i-1} . Let $\{x_n, n \geq 0\}$ be the Markov chain generated by the Metropolis-Hastings algorithm and $\mathbb{P}^n(x_0, \cdot)$ the distribution of the state x_n with the initial state x_0 . Denote by $p_i(x_{i-1}, dx_i)$ the Markov transition distribution at the i th iteration. We have the following convergence results whose proofs are in the Appendix.

Theorem 1 (Ergodicity) *Suppose that*

$$p_i(x_{i-1}, dx_i) \geq \beta \Pi(dx_i), \text{ for all } i \geq 1, \quad (1)$$

with $0 < \beta < 1$. Then

$$\|\mathbb{P}^n(x_0, \cdot) - \Pi(\cdot)\|_{TV} \leq 2(1 - \beta)^n \rightarrow 0 \text{ as } n \rightarrow \infty, \quad (2)$$

for any initial x_0 , where $\|\cdot\|_{TV}$ denotes the total variation distance.

Theorem 2 (Strong law of large numbers) *Suppose that $h(x)$ is a bounded function on E and that (1) holds. Let $S_n = \sum_{i=1}^n h(x_i)$. Then,*

$$\frac{S_n}{n} \rightarrow \mathbb{E}_{\Pi}(h) \text{ almost surely.} \quad (3)$$

Corollary 1 *Suppose that h is bounded. Theorems 1 and 2 hold for each of the following cases.*

- (i) $q_i(z|x) \geq \beta\pi(z)$ with $0 < \beta < 1$ for all $i \geq 1$ and $x, z \in E$.
- (ii) The proposal density $q_i(z|x)$ is a mixture of the form

$$q_i(z|x) = \omega q_{1,i}(z|x) + (1 - \omega)q_{2,i}(z|x),$$

$0 < \omega < 1$, and $q_{1,i}(z|x) \geq \beta\pi(z)$ with $0 < \beta < 1$ for all $z \in E$.

- (iii) Let $p_{1,i}(x_{i-1}, dx_i)$ and $p_{2,i}(x_{i-1}, dx_i)$ be transition distributions, each has stationary distribution π . Suppose that $p_{1,i}(x_{i-1}, dx_i)$ is based on the proposal density $q_{1,i}(z|x)$, where $q_{1,i}(z|x) \geq \beta\pi(z)$ for all $x, z \in E$ and $0 < \beta < 1$. The transition $p_i(x_{i-1}, dx_i)$ at the i th iterate is a mixture of the form

$$p_i(x_{i-1}, dx_i) = \omega p_{1,i}(x_{i-1}, dx_i) + (1 - \omega)p_{2,i}(x_{i-1}, dx_i),$$

with $0 < \omega < 1$.

- (iv) Let $p_{1,i}$ and $p_{2,i}$ be the transition distributions as in (iii). The transition at the i th iterate is a composition of the form

$$\begin{aligned} p_i(x_{i-1}, dx_i) &= p_{1,i} p_{2,i}(x_{i-1}, dx_i) \\ &= \int_z p_{1,i}(x_{i-1}, dz) p_{2,i}(z, dx_i), \end{aligned}$$

or

$$\begin{aligned} p_i(x_{i-1}, dx_i) &= p_{2,i} p_{1,i}(x_{i-1}, dx_i) \\ &= \int_z p_{2,i}(x_{i-1}, dz) p_{1,i}(z, dx_i), \end{aligned}$$

- (v) Let $p_{1,i}$ and $p_{2,i}$ be the transition distributions as in (iii). The transition at the i th iterate is a composition of m_1 repetitions of $p_{1,i}$ and m_2 repetitions of $p_{2,i}$, i.e. $p_i = p_{1,i}^{m_1} p_{2,i}^{m_2}$ or $p_i = p_{2,i}^{m_2} p_{1,i}^{m_1}$.

3 Reversible proposals

In the Metropolis-Hastings algorithm, it is desirable to have a proposal that depends on the current state, is reversible, and marginally has an invariant distribution of choice. We refer to such a proposal as a reversible proposal. The dependence between the current and proposed states helps in moving locally and helps the chain mix more rapidly and converge. As will be seen later, reversibility simplifies the acceptance probability in the MH algorithm and makes it close to one if the marginal distribution is a good approximation to the target. Reversibility also means that it is only necessary to be able to generate from the proposal distribution, and it is unnecessary to be able to evaluate it. This is important in our case because the proposal densities are mixtures of conditional t densities with dependence parameters that are integrated out. Section 3.1 provides the theory for reversible proposals that we use in our article. Section 3.2 introduces a reversible multivariate t density and a reversible mixture of multivariate t densities. The proofs of all results in this section are in the Appendix.

3.1 Some theory for reversible proposals

Definition 1 (Reversible transition density) Suppose that $\zeta(z)$ is a density in z and $T(z|x)$ is a transition density from x to z such that

$$\zeta(x)T(z|x) = \zeta(z)T(x|z) \quad \text{for any } x \text{ and } z.$$

Then,

$$\zeta(z) = \int \zeta(x)T(z|x)dx,$$

and we say that $T(z|x)$ is a reversible Markov transition density with invariant density $\zeta(z)$.

The following two lemmas provide some properties of reversible transition densities that are used in our work.

Lemma 1 (Properties of reversible transition densities) Suppose that $\zeta(z)$ is a density in z . Then, in each of the cases described below, $T(z|x)$ is a reversible Markov transition density with invariant density $\zeta(z)$.

- (i) $T(z|x) = \zeta(z)$.
- (ii) Let $z = (z_A, z_B)$ be a partition of z and define $T(z|x) = \zeta(z_A|z_B)I(z_B = x_B)$, where $I(z_B = x_B)$ is an indicator variable that is 1 if $z_B = x_B$ and is zero otherwise.
- (iii) Suppose that for each parameter value ρ , $T(z|x; \rho)$ is a reversible Markov transition density with invariant density $\zeta(z)$. Let $T(z|x) = \int T(z|x; \rho)\lambda(d\rho)$, where $\lambda(d\rho)$ is a probability measure in ρ .

The next lemma gives a result on a mixture of transition densities each having its own invariant density.

Lemma 2 (Mixture of reversible transition densities)

- (i) Suppose that for each $k = 1, \dots, G$, $T_k(z|x)$ is a reversible Markov transition kernel with invariant density $\zeta_k(z)$. Define the mixture density $\zeta(z)$ and the mixture $T(z|x)$ of transition densities as

$$\zeta(z) = \sum_{k=1}^G \omega_k \zeta_k(z) \quad \text{and} \quad T(z|x) = \sum_{k=1}^G \omega(k|x) T_k(z|x),$$

where $\omega_1 + \dots + \omega_G = 1$, $\omega_k \geq 0$ and $\omega(k|x) = \omega_k \zeta_k(x) / \zeta(x)$ for all $k = 1, \dots, G$. Then, $T(z|x)$ is a reversible Markov transition density with invariant density $\zeta(z)$.

- (ii) If the invariant densities $\zeta_k(z)$ are all the same, then $\omega(k|x) = \omega_k$ for all k and $\zeta(z) = \zeta_1(z)$.
- (iii) Suppose that $T(z|x)$ is a reversible Markov transition density with invariant density $\zeta(z)$. Then $q(z|x) = \omega \zeta(z) + (1 - \omega)T(z|x)$, $0 \leq \omega \leq 1$, is a reversible Markov transition density with invariant density $\zeta(z)$.

Corollary 2 (Mixture of conditional densities) Let $z = (z_A, z_B)$ be a partition of z and define $T_k(z|x) = \zeta_k(z_A|z_B)I(z_B = x_B)$, for $k = 1, \dots, G$. Then, each $T_k(z|x)$ is a reversible density with invariant density $\zeta_k(z)$, and Lemma 2 holds.

The next lemma shows the usefulness of using reversible Markov transition densities as proposals in a Metropolis-Hastings scheme.

Lemma 3 (Acceptance probability for a reversible proposal) Consider a target density $\pi(z)$. We propose z , given the state x from the reversible Markov transition density $T(z|x)$, which has invariant density $\zeta(z)$. Then, the acceptance probability of the proposal is

$$\alpha(z, x) = \min \left\{ 1, \frac{\pi(z)\zeta(x)}{\pi(x)\zeta(z)} \right\}.$$

The lemma shows that the acceptance probability has the same form as for an independent proposal from the invariant density $\zeta(z)$, even though the proposal may depend on the previous state and on parameters that are in the transition density $T(z|x)$ but not in $\zeta(z)$. This means the following: (i) To compute the acceptance probability $\alpha(z, x)$, it is only necessary to be able to simulate from $T(z|x)$, and it is unnecessary to be able to compute it. This is useful for our work where we cannot evaluate $T(z|x)$ analytically because it is a mixture over a parameter ρ . (ii) The acceptance probability will be high if the invariant density $\zeta(z)$ is close to the target density $\pi(z)$. In fact, if $\zeta(z) = \pi(z)$, then the acceptance probability is 1.

3.2 Constructing reversible t distributions

Pitt and Walker (2006) construct a univariate Markov transition which has a univariate t distribution as the invariant distribution. We now extend this approach to construct a Markov transition density with a multivariate t density as its invariant distribution. This reversible multivariate t process is new to the literature. We then generalize it to the case in which the invariant distribution is a mixture of multivariate t distributions. We denote by $t_d(z; \mu, \Sigma, \nu)$ the d -variate t density with location vector μ , scale matrix Σ and degrees of freedom ν .

Lemma 4 (Reversible t transition density) Let $\zeta(z; \psi) = t_d(z; \mu, \Sigma, \nu)$ and $T(z|x; \psi, \rho) = t_d(z; \tilde{\mu}(x), \tilde{\Sigma}(x), \tilde{\nu})$, where ψ is the set of parameters $\{\mu, \Sigma, \nu\}$, ρ is a cor-

relation coefficient,

$$\begin{aligned}\tilde{\mu}(x) &= (1 - \rho)\mu + \rho x, \\ \tilde{\Sigma}(x) &= \frac{\nu}{\nu + d}(1 - \rho^2) \left(1 + \frac{1}{\nu}(x - \mu)' \Sigma^{-1}(x - \mu) \right) \Sigma, \\ \tilde{\nu} &= \nu + d.\end{aligned}\tag{4}$$

Then,

- (i) For each fixed ρ , $T(z|x; \psi, \rho)$ is a reversible Markov transition density with invariant density $\zeta(z; \psi)$.
- (ii) Let $T(z|x; \psi) = \int T(z|x; \psi, \rho) \lambda(d\rho)$, where $\lambda(d\rho)$ is a probability measure. Then, $T(z|x; \psi)$ is a reversible Markov transition density with invariant density $\zeta(z; \psi)$.

We now follow Lemma 2 and define a reversible transition density that is a mixture of reversible t transition densities. Suppose $\zeta_k(z; \psi_k) = t_d(z; \mu_k, \Sigma_k, \nu_k)$ and $T_k(z|x; \psi_k, \rho_k) = t_d(z; \tilde{\mu}_k(x), \tilde{\Sigma}_k(x), \tilde{\nu}_k)$, where $\tilde{\mu}_k(x)$, $\tilde{\Sigma}_k(x)$ and $\tilde{\nu}_k$ are defined in terms of $(\mu_k, \Sigma_k, \rho_k)$ as in (4). Let $\psi = \{\psi_1, \dots, \psi_G\}$ and

$$T_k(z|x; \psi_k) = \int T_k(z|x; \psi_k, \rho_k) \lambda_k(d\rho_k).$$

Lemma 5 (Mixture of t transition densities) *Let*

$$\begin{aligned}g_M(z; \psi) &= \sum_{k=1}^G \omega_k \zeta_k(z; \psi_k) \text{ and} \\ T_{g_M}^{CMH}(z|x; \psi) &= \sum_{k=1}^G \omega(k|x) T_k(z|x; \psi_k),\end{aligned}\tag{5}$$

where $\omega(k|x) = \omega_k \zeta_k(x; \psi_k) / g_M(x; \psi)$. Then,

- (i) $g_M(z; \psi)$ is a mixture of t densities and $T_{g_M}^{CMH}(z|x; \psi)$ is a mixture of t transition densities with $T_{g_M}^{CMH}(z|x; \psi)$ a reversible transition density with invariant $g_M(z; \psi)$;
- (ii) if the proposal density is $T_{g_M}^{CMH}(z|x; \psi)$ and the target density is $\pi(z)$, then the Metropolis-Hastings acceptance probability is

$$\alpha(z, x) = \min \left\{ 1, \frac{\pi(z) g_M(x; \psi)}{\pi(x) g_M(z; \psi)} \right\}.\tag{6}$$

We note that it is straightforward to generate from $T_{g_M}^{CMH}(z|x; \psi)$, given x , because it is a mixture of transition densities, each of which is a mixture. However, it is difficult to compute $T_{g_M}^{CMH}(z|x; \psi)$ because it is difficult to compute each of $T_k(z|x; \psi_k)$ as it is a t density mixed over ρ_k . However, by Part (ii) of Lemma 5, it is straightforward to compute the acceptance probability $\alpha(z, x)$.

3.3 Constructing reversible mixtures of conditional t densities

Suppose that the vector z has density $g_M(z; \psi)$ which is a mixture of multivariate t densities as in equation (5), with $\zeta_k(z; \psi_k) = t_d(z; \mu_k, \Sigma_k, \nu_k)$. We partition z as $z = (z_A, z_B)$, where z_A is $d_A \times 1$ and we partition the μ_k and Σ_k conformally, as

$$\mu_k = \begin{pmatrix} \mu_{k,A} \\ \mu_{k,B} \end{pmatrix} \text{ and } \Sigma_k = \begin{pmatrix} \Sigma_{k,AA} & \Sigma_{k,AB} \\ \Sigma_{k,BA} & \Sigma_{k,BB} \end{pmatrix}, \quad k = 1, \dots, G.$$

Then, $\zeta_k(z_A|z_B; \psi_k) = t_{d_A}(z_A; \tilde{\mu}_k(z_B), \tilde{\Sigma}_k, \tilde{\nu}_k)$, where $\tilde{\mu}_k(z_B) = \mu_{k,A} + \Sigma_{k,AB} \Sigma_{k,BB}^{-1} (z_B - \mu_{k,B})$, $\tilde{\Sigma}_k = \Sigma_{k,AA} - \Sigma_{k,AB} \Sigma_{k,BB}^{-1} \Sigma_{k,BA}$, and $\tilde{\nu}_k = \nu_k + d_A$.

Lemma 6 (Reversible mixture of conditional densities) *Define the transition kernel*

$$T_{g_M}^{BS}(z|x; \psi) = \sum_{k=1}^G \frac{\omega_k \zeta_k(x; \psi_k)}{g_M(x; \psi)} \zeta_k(z_A|z_B; \psi_k) I(z_B = x_B).\tag{7}$$

Then $T_{g_M}^{BS}(z|x; \psi)$ is reversible with invariant density $g_M(z; \psi)$.

4 The adaptive correlated Metropolis-Hastings (ACMH) sampler

The way we implemented the ACMH sampler is now described, although Sections 2.2 and 3 allow us to construct the sampling scheme in a number of ways. Section 4.1 outlines the Variational Approximation method for estimating mixtures of multivariate t distributions. Sections 4.2 and 4.3 discuss component-wise sampling and adaptive random walk sampling. Section 4.4 summarizes the ACMH sampler.

4.1 Estimating mixtures of multivariate t densities

Given a mixture of multivariate t densities, Section 3.2 describes a method to construct reversible mixtures of t . This section outlines a fast Variational Approximation method for estimating such a mixture of t .

Suppose that $p(D|\theta)$ is the likelihood computed under the assumption that the data generating process of D is a mixture of t density $m(x|\theta)$, with θ its parameters. Let $p(\theta)$ be the prior, then Bayesian inference is based on the posterior $p(\theta|D)$, which is often difficult to handle. Variational Approximation approximates this posterior by a more tractable distribution

$q_{\text{va}}(\theta)$ by minimizing the Kullback-Leibler divergence

$$\text{KL}(q_{\text{va}}) = \int \log \frac{q_{\text{va}}(\theta)}{p(\theta|D)} q_{\text{va}}(\theta) d\theta,$$

among some restricted class of densities $q_{\text{va}} \in \mathcal{Q} = \{q_{\text{va}}(\cdot|\lambda), \lambda \in \Lambda\}$. Because,

$$\log p(D) = \int \log \frac{p(\theta)p(D|\theta)}{q_{\text{va}}(\theta)} q_{\text{va}}(\theta) d\theta \quad (8)$$

$$+ \int \log \frac{q_{\text{va}}(\theta)}{p(D|\theta)} q_{\text{va}}(\theta) d\theta, \quad (9)$$

minimizing $\text{KL}(q_{\text{va}})$ is equivalent to maximizing

$$L(\lambda) = \int \log \frac{p(\theta)p(D|\theta)}{q_{\text{va}}(\theta|\lambda)} q_{\text{va}}(\theta|\lambda) d\theta. \quad (10)$$

Because of the non-negativity of the Kullback-Leibler divergence term in (9), (10) is a lower bound on $\log p(D)$. We refer the reader to Tran et al. (2013) who describe in detail how to fit mixtures of t using Variational Approximation, in which the number of components is automatically selected using the split and merge algorithm by maximizing the lower bound. The accuracy of Variational Approximation is experimentally studied in Nott et al. (2012). See also Corduneanu and Bishop (2001) and McGrory and Titterton (2007) who use Variational Approximation for estimating mixtures of normals.

Denote by $\hat{\lambda}$ the maximizer of (10), the posterior $p(\theta|D)$ is approximated by $q_{\text{va}}(\theta|\hat{\lambda})$. From our experience, the estimate $q_{\text{va}}(\theta|\hat{\lambda})$ often has a small tail, but it can quickly locate the mode of the true posterior $p(\theta|D)$. In our context, $m(x|\hat{\theta})$ with $\hat{\theta}$ the mode of $q_{\text{va}}(\theta|\hat{\lambda})$ is used as the mixture of t in the ACMH sampler.

We now explain more fully the main difference between the Variational Approximation approach to constructing proposal densities of de Freitas et al. (2001) and our approach. de Freitas et al. (2001) estimate $\pi(x)$ directly as $\hat{\pi}_{\text{vb}}(x)$ using Variational Approximation, i.e. $\hat{\pi}_{\text{vb}}$ minimizes

$$\text{KL}(\pi_{\text{va}}) = \int \log \frac{\pi_{\text{va}}(x)}{\pi(x)} \pi_{\text{va}}(x) dx$$

among some restricted class of densities, such as normal densities. de Freitas et al. (2001) then use $\hat{\pi}_{\text{vb}}(x)$ to form the fixed proposal density. The estimate $\hat{\pi}_{\text{vb}}(x)$ often has much lighter tails than $\pi(x)$ (see, e.g., de Freitas et al., 2001), therefore such a direct use of Variational Approximation estimates for the proposal density in MCMC can be problematic. Another problem with their approach is that $\hat{\pi}_{\text{vb}}(x)$ needs to be derived afresh for each separate target density $\pi(x)$ and this may be difficult for some targets. In our approach we

act as if the target density $\pi(x)$ is a t mixture $m(x|\theta)$ with parameters θ , and obtain a point estimate of θ using Variational Approximation. The approach is general because it is the same for all targets, and does not suffer from the problem of light tails.

4.2 Metropolis within Gibbs component-wise sampling

In high dimensions, generating the whole proposal vector at the one time may lead to a large difference between the values of the target π , and the proposal q , at the proposed and current states. This may result in high rejection rates in the Metropolis-Hastings algorithm, making it hard for the sampling scheme to move. To overcome this problem, we use Metropolis within Gibbs component-wise sampling in which the coordinates of x are divided into two or more components at each iterate. Without loss of generality, it is only necessary to consider two components, a component x_B that remains unchanged and a complementary component x_A generated conditional on x_B . We will refer to B and A as the index vectors of x_B and x_A respectively. Let d_B and d_A be the dimensions of B and A , respectively. We note that B and A can change at random or systematically from iteration to iteration. See Johnson et al. (2011) for a further discussion of Metropolis within Gibbs sampling.

We can carry out a Metropolis within Gibbs sampling step based on reversible mixtures of conditional t distributions as in Section 3.3.

In some applications, there are natural groupings of the parameters, such as the group of mean parameters and the group of variance parameters. Otherwise, the coordinates x_B can be selected randomly. For example, each coordinate is independently included in B with probability p_B . The number of coordinates d_A in x_A should be kept small in order for the chain to move easily. We find that it is useful to set p_B so that the expected value of d_A is about 10, i.e., $p_B \approx 1 - 10/d$.

4.3 The adaptive random walk Metropolis-Hastings proposal

Using mixtures of t for the proposal distribution helps to quickly and efficiently locate the modes of the target distribution. In addition, it is useful to add some random walk Metropolis-Hastings steps to explore more effectively the tail regions around the local modes; see Section 6.1.

We use the following version of an adaptive random walk step, which takes into account the potential multimodality of the target. Let x be the current

state and $g_M(x;\psi)$ the latest mixture of t as in (5). Let $\hat{k}(x;\psi) = \operatorname{argmax}_k \{\omega_k \zeta_k(x;\psi_k)\}$, i.e. $\hat{k}(x;\psi)$ is the index of the component of the mixture that x is most likely to belong to. Let $\phi_d(x;a,B)$ be a d -variate normal density with mean vector a and covariate matrix B . The random walk proposal density is $q^{RW}(z|x;\hat{k}(x;\psi)) = \phi_d(z;x,\kappa \tilde{\Sigma}_{\hat{k}(x;\psi)})$, where

$$\tilde{\Sigma}_{\hat{k}(x;\psi)} = \nu_{\hat{k}(x;\psi)} / (\nu_{\hat{k}(x;\psi)} - 2) \Sigma_{\hat{k}(x;\psi)}$$

if $\nu_{\hat{k}(x;\psi)} > 2$ and is equal to $\Sigma_{\hat{k}(x;\psi)}$ otherwise. The scaling factor $\kappa = 2.38^2/d$ (see Roberts and Rosenthal, 2009).

4.4 Description of the ACMH sampler

This section gives the details of the ACMH sampler, which follows the framework in Section 2 to ensure convergence. The sampler consists of a reversible proposal density together with a random walk proposal. We shall first describe the reversible proposal density. Let $g_0(z)$ be the heavy tailed component and $g_M(z;\hat{\psi}(\mathcal{H}^{n-1}))$ the mixture of t densities described in (5), where \mathcal{H}^{n-1} is the history vector obtained after iteration n based on the trial chain X' and $\hat{\psi}$ is the estimate of ψ based on \mathcal{H}^{n-1} . Let $T_{g_0}(z|x) = g_0(z)$ be the reversible transition density whose invariant density is $g_0(z)$ (see Part (i) of Lemma 1). Let $T_{g_M}^{\text{CMH}}(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ be the correlated reversible transition density defined in equation (5) and let $T_{g_M}^{\text{BS}}(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ be the component-wise mixture reversible transition density defined in (7). We now define the mixtures,

$$\begin{aligned} T_{g_M}(z|x;\hat{\psi}(\mathcal{H}^{n-1})) &= (1-\gamma)T_{g_M}^{\text{CMH}}(z|x;\hat{\psi}(\mathcal{H}^{n-1})) + \\ &\gamma T_{g_M}^{\text{BS}}(z|x;\hat{\psi}(\mathcal{H}^{n-1})), \\ q^*(z;\hat{\psi}(\mathcal{H}^{n-1})) &= \beta_0 g_0(z) + (1-\beta_0)g_M(z;\hat{\psi}(\mathcal{H}^{n-1})), \\ T_{q^*}(z|x;\hat{\psi}(\mathcal{H}^{n-1})) &= \frac{\beta_0 g_0(x)}{q^*(x;\hat{\psi}(\mathcal{H}^{n-1}))} T_{g_0}(z|x) + \\ &\frac{(1-\beta_0)g_M(x;\hat{\psi}(\mathcal{H}^{n-1}))}{q^*(x;\hat{\psi}(\mathcal{H}^{n-1}))} T_{g_M}(z|x;\hat{\psi}(\mathcal{H}^{n-1})), \end{aligned}$$

and

$$\begin{aligned} q(z|x;\hat{\psi}(\mathcal{H}^{n-1})) &= \delta q^*(z;\hat{\psi}(\mathcal{H}^{n-1})) \\ &+ (1-\delta)T_{q^*}(z|x;\hat{\psi}(\mathcal{H}^{n-1})) \\ &= \delta \beta_0 g_0(z) + \delta(1-\beta_0)g_M(z;\hat{\psi}(\mathcal{H}^{n-1})) + \\ &(1-\delta)\beta_0 \frac{g_0(x)}{q^*(x;\hat{\psi}(\mathcal{H}^{n-1}))} T_{g_0}(z|x) + \\ &(1-\delta)(1-\beta_0) \frac{g_M(x;\hat{\psi}(\mathcal{H}^{n-1}))}{q^*(x;\hat{\psi}(\mathcal{H}^{n-1}))} T_{g_M}(z|x;\hat{\psi}(\mathcal{H}^{n-1})), \end{aligned} \tag{11}$$

with $0 \leq \gamma, \beta_0, \delta \leq 1$. Note that δ is the probability of generating an independent proposal and γ is related to the probability of doing component-wise sampling. Then,

- Lemma 7** (i) $T_{g_M}(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ is a reversible Markov transition density with invariant density $g_M(z;\hat{\psi}(\mathcal{H}^{n-1}))$.
(ii) $T_{q^*}(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ is a reversible Markov transition density with invariant density $q^*(z;\hat{\psi}(\mathcal{H}^{n-1}))$.
(iii) $q(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ is a reversible Markov transition density with invariant density $q^*(z;\hat{\psi}(\mathcal{H}^{n-1}))$.
(iv) If $q(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ is a proposal density with target density $\pi(z)$, then the acceptance probability is

$$\alpha(z,x;\hat{\psi}(\mathcal{H}^{n-1})) = \min \left\{ 1, \frac{\pi(z) q^*(x;\hat{\psi}(\mathcal{H}^{n-1}))}{\pi(x) q^*(z;\hat{\psi}(\mathcal{H}^{n-1}))} \right\}.$$

Description of the ACMH proposal density. The ACMH sampler consists of a reversible proposal density together with a random walk proposal. Let $p_n(x_{n-1}, dx_n)$ be the transition kernel at iteration n of the main chain X . Denote by $p_{1,n}, p_{2,n}$ the transition kernel with respect to the reversible proposal $q(z|x;\hat{\psi}(\mathcal{H}^{n-1}))$ and the random walk proposal $q^{AR}(z|x;\hat{k}(x;\hat{\psi}(\mathcal{H}^{n-1})))$ respectively.

- (1) $p_n = p_{1,n} p_{2,n}$ at $n = \iota_{RW}, 2\iota_{RW}, \dots$ (see Corollary 1 (iv)). That is, a composition of a correlated Metropolis-Hastings step with reversible proposal and a random walk step is performed after every $\iota_{RW} - 1$ iterations. In our implementation we take $\iota_{RW} = 10$.
- (2) In all the other steps, we take $p_n = p_{1,n}$.

Convergence of the ACMH sampler. If we choose $g_0(z)$ such that $g_0(z) \geq \beta_0 \pi(z)$ for some $0 < \beta_0 < 1$, then $q(z|x;\hat{\psi}(\mathcal{H}^{n-1})) \geq \delta \beta_0 = \beta$. By Corollary 1, we have a formal justification of the convergence of the ACMH sampler.

Corollary 3 Suppose that

$$g_0(z) \geq \beta_0 \pi(z) \text{ for all } z \in E, \tag{12}$$

for some $0 < \beta_0 < 1$. Then Theorems 1 and 2 hold for the ACMH sampler for any history \mathcal{H}^n .

For a general target $\pi(z)$, $g_0(z)$ can be informally selected such that it is sufficiently heavy-tailed to make (12) hold. In Bayesian inference, $\pi(z)$ is a posterior density that is proportional to $p(y|z)p(z)$ with $p(z)$ the prior and $p(y|z)$ the likelihood. Suppose that the likelihood is bounded; this is the case if the maximum likelihood estimator exists. If $p(z)$ is a proper density and we can generate from it, then we can set $g_0(z) = p(z)$ and it is straightforward to check that the condition (12) holds. The boundedness condition (12) is satisfied in all the examples in this paper.

We now briefly discuss the cost of running the two chain algorithm as in our article, compared to running a single chain adaptive algorithm. If the target density is inexpensive to evaluate, then the cost of running the two chain sampler is very similar to the cost of running just one chain because the major cost is incurred in updating the proposal distribution. If it is expensive to evaluate the target, then we can run the two chains in parallel on two (or more) processors. This is straightforward to do in programs such as Matlab because multiple processors are becoming increasingly common on modern computers.

Section 5 discusses the initial proposal q_0 , the history vector \mathcal{H}^0 and g_0 .

4.4.1 Two-stage adaptation

We run the adaptive sampling scheme in two stages. Adaptation in the first stage is carried out more intensively by re-estimating the mixture of t distributions after every 2000 iterations, and then every 4000 iterations in the second stage. When estimating the mixtures of t in the first stage, we let the Variational Approximation algorithm determine the number of components. While in the second stage, we fix the number of components at that number in the mixture obtained after the first stage. This makes the procedure faster and helps to stabilize the moves. In addition, it is likely that the number of components is unchanged in this second stage.

4.4.2 Selecting the control parameters

When the mixture of t approximation $g_M(z; \hat{\psi}(\mathcal{H}^{n-1}))$ becomes closer to the target, we expect the proposal $q(z|x; \hat{\psi}(\mathcal{H}^{n-1}))$ to be close to $g_M(z; \hat{\psi}(\mathcal{H}^{n-1}))$. We can do so by setting $\delta = \delta_n \rightarrow 1$ as n increases and setting a small value to β_0 . In our implementation we take $\beta_0 = 0.001$, $\gamma = 0.2$ and a sequence δ_n as follows. Let N be the length of the chain we wish to generate and suppose that $N = a_N b_N$. We set $\delta_{kb_N+j} = (k+1)/a_N$

for $k = 0, \dots, a_N - 1$ and $j = 1, \dots, b_N$. In our implementation we take $a_N = 10$. For the correlation parameter ρ , we simply select the probability measure $\lambda(\rho)$ as the Beta(1,1) distribution. These values were set after some experimentation. However, it is likely that we can further improve the efficiency of the ACMH sampler with a more careful (and possibly adaptive) choice of these control parameters.

5 Initial exploration

The purpose of the ACMH sampler is to deal with non-standard and multimodal target distributions. The sampler works more efficiently if the adaptive chain starts from an initial mixture distribution that is able to roughly locate the modes. We therefore attempt to initialize the history vector \mathcal{H}^0 by a few draws generated approximately from π by an algorithm that can explore efficiently the whole support of the target, and then estimate the initial mixture of t based on these draws. Our paper uses simulated annealing (Neal, 2001) to initialize the sampler. An alternative is to use the Wang-Landau algorithm (Wang and Landau, 2001a,b). However, this algorithm requires the user to partition the parameter space appropriately which is difficult to do in many applications.

Simulated annealing. Simulated annealing works by moving from an easily-generated distribution to the distribution of interest through a sequence of bridging distributions. Annealed sampling has proved useful in terms of efficiently exploring the support of the target distribution (Neal, 2001). Let $\pi_0(x)$ be some easily-generated distribution, such as a t distribution, and ψ_t , $t = 0, 1, \dots, T$ a sequence of real numbers such that $0 = \psi_0 < \dots < \psi_T = 1$. A convenient choice is $\psi_t = t/T$. Let

$$\eta_t(x) = \pi_0(x)^{1-\psi_t} \pi(x)^{\psi_t}.$$

Note that η_0 is the initial distribution π_0 and η_T is the target π . We sample from this sequence of distributions using the sequential Monte Carlo method (see, e.g. Del Moral et al., 2006; Chopin, 2004), as follows.

1. Generate $x_i \sim \eta_0(\cdot)$, $i = 1, \dots, N_p$, where N_p is the number of particles.
2. For $t = 1, \dots, T$
 - (i) Reweighting: compute the weights

$$\tilde{w}_i = \frac{\eta_t(x_i)}{\eta_{t-1}(x_i)}, \quad w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{N_p} \tilde{w}_j}.$$

- (ii) Resampling: sample from $(x_i, w_i)_{i=1, \dots, N_p}$ using stratified sampling. Let $(\tilde{x}_i)_{i=1, \dots, N_p}$ be the resampled particles.

- (iii) Markov move: for $i = 1, \dots, N_p$, generate $x_i^{(m)} \sim P_{\eta_t}(\cdot | x_i^{(m-1)})$, $m = 1, \dots, M$, where $P_{\eta_t}(\cdot | \cdot)$ is a Markov kernel with invariant distribution η_t and $x_i^{(0)} = \tilde{x}_i$. M is the burnin number.
- (iv) Set $x_i \leftarrow x_i^{(M)}$, $i = 1, \dots, N_p$.

The above sequential Monte Carlo algorithm produces particles x_i that are approximately generated from the target π (Del Moral et al., 2006). We can now initialize the history vector \mathcal{H}^0 using these particles and q_0 by the mixture of t estimated from \mathcal{H}^0 . Typically, T should take a large value for multimodal and high-dimensional targets. In the default setting of the ACMH sampler, we set $T=10$, $N_p=500$ and $M=10$. The initial distribution π_0 is a multivariate t distribution with location $\mu_0 = (0, \dots, 0)'$, scale matrix $\Sigma_0 = I_d$ and 3 degrees of freedom. However, it is useful to estimate μ_0 and Σ_0 from a short run of an adaptive random walk sampler, and we follow this approach in the real data examples.

In the default setting of the ACMH sampler, we select the heavy-tailed component $g_0(z)$ as $q_0(z)$ except that all the degrees of freedom of the t component of g_0 are set to 1, so that the boundedness condition (12) is likely to be satisfied. However, in all the examples below, g_0 is context-specified to make sure that (12) holds.

6 Simulations

A common performance measure for an MCMC sampler is the integrated autocorrelation time (IACT). For simplicity, consider first the univariate case and let $\{x_i, i = 1, \dots, M\}$ be the generated iterates from the Markov chain. Then the IACT is defined as

$$\text{IACT} = 1 + 2 \sum_{t=1}^{\infty} \rho_t,$$

where $\rho_t = \text{corr}(x_1, x_{t+1})$ is the autocorrelation of the chain at lag t . Provided that the chain has converged, the mean μ of the target distribution is estimated by $\bar{x} = \sum_i x_i / M$ whose variance is

$$\begin{aligned} \text{Var}(\bar{x}) &= \frac{\sigma^2}{M} \left(1 + 2 \sum_{t=1}^{M-1} \left(1 - \frac{t}{M} \right) \rho_t \right) \\ &\approx \frac{\sigma^2}{M} \left(1 + 2 \sum_{t=1}^{\infty} \rho_t \right) = \text{IACT} \cdot \frac{\sigma^2}{M}, \end{aligned}$$

where σ^2 is the variance of the target distribution. This shows that the IACT can be used as a measure of performance and that the smaller the IACT, the better the sampler. Following Pitt et al. (2012), we estimate the

IACT by

$$\widehat{\text{IACT}} = 1 + 2 \sum_{t=1}^{L^*} \hat{\rho}_t,$$

where $\hat{\rho}_t$ are the sample autocorrelations, and $L^* = \min\{1000, L\}$, with L the first index t such that $|\hat{\rho}_t| \leq 2/\sqrt{K_t}$ where K_t is the sample size used to estimate $\hat{\rho}_t$. That is, L is the lowest index after which the estimated autocorrelations are randomly scattered about 0. When $d > 1$ we take, for simplicity, the average IACT over the d coordinates, or the maximum IACT.

Another performance measure is the squared jumping distance, (see, e.g., Pasarica and Gelman, 2010, and the references in that paper). For the univariate case,

$$\text{Sq distance} = \frac{1}{N-1} \sum_{i=1}^N |x_{i+1} - x_i|^2 \approx 2\sigma^2(1 - \rho_1).$$

Therefore, the larger the squared distance the better. When $d > 1$, we take the average squared distance or the minimum squared distance over the d coordinates. We also report the acceptance rates in the examples below.

The IACT and squared distance are good performance measures when the target is unimodal. If the target is multimodal, these measures may not be able to determine whether or not the chain has converged to the target, as discussed below. We introduce another measure which suits the context of a simulation example where a test data set $\mathcal{D}_T = \{x_s = (x_{s1}, \dots, x_{sd})', s = 1, \dots, S\}$ generated from the target π is available. Let \hat{f}_i be the kernel density estimate of the i th marginal π_i of the target. The Kullback-Leibler divergence between π_i and \hat{f}_i is

$$\int \log \left(\frac{\pi_i(x_i)}{\hat{f}_i(x_i)} \right) \pi_i(x_i) dx_i \approx C_i - \text{LPDS}_i,$$

where $C_i = \int \pi_i(x_i) \log \pi_i(x_i) dx_i$ is independent of \hat{f}_i and

$$\text{LPDS}_i = \frac{1}{S} \sum_{s=1}^S \log \hat{f}_i(x_{si}) \approx \int \pi_i(x_i) \log \hat{f}_i(x_i) dx_i$$

is the log predictive density score for the i th marginal. Clearly, the bigger the LPDS_i , the closer the estimate \hat{f}_i to the true marginal π_i . We define the log predictive density score over the d marginals by

$$\text{LPDS} = \frac{1}{d} \sum_{i=1}^d \text{LPDS}_i.$$

The bigger the log predictive density score, the better the MCMC sampler.

6.1 Target distributions

The first target is a mixture of two multivariate skewed normal distributions

$$\pi_{\text{msn}}(x) = \sum_{k=1}^2 \varphi_k \mathcal{SN}_d(x; \mu_k, \Sigma_k, \lambda_k), \quad (13)$$

where $\mathcal{SN}_d(x; \mu, \Sigma, \lambda)$ denotes the density of a d -dimensional skewed normal distribution with location vector μ , scale matrix Σ and shape vector λ . See, e.g., Azzalini and Capitanio (1999), for an introduction to multivariate skewed normal distribution. We set $\mu_1 = (-5, \dots, -5)'$, $\mu_2 = (5, \dots, 5)'$, $\Sigma_1 = \Sigma_2 = 5(\sigma_{ij})_{i,j}$ with $\sigma_{ij} = (-0.5)^{|i-j|}$, $\lambda_1 = (-10, \dots, -10)'$, $\lambda_2 = (10, \dots, 10)'$ and $\varphi_1 = 0.6$, $\varphi_2 = 0.4$. It is straightforward to sample directly and exactly from a skewed normal (Azzalini and Capitanio, 1999), and therefore from $\pi_{\text{msn}}(x)$. However, this is a non-trivial problem for MCMC simulation, especially in higher dimensions, because the target is multimodal with an almost-zero probability valley between the two modes; see the left panel of Figure 1 for a plot of $\pi_{\text{msn}}(x)$ when $d=2$.

Let $f_k(x) = \mathcal{SN}_d(x; \mu_k, \Sigma_k, \lambda_k)$. By the properties of the multivariate skewed normal distribution (see, Azzalini and Capitanio, 1999), $f_k(x) \leq 2g_k(x)$, where $g_k(x) = N_d(x; \mu_k, \Sigma_k)$ is the density of the d -dimensional normal distribution with mean μ_k and covariance matrix Σ_k . The boundedness condition (12) is satisfied by setting $g_0(z) = \varphi_1 g_1(z) + \varphi_2 g_2(z)$, because then $\pi(z)/g_0(z)$ is bounded.

The second target density is the banana-shaped distribution considered in Haario et al. (1999)

$$\pi_b(x) = N_d(\phi_b(x); 0, \Sigma), \quad (14)$$

where $\Sigma = \text{diag}(100, 1, \dots, 1)$,

$$\phi_b(x) = (x_1, x_2 + bx_1^2 - 100b, x_3, \dots, x_d)$$

and $b=0.03$. See the right panel of Figure 1 for a plot of $\pi_b(x)$ when $d=2$. As shown, the banana-shaped density has a highly non-standard support with very long and narrow tails. It is challenging to sample from this target (Haario et al., 1999, 2001; Roberts and Rosenthal, 2009).

It can be shown after some algebra that the first marginal of $\pi_b(x)$ is $N(0, 10^2)$ and for $i = 3, \dots, d$ the marginals are independent $N(0, 1)$. It can be visually seen that the support of the second marginal is basically in the interval $(-50, 50)$. We therefore informally impose the condition (12) by selecting $g_0(z) = t_d(z; 0, \tilde{\Sigma}, 5)$, a multivariate t density with location 0, scale matrix $\tilde{\Sigma} = \text{diag}(100, 100, 1, \dots, 1)$ and 5 degrees of freedom. Typically, this ensures that the support of $g_0(z)$ covers the

support of $\pi_b(z)$ and therefore the boundedness condition (12) holds.

6.2 Performance of the ACMH sampler

This section reports the performance of the ACMH sampler and compares it to the adaptive random walk sampler (ARWMH) of Haario et al. (2001) and the adaptive independent Metropolis-Hastings sampler (AIMH) of Giordani and Kohn (2010). For all the samplers, we ran 50,000 iterations with another 50,000 for burnin.

6.2.1 The usefulness of the adaptive random walk step

We first demonstrate the importance of the adaptive random walk step by comparing the performance of the ACMH sampler to a variant of it that does not perform the random walk step. To make it easier to see the resulting estimates, we consider the target (13) with $d = 1$. The left panel in Figure 2 plots the kernel density estimates of the target estimated from the chains with and without the random walk step, as well as the true density. All the kernel density estimation reported in this paper is done using the built-in Matlab function `ksdensity` with the default setting. The right panel also plots the estimated kernel densities of the first marginal when sampling from the banana-shaped target (14). The first marginal of the banana-shaped target has very long tails (see Figure 1) and it is challenging for adaptive MCMC samplers to efficiently explore the extremes of these tails. The plots show that the chain with the random walk step explores the tail areas around the local modes more effectively.

We now formally justify the claim above using the censored likelihood scoring rule proposed in Diks et al. (2011). This scoring rule is a performance measure for assessing the predictive accuracy of a density estimator $\hat{f}(x)$ over a specific region of interest, which is the tail area in our problem. Let A denote the region of interest, \mathcal{D} a set of n observations. Then the censored likelihood score is defined as

$$S(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{x \in \mathcal{D}} \left(1_{x \in A} \log \hat{f}(x) + 1_{x \in A^c} \log \int_{A^c} \hat{f}(z) dz \right), \quad (15)$$

where A^c is the complement of set A . This scoring rule works similarly to the popular logarithmic scoring rule (Good, 1952); in particular the bigger $S(\hat{f}, \mathcal{D})$ is, the better the performance of \hat{f} . However the censored likelihood score takes into account the predictive accuracy

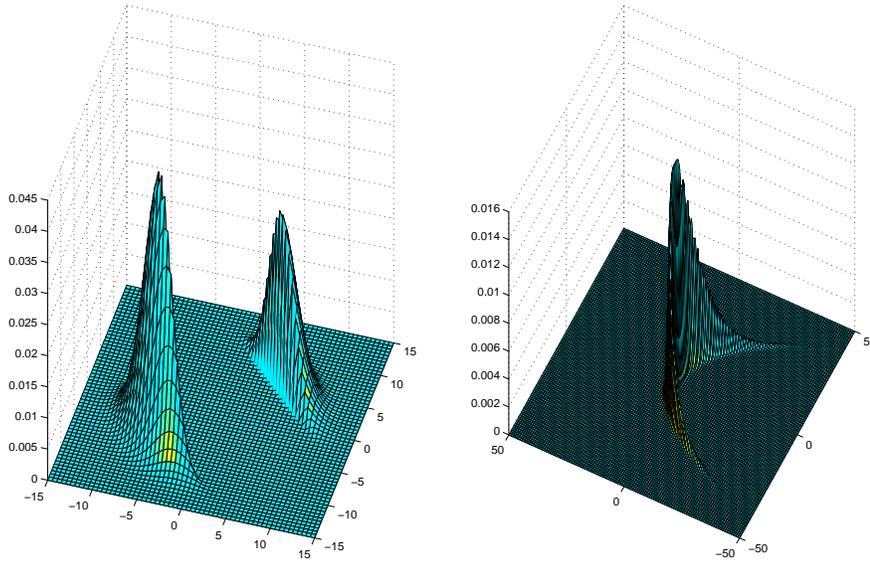


Fig. 1 Plots of the probability density functions of the mixture example (left) and the banana-shaped example (right) for $d=2$.

in a particular region of interest; see Diks et al. (2011) for a more detailed interpretation.

We consider the case of the mixture target $\pi_{\text{msn}}(x)$ with $d=1$ and are interested in how efficiently the ACMH samplers, with and without the random walk step, explore the left and right tails of $\pi_{\text{msn}}(x)$. Let $\hat{f}_1(x)$ and $\hat{f}_2(x)$ be the kernel densities estimated from the chains with and without the random walk step, respectively. We compute the score (15) for \hat{f}_1 and \hat{f}_2 based on $n=5000$ independent draws from the target $\pi_{\text{msn}}(x)$, in which the tail area is defined as $A=\{x \in \mathbb{R}: x < -15 \text{ or } x > 15\}$. We replicate the computation 10 times. The scores averaged over the replications with respect to the ACMH samplers with and without the random walk step are 0.98 and 0.96 respectively. This result formally justifies the claim that the random walk step helps the sampler to explore the tail area more effectively.

We also ran long chains with 200,000 iterations after discarding another 200,000 for burnin, then the difference between the censored likelihood scores of the ACMH samplers with and without the random walk step is 0.0008. That is, the difference decreases when the number of iterations increases. This result suggests that the ACMH sampler without the random walk step is able to explore the tail area effectively if it is run long enough.

6.2.2 The usefulness of the component-wise sampling step

To illustrate the effect of the component-wise sampling step, we sample from the banana-shaped target using the ACMH samplers with and without this step. The coordinates x_B that are kept unchanged, and therefore the size d_B , are selected randomly as in Section 4.2. Table 1 summarizes the performance measures for these two samplers averaged over 10 replications. The result shows that in general the sampler that performs the component-wise sampling step outperforms the one that does not.

6.2.3 The usefulness of the reversible Metropolis-Hastings step

We demonstrate the importance of the reversible step by comparing the ACMH sampler with a version of it in which the δ parameter in Section 4.4 is set to one, i.e. only the independent Metropolis-Hastings step is performed. Table 2 summarizes the performance measures for these two samplers averaged over 10 replications. The results suggest that the correlated step helps improve significantly on the performance of the ACMH sampler.

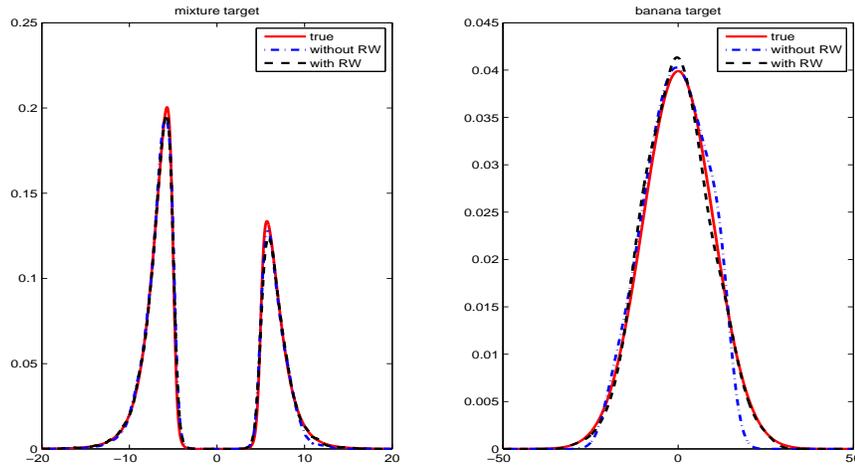


Fig. 2 The left panel plots the kernel density estimates of the mixture target obtained from the ACMH chains with and without the random walk step, as well as the true density. Similarly, the right panel plots the densities with respect to the first marginal of the banana-shaped target.

d	Algorithm	Acceptance rate (%)	IACT	Sq distance
10	Without component-wise sampling	38	38.61	2.63
	With component-wise sampling	56	19.45	5.92
20	Without component-wise sampling	32	57.11	0.71
	With component-wise sampling	34	47.33	1.52
40	Without component-wise sampling	14	171.4	0.18
	With component-wise sampling	26	80.17	0.83

Table 1 Importance of component-wise sampling.

d	Algorithm	Acceptance rate (%)	IACT	Sq distance
10	Without CMH	35	26.71	2.49
	With CMH	56	19.45	5.92
20	Without CMH	26	62.11	0.88
	With CMH	34	47.33	1.52
40	Without CMH	0.1	180.2	0.01
	With CMH	26	80.17	0.83

Table 2 Importance of the correlated Metropolis-Hastings step.

6.2.4 Comparison of the ACMH sampler to other adaptive samplers

We now compare the ACMH sampler to the ARWMH and AIMH samplers for the mixture of skewed normals and banana-shaped targets.

The mixture target. Figure 3 plots the chains with respect to the first marginal for three cases: $d=2$, $d=5$ and $d=10$. The ARWMH never converges to the target even when $d=1$. It looks as if the ARWMH has converged but in fact it is always stuck at a local mode because the target has two modes that are almost separate. In such cases, the performance of ARWMH may be mistakenly considered to be good in terms of IACT and squared jumping distance, while its log predictive

density score will be large because the estimated density is far from the true density. This justifies the introduction of the log predictive density score as a performance measure. The AIMH sampler works well when d is as small as 3 in this hard example. As expected with samplers based on independent Metropolis-Hastings steps only, it is almost impossible for the AIMH to move the chain when d is large. Figure 3 shows that the ACMH sampler converges best.

We now compare the performance of the three adaptive samplers more formally over 5 replications. The acceptance rates, IACT, squared distance values and log predictive density scores are computed and averaged over the d marginals and 5 replications. Table 3 summarizes the result. The ACMH sampler compares

d	Algorithm	Acceptance rate (%)	IACT	Sq distance	LPDS
2	ARWMH	36	9.15	1.73	-73.1
	AIMH	14	13.6	10.5	-2.81
	ACMH	70	3.97	37.4	-2.80
5	ARWMH	32	15.2	1.13	-42.5
	AIMH	11	22.2	6.12	-2.89
	ACMH	68	3.91	34.3	-2.84
10	ARWMH	29	29.5	0.63	-43.3
	AIMH	0.01	1885	0.006	-20.8
	ACMH	68	9.40	40.27	-2.86

Table 3 Mixture of skewed normals target: The table reports the acceptance rates, autocorrelation times, squared distances and log predictive density scores for the three adaptive sampling schemes ARWMH, AIMH, and ACMH and three values of the dimension d . The values are averaged over 5 replications.

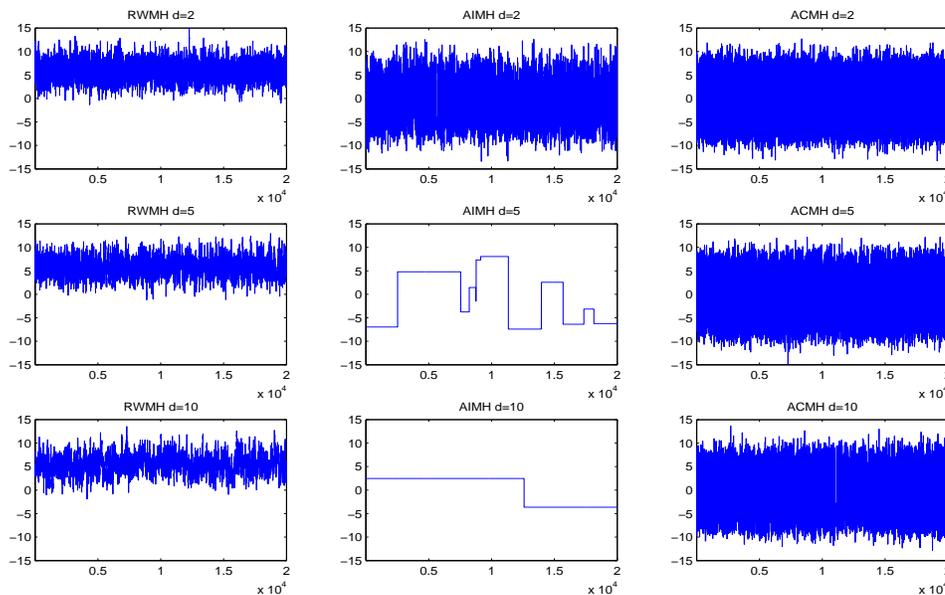


Fig. 3 Mixture of skewed normals target: Plots of the iterates of the three adaptive sampling schemes for the first marginal. Columns 1 to 3 correspond to the ARWMH, AIMH and ACMH sampling schemes. Rows 1 to 3 correspond to the dimensions $d=2$, 5 and 10.

favorably with the other two samplers. It looks as if the ARWMH chain has converged and performs well in terms of IACT and squared distance, but in fact the chain is trapped in a local mode. We conclude from these results that if the target is multimodal, performance measures based on IACT and squared distance may be misleading.

Banana-shaped target. Because this target is unimodal, the IACT and squared distance can be used as performance measures. Note that we do not use the log predictive score in this example because it is difficult to obtain an independent test data set that is generated exactly from the target (14). Table 4 summarizes the results in terms of percent acceptance rates, IACT, Sq distance and CPU time in seconds, where the CPU time is the average CPU time over the ten

replications. The results are based on 50000 iterations with another 50000 iterations used for burn-in. The table also reports two other performance measures that are used by Schmidl et al. (2013). The first is $\Pi/\text{time} = \text{Number of iterates}/(\text{IACT} \times \text{CPU time})$, which is an estimate of the number of independent iterates per unit time, which in this case is seconds. The second measure $\text{Acc}/\text{IACT} = 1000 \times \text{Acceptance rate}/\text{IACT}$. The table shows that the ACMH sampler outperforms the other two in terms of Acceptance rate, IACT, Sq distance, Acc/IACT . However, it is worse than the other two samplers in terms of Π/time as it takes longer to run. The code was written in Matlab and run on an Intel Core i7 3.2GHz desktop running on a single processor. The time taken by the ACMH sampler can be reduced appreciably by taking the following steps. (i) First, by

profiling the code we find that a major part of the time to run the sampler is taken by the variational approximation procedure used to obtain the mixture of t proposal. The running time can be shortened appreciably by write the variational approximation part of the code in C or Fortran and using mex files. (ii) Second, the running time can also be shortened by running the two chains on separate processors in parallel and also by using parallel processing for the independent draws.

7 Applications

7.1 Covariance matrix estimation for financial data

This section applies the ACMH sampler to estimate the covariance matrix of ten monthly U.S. industry portfolios returns. The data is taken from the Ken French data library and consists of $N=990$ observations $\mathbf{y} = \{y_i, i = 1, \dots, N\}$ from July 1925 to December 2008. We use the following 10 industry portfolios: consumer non-durable, consumer durable, manufacturing, energy, business equipment, telephone and television transmission, shops, health, utilities, and others.

We assume that the $y_i, i=1, \dots, N$, are independently distributed as $N_p(y_i; 0, \Sigma)$ with $p=10$. We are interested in Bayesian inference of Σ . Yang and Berger (1994) propose the following reference prior for Σ ,

$$p_{\text{ref}}(\Sigma) \propto \frac{1}{|\Sigma| \prod_{i < j} (r_i - r_j)},$$

where $r_1 \geq r_2 \geq \dots \geq r_p$ are the eigenvalues of Σ . This reference prior puts more mass on covariance matrices having eigenvalues that are close to each other. That is, it shrinks the eigenvalues in order to produce a covariance matrix estimator with a better *condition number* defined as the ratio between the largest and smallest eigenvalues (see, e.g., Belsley et al., 1980). In order to formally impose the boundedness condition (12), we modify the reference prior and use

$$p_{\text{ref}, \epsilon}(\Sigma) \propto \frac{1}{|\Sigma| \prod_{i < j} (r_i - r_j)} 1_{A_\epsilon}(\Sigma),$$

where A_ϵ is the set of symmetric and positive definite matrices Σ such that $\min_{i,j} |r_i - r_j| > \epsilon$. We take $\epsilon = 10^{-6}$ in the implementation. Then, the posterior distribution of Σ is

$$p_\epsilon(\Sigma | \mathbf{y}) \propto \frac{\exp \left\{ -\frac{1}{2} \text{trace}(\Sigma^{-1} S) \right\}}{|\Sigma|^{N/2+1} \prod_{i < j} (r_i - r_j)} 1_{A_\epsilon}(\Sigma),$$

where $S = \sum_{i=1}^N (y_i - \bar{y})(y_i - \bar{y})'$ with \bar{y} the sample mean. To impose the boundedness condition, we select $g_0(\Sigma)$

to be the inverse Wishart distribution with scale matrix S and degrees of freedom $N-p+1$,

$$g_0(\Sigma) \propto \frac{1}{|\Sigma|^{N/2+1}} \exp \left\{ -\frac{1}{2} \text{trace}(\Sigma^{-1} S) \right\}.$$

It is then straightforward to check that there exists a constant β_0 such that $g_0(\Sigma) \geq \beta_0 p_\epsilon(\Sigma | \mathbf{y})$.

Because of the positive-definite constraint on Σ , it is useful to transform from the space of positive-definite matrices to an unconstrained space using the one-to-one transformation $\Sigma^* = \log(\Sigma)$ or $\Sigma = \exp(\Sigma^*)$, where Σ^* is a symmetric matrix (Leonard and Hsu, 1992). We can generate Σ by generating the unconstrained lower triangle of Σ^* . Let $\Sigma^* = QR^*Q'$ where $R^* = \text{diag}(r_1^*, \dots, r_p^*)$ with $r_1^* \geq r_2^* \geq \dots \geq r_p^*$ and Q the orthogonal matrix. Then $\Sigma = Q \text{diag}(e^{r_1^*}, \dots, e^{r_p^*}) Q'$. We are now working on an unconstrained space of Σ^* , we therefore can fit multivariate mixtures of t to the iterates. The dimension of the parameter space is $d = p(p+1)/2 = 55$. Note that, in order to be able to generate Σ^* from $g_0(\Sigma)$, we first generate Σ from $g_0(\Sigma)$ and then transform it to Σ^* as follows. Let $\Sigma = Q\Lambda Q'$ where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_p)$ with $\lambda_1 \geq \dots \geq \lambda_p > 0$, then $\Sigma^* = Q \text{diag}(\log(\lambda_1), \dots, \log(\lambda_p)) Q'$.

Each sampler was run for 500,000 iterations after discarding the first 500,000 burnin iterations. To reduce the computing time for the ACMH sampler in such a long run, we stop updating the mixture of t distribution after the burnin period. Figure 4 plots the iterates from the three chains for the first and second marginals as well as the first 500 autocorrelations of these iterates. For the ARWMH sampler, mixing is very poor and the chain moves very slowly. The AIMH and ACMH samplers mix better. Table 5 summarizes the results for the three samplers, and reports the acceptance rates, the IACT values and squared distances (both averaged over the marginals), the maximum IACT (max IACT) and minimum squared distance (min Sq distance) (among the 55 marginal chains), and the CPU times taken by each sampler. It also reports $\text{II}/\text{time} = \text{Number of iterations} / (\text{IACT} \times \text{CPU time})$, $\text{min II}/\text{time} = \text{Number of iterations} / (\text{max IACT} \times \text{CPU time})$, $\text{Acc}/\text{IACT} = \text{Acceptance rate} / \text{IACT}$ and $\text{min Acc}/\text{IACT} = \text{Acceptance rate} / (\text{max IACT})$. In this example the ACMH sampler outperforms the other two samplers on almost all the performance measures.

7.2 Spam filtering

Automatic spam filtering is an important function for any email service provider. The researchers at the Hewlett-Packard Labs created a spam email data set consisting of 4061 messages, each of which has been already

d	Algorithm	Acceptance rate (%)	IACT	Sq distance	CPU time	II/time	Acc/IACT
5	ARWMH	14	81.83	1.23	21	29.1	171
	AIMH	20	44.52	5.80	14	80.2	449
	ACMH	64	24.33	17.0	253	8.12	2631
10	ARWMH	15	150.1	0.41	22	15.1	100
	AIMH	31	49.65	3.06	15	67.1	624
	ACMH	56	19.45	5.92	250	10.3	2879
20	ARWMH	18	168.8	0.15	26	11.4	107
	AIMH	10	174.6	0.33	16	17.9	57
	ACMH	34	47.33	1.52	368	2.87	718
40	ARWMH	24	208	0.05	40	6.01	115
	AIMH	0	1991	0	16	1.57	0
	ACMH	26	80.17	0.83	395	1.58	324

Table 4 Banana-shaped example: performance measures averaged over 10 replications. II/sec = Number of iterates/(IACT \times CPU time) and Acc/IACT = 1000 \times Acceptance rate/IACT.

	Covariance Example			Spam Example		
	ARWMH	AIMH	ACMH	ARWMH	AIMH	ACMH
Acceptance rate (%)	7.7	27	30	20	1.2	14
Avg IACT	476	148	28	288	333	156
Max IACT	574	283	42	484	411	344
CPU Time (mins)	21.9	13.7	36.5	10.9	5.9	19.2
Avg Sq Dist ($\times 10^4$)	0.2	12	17	2121	698	7379
Min Sq Dist ($\times 10^4$)	0.07	8.6	12	1.5	1.1	13
Avg II/time	48	246	489	64	102	67
Min II/time	40	129	326	40	82	30
Avg Acc/IACT (times 10^3)	16	182	1071	69	4	90
Min Acc/IACT (times 10^3)	13	95	714	41	3	41

Table 5 Real data examples. Average II/time = Number of iterations/ (Average IACT \times CPU time), Min II/time = Number of iterations/ (Max IACT \times CPU time), Avg Acc/IACT = 1000 \times Acceptance rate / (Avg IACT), and Min Acc/IACT = 1000 \times Acceptance rate / (Max IACT).

been classified as an email or a spam together with 57 attributes (predictors) which are relative frequencies of commonly occurring words. The goal is to design a spam filter that can filter out spam before clogging the user’s mailbox. The data set is available at www-stat.stanford.edu/~tibs/ElemStatLearn.

A suitable statistical model for this goal is logistic regression, where the probability of being a spam, given the predictor vector x , is modeled as

$$\mu(x, \theta) = P(y = 1|x, \beta_0, \beta) = \frac{\exp(\beta_0 + x'\beta)}{1 + \exp(\beta_0 + x'\beta)},$$

with $\theta = (\beta_0, \beta)'$ the coefficient vector. For a future message with attributes x , using Bayesian inference, our goal is to estimate the posterior probability that the message is classified as a spam,

$$\mu(x) = \mathbb{E}_{\theta|D}(\mu(x, \theta)) = \int \mu(x, \theta)p(\theta|D)d\theta, \quad (16)$$

where $p(\theta|D)$ denotes the posterior distribution of θ given a training data set D .

We employ the weakly informative prior for θ proposed in Gelman et al. (2008). The prior is constructed

by first standardizing the predictors to have mean zero and standard deviation 0.5, and then putting independent Cauchy distributions on the coefficients. As a default choice, Gelman et al. (2008) recommended a central Cauchy distribution with scale 10 for the intercept β_0 and central Cauchy distributions with scale 2.5 for the other coefficients. This is a regularization prior and Gelman et al. (2008) argue that it has many advantages; in particular, it works automatically without the need to elicit hyperparameters. We set g_0 to be the prior, which ensures that the boundedness condition (12) is satisfied because the maximum likelihood estimator for logistic regression exists.

We first use the whole data set and run each of the three samplers for 200,000 iterations after discarding 200,000 burnin iterations. The dimension in this example is $d = 58$. Figure 5 plots the iterates from the three chains for the first and second marginals as well as the first 500 autocorrelations of these iterates. As in the covariance estimation example, Table 5 summarizes the results for the three samplers. The ACMH sampler outperforms the other two samplers except for the Avg II/time and Min II/time, where the results are mixed,

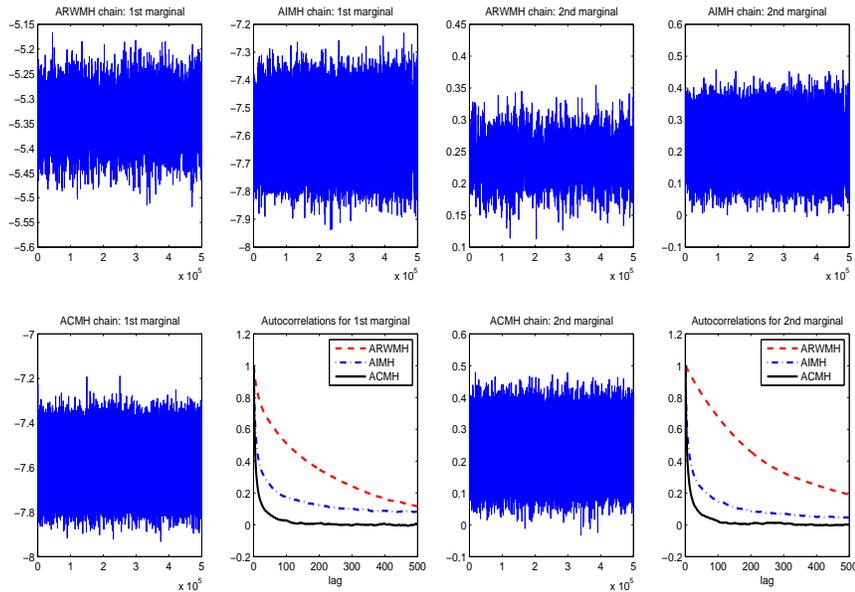


Fig. 4 Covariance example: Plots of the iterates of the 1st and 2nd marginals and the first 500 autocorrelations of these iterates for the three adaptive samplers.

because of the longer running times. However, as noted at the end of Section 6.2.4, it is straightforward to make the ACMH run faster.

We also consider the predictive ability of the binary models estimated by the three chains. The continuous ranked probability score (CRPS) is a widely used prediction measure in the forecasting community, see, e.g., Gneiting and Raftery (2007) and Hersbach (2000). Let F be the cumulative distribution function (cdf) of the predictive distribution in use and y be an actual observation. The CRPS is defined as

$$\text{CRPS}(F|y) = \int_{\mathbb{R}} (F(z) - \mathbb{1}_{z \geq y})^2 dz.$$

When F_{μ} is the cdf of a Bernoulli variable Y with probability of success $\mu = P(Y=1)$, the CRPS is given by

$$\text{CRPS}(F_{\mu}|y=0) = \mu^2 \text{ and } \text{CRPS}(F_{\mu}|y=1) = (1-\mu)^2$$

with $\mu(x)$ given in (16). Let D^T be a test data set, we compute the CRPS (based on D^T) by

$$\text{CRPS} = \sum_{(x,y) \in D^T} \text{CRPS}(F_{\mu(x)}, y). \quad (17)$$

Under this formulation, it is understood that smaller CRPS means better predictive performance.

We now randomly partition the full data set into two roughly equal parts: one is used as the training set D and the other as the test set D^T . We would like to assess the performance of the samplers in terms of pre-

dictive accuracy. To do so, we compute the CRPS for each sampler as in (17). To take into account the randomness of the partition, we average the CRPS values over 5 such random partitions. We first run each sampler for 50,000 iterations with another 50,000 burnin iterations. The averaged CRPS values for the ARWMH, AIMH and ACMH are 135.23, 135.10, 133.14 respectively. This result suggests that the ACMH has the best predictive accuracy for that number of iterations. We then carried out longer runs for each sampler with 500,000 iterations after discarding 500,000 burnin iterations. The averaged CRPS values for the ARWMH, AIMH and ACMH are now 123, 123.1, 122.8 respectively. This means that all the samplers converge to the target if they are run long enough.

8 Discussion

This article develops a general-purpose adaptive correlated Metropolis-Hastings sampler that will work well for multimodal as well as unimodal targets. Its main features are the use of reversible proposals, the absence of a requirement for diminishing adaptation, and having as its major component the mixture of t model fitted by Variational Approximation. The ACMH sampler combines exploratory and exploitative stages and consists of various steps including correlated, random walk and Metropolis within Gibbs component-wise sampling steps. This makes the sampler explore the target more

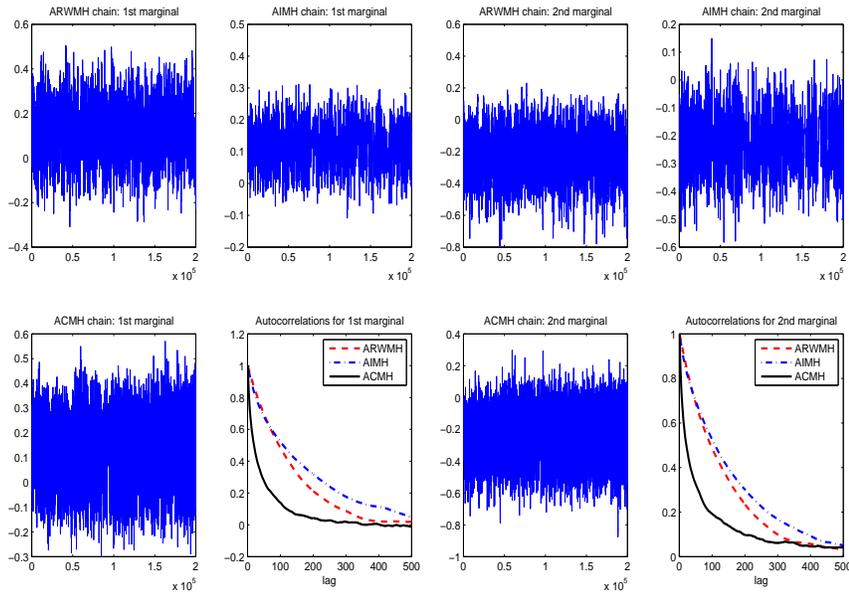


Fig. 5 Spam email example: Plots of the iterations of the 1st and 2nd marginals and the first 500 autocorrelations of these iterations for the three adaptive samplers.

effectively both globally and locally, and improves the acceptance rate in high dimensional problems. The convergence to the target is theoretically guaranteed without the need to impose Diminishing Adaptation.

There are two important and immediate extensions of our work. First, the ACMH sampler can be extended in a straightforward way to allow for a reversible proposal whose invariant distribution is a mixture of t that is constructed in a different way to the Variational Approximation approach, e.g. as in Hoogerheide et al. (2012). More generally, the ACMH sampler can be extended to a reversible proposal whose invariant distribution is a more general mixture, e.g. a mixture of multivariate betas or a mixture of Wishart densities, or a mixture of copula densities. Second, the current article considers applications where the likelihood can be evaluated explicitly (up to a normalizing constant). However, our methods apply equally well to problems where the likelihood can only be estimated unbiasedly and adaptive MCMC is carried out using the unbiased estimate instead of the likelihood as in, for example, Andrieu et al. (2010) and Pitt et al. (2012).

Acknowledgment

The research of Minh-Ngoc Tran and Robert Kohn was partially supported by Australian Research Council grant DP0667069.

References

- Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Series B*, 72(2):1–33.
- Andrieu, C. and Moulines, E. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *Annals of Applied Probability*, 16(3):1462–1505.
- Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373.
- Athreya, K. B. and Ney, P. (1978). A new approach to the limit theory of recurrent Markov chains. *Transactions of the American Mathematical Society*, 245:493–501.
- Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew-normal distribution. *J. Roy. Statist. Soc., series B*, 61(3):579–602.
- Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). *Regression Diagnostics, Identifying Influential Data and Sources of Collinearity*. John Wiley, New York.
- Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.
- Corduneanu, A. and Bishop, C. (2001). Variational Bayesian model selection for mixture distributions. In Jaakkola, T. and Richardson, T., editors, *Artificial Intelligence and Statistics*, volume 14, pages 27–34. Morgan Kaufmann.
- Craiu, R. V., Rosenthal, J., and Yang, C. (2009). Learn from thy neighbor: Parallel-chain and regional adap-

- tive MCMC. *Journal of the American Statistical Association*, 104:1454–1466.
- de Freitas, N., Højten-Sørensen, P., Jordan, M., and Russell, S. (2001). Variational MCMC. In Breese, J. and Kollar, D., editors, *Uncertainty in Artificial Intelligence (UAI), Proceedings of the Seventeenth Conference*, pages 120–127.
- Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68:411–436.
- Diks, C., Panchenko, V., and Dijk, D. (2011). Likelihood-based scoring rules for comparing density forecasts in tails. *Journal of Econometrics*, 163:215–230.
- Gelman, A., Jakulin, A., Grazia, P., and Su, Y.-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics*, 2:1360–1383.
- Geyer, C. J. and Thompson, E. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Amer. Statist. Assoc.*, 90:909V920.
- Giordani, P. and Kohn, R. (2010). Adaptive independent Metropolis-Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics*, 19(2):243–259.
- Giordani, P., Mun, X., Tran, M.-N., and Kohn, R. (2012). Flexible multivariate density estimation with marginal adaptation. *Journal of Computational and Graphical Statistics*. DOI:10.1080/10618600.2012.672784.
- Gneiting, T. and Raftery, A. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378.
- Good, I. (1952). Rational decisions. *Journal of the Royal Statistical Society, Series B*, 14:107–114.
- Grimmett, G. and Stirzaker, D. (2001). *Probability and Random Processes*. Oxford University Press, New York, 3 edition.
- Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14:375–395.
- Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7:223–242.
- Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15:559–570.
- Holden, L., Hauge, R., and Holden, M. (2009). Adaptive independent Metropolis-Hastings. *The Annals of Applied Probability*, 19(1):395–413.
- Hoogerheide, L., Opschoor, A., and van Dijk, H. K. (2012). A class of adaptive importance sampling weighted EM algorithms for efficient and robust posterior and predictive simulation. *Journal of Econometrics*, 171(2):101–120.
- Johnson, A. A., Jones, G. L., and Neath, R. C. (2011). Component-wise Markov chain Monte Carlo: Uniform and geometric ergodicity under mixing and composition. *Technical report*. <http://arxiv.org/pdf/0903.0664.pdf>.
- Leonard, T. and Hsu, J. S. J. (1992). Bayesian inference for a covariance matrix. *The Annals of Statistics*, 20(4):1669–1696.
- McGrory, C. A. and Titterton (2007). Variational approximations in Bayesian model selection for finite mixture distributions. *Computational Statistics and Data Analysis*, 51:5352–5367.
- Neal, R. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139.
- Nott, D. J., Tran, M.-N., and Leng, C. (2012). Variational approximation for heteroscedastic linear models and matching pursuit algorithms. *Statistics and Computing*, 22(2):497–512.
- Pasarica, C. and Gelman, A. (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, 20:343–364.
- Pitt, M. K., Silva, R. S., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 172(2):134–151.
- Pitt, M. K. and Walker, S. G. (2006). Extended constructions of stationary autoregressive processes. *Statistics & Probability Letters*, 76(12):1219–1224.
- Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov Chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2):458–475.
- Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367.
- Schmidl, D., Czado, C., Hug, S., and Theis, F. J. (2013). A vine-copula based adaptive mcmc sampler for efficient inference of dynamical systems. *Bayesian Analysis Journal*, 8(1):1–22.
- Schmidler, S. (2011). Exploration vs. exploitation in adaptive MCMC. Technical report, AdapSki Invited Presentation.
- Tran, M.-N., Giordani, P., Mun, X., Kohn, R., and Pitt, M. K. (2013). Copula-type estimators for flexible multivariate density modeling using mixtures. *Journal of Computational and Graphical Statistics*. To appear.
- Wang, F. and Landau, D. (2001a). Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram.

Physical Review E, 64(5):56101.

Wang, F. and Landau, D. (2001b). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053.

Yang, R. and Berger, J. O. (1994). Estimation of a covariance matrix using the reference prior. *The Annals of Statistics*, 22(3):1195–1211.

Appendix

Proof (Proof of Theorem 1) The proof is based on the split chain construction of Athreya and Ney (1978). We can write

$$p_i(x_{i-1}, dx_i) = \beta \Pi(dx_i) + (1 - \beta) \nu_i(x_{i-1}, dx_i),$$

where, by (1),

$$\nu_i(x_{i-1}, dx_i) = \frac{p_i(x_{i-1}, dx_i) - \beta \Pi(dx_i)}{1 - \beta}$$

is a transition distribution with invariant distribution $\Pi(\cdot)$. By induction, it is easy to show that

$$\begin{aligned} \mathbb{P}^n(x_0, dx_n) &= \int \mathbb{P}^{n-1}(x_0, dx_{n-1}) p_n(x_{n-1}, dx_n) \\ &= (1 - \beta)^n \nu^n(x_0, dx_n) + (1 - (1 - \beta)^n) \Pi(dx_n), \end{aligned}$$

for $n \geq 1$, where

$$\nu^n(x_0, dx_n) = \int_{x_1} \cdots \int_{x_{n-1}} \nu_1(x_0, dx_1) \cdots \nu_n(x_{n-1}, dx_n).$$

So

$$\mathbb{P}^n(x_0, dx_n) - \Pi(dx_n) = (1 - \beta)^n (\nu^n(x_0, dx_n) - \Pi(dx_n)). \quad (18)$$

This implies that for any set $A \in \mathcal{E}$,

$$\begin{aligned} |\mathbb{P}^n(x_0, A) - \Pi(A)| &= \left| \int_A (\mathbb{P}^n(x_0, dx_n) - \Pi(dx_n)) \right| \\ &\leq 2(1 - \beta)^n. \end{aligned}$$

It follows that

$$\|\mathbb{P}^n(x_0, \cdot) - \Pi(\cdot)\|_{TV} \leq 2(1 - \beta)^n,$$

for any initial x_0 .

We introduce some notation. Denote by $\mu_0(\cdot)$ the distribution of the initial x_0 . For $j > i$, define,

$$\mathbb{P}^{j|i}(x_i, dx_j) = \int_{x_{i+1}} \cdots \int_{x_{j-1}} p_{i+1}(x_i, dx_{i+1}) \cdots p_j(x_{j-1}, dx_j).$$

$$\mathbb{E}_{\mu_0 \mathbb{P}^i}(h) = \mu_0 \mathbb{P}^i(h) = \int_{x_0} \int_{x_i} \mu_0(dx_0) \mathbb{P}^i(x_0, dx_i) h(x_i).$$

$$\begin{aligned} \mathbb{E}_{\mu_0 \mathbb{P}^i \mathbb{P}^{j|i}}(h \circ h) &= \mu_0 \mathbb{P}^i \mathbb{P}^{j|i}(h \circ h) \\ &= \int_{x_0} \int_{x_i} \int_{x_j} \mu_0(dx_0) \mathbb{P}^i(x_0, dx_i) \mathbb{P}^{j|i}(x_i, dx_j) h(x_i) h(x_j). \end{aligned}$$

$$\mathbb{E}_{\Pi}(h) = \Pi(h) = \int_x \Pi(dx) h(x).$$

Lemma 8 *Suppose that the assumptions in Theorems 1 and 2 hold. Then,*

(i)

$$\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mu_0 \mathbb{P}^i}(h) = \mathbb{E}_{\Pi}(h) + O\left(\frac{1}{n}\right).$$

(ii)

$$\frac{1}{n^2} \sum_{i=1}^n \mathbb{E}_{\mu_0 \mathbb{P}^i}(h^2) = \mathbb{E}_{\Pi}(h^2) + O\left(\frac{1}{n}\right).$$

(iii)

$$\frac{1}{n^2} \sum_{j=1}^n \sum_{i=1}^n \mathbb{E}_{\mu_0 \mathbb{P}^i}(h) \mathbb{E}_{\mu_0 \mathbb{P}^j}(h) = \mathbb{E}_{\Pi}(h)^2 + O\left(\frac{1}{n}\right).$$

(iv)

$$\frac{1}{n^2} \sum_{j=2}^n \sum_{i=1}^{j-1} \mathbb{E}_{\mu_0 \mathbb{P}^i \mathbb{P}^{j|i}}(h \circ h) = \mathbb{E}_{\Pi}(h)^2 + O\left(\frac{1}{n}\right).$$

Proof (Proof of Lemma 8) Without loss of generality we take $h \geq 0$, because we can consider the positive and negative parts of h separately. Let h_{\max} be the maximum value of h . To obtain Part (i),

$$\begin{aligned} &\left| \int (P^i(x_0, dx) - \Pi(dx)) h(x) \right| = \\ &(1 - \beta)^i \left| \int (\nu^i(x_0, dx) - \Pi(dx)) h(x) \right| \\ &\leq 2(1 - \beta)^i h_{\max} \frac{1}{n} \sum_{i=1}^n \left| \int (P^i(x_0, dx) - \Pi(dx)) h(x) \right| \\ &\leq \frac{2h_{\max}}{n} \sum_{i=1}^n (1 - \beta)^i \\ &= O\left(\frac{1}{n}\right). \end{aligned}$$

Part (ii) is obtained similarly. To obtain Part (iii),

$$\begin{aligned} \mu_0 P^i(h) \mu_0 P^j(h) - \Pi(h)^2 &= (\mu_0 P^i(h) - \Pi(h)) \times \\ &(\mu_0 P^j(h) - \Pi(h)) + \Pi(h) (\mu_0 P^j(h) - \Pi(h)) \\ &+ (\mu_0 P^i(h) - \Pi(h)) \Pi(h) \end{aligned}$$

and the result follows from Part (i). Part (iv) is obtained similarly to Part (iii).

Proof (Proof of Theorem 2) By (ii)-(iv) of Lemma 8,

$$\begin{aligned} \text{Var}\left(\frac{S_n}{n}\right) &= \frac{2}{n^2} \sum_{j=2}^n \sum_{i=1}^{j-1} (\mathbb{E}_{\mu_0 \mathbb{P}^i \mathbb{P}^j | i}(h \circ h) - \mathbb{E}_{\Pi}(h)^2) \\ &+ \frac{1}{n^2} \sum_{i=1}^n (\mathbb{E}_{\mu_0 \mathbb{P}^i}(h^2) - \mathbb{E}_{\Pi}(h)^2) \\ &+ \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (\mathbb{E}_{\Pi}(h)^2 - \mathbb{E}_{\mu_0 \mathbb{P}^i}(h) \mathbb{E}_{\mu_0 \mathbb{P}^j}(h)) \\ &= O\left(\frac{1}{n}\right). \end{aligned}$$

The rest of the proof is similar to that in p. 326 of Grimmett and Stirzaker (2001). (iii)

Proof (Proof of Corollary 1) (i) Note that the acceptance probability at the i th iterate of the Metropolis-Hastings algorithm is

$$\alpha_i(z, x) = \left(1, \frac{\pi(z) q_i(x|z)}{\pi(x) q_i(z|x)}\right),$$

and the Markov transition distribution is

$$\begin{aligned} p_i(x_{i-1}, dx_i) &= \alpha_i(x_i, x_{i-1}) q_i(x_i | x_{i-1}) dx_i + \\ &\delta_{x_{i-1}}(dx_i) \left(1 - \int \alpha_i(z, x_{i-1}) q_i(z | x_{i-1}) dz\right) \end{aligned}$$

From $q_i(z|x) \geq \beta\pi(z)$ for all x, z , we can show that

$$\alpha_i(z, x) q_i(z|x) \geq \beta\pi(z),$$

which implies that

$$p_i(x_{i-1}, dx_i) \geq \beta\Pi(dx_i).$$

Therefore, the results in Theorems 1 and 2 follow. Proof of (ii) is straightforward. To prove (iii), note that

$$p_{1,i}(x_{i-1}, dx_i) \geq \beta\Pi(dx_i),$$

therefore

$$p_i(x_{i-1}, dx_i) \geq \omega\beta\Pi(dx_i),$$

which implies the results in Theorems 1 and 2. To prove (iv), note that

$$\begin{aligned} p_i(x_{i-1}, dx_i) &= p_{1,i} p_{2,i}(x_{i-1}, dx_i) \\ &\geq \int_z \beta\Pi(dz) p_{2,i}(z, dx_i) = \beta\Pi(dx_i). \end{aligned}$$

Also,

$$\begin{aligned} p_i(x_{i-1}, dx_i) &= p_{2,i} p_{1,i}(x_{i-1}, dx_i) \\ &\geq \int_z p_{2,i}(x_{i-1}, dz) \beta\Pi(dx_i) = \beta\Pi(dx_i). \end{aligned}$$

Theorems 1 and 2 then follow. Part (v) is obtained similarly to Part (iv).

Proof (Proof of Lemma 1)

$$(i) \quad \zeta(x) T(z|x) = \zeta(x) \zeta(z) = \zeta(z) T(x|z).$$

(ii)

$$\begin{aligned} \zeta(x) T(z|x) &= \zeta(x) \zeta(z_{A, z_B}) / \zeta(z_B) I(z_B = x_B) \\ &= I(z_B = x_B) \zeta(x) \zeta(z) / \zeta(z_B) \\ &= \zeta(z) T(x|z). \end{aligned}$$

(iii)

$$\begin{aligned} \zeta(x) T(z|x) &= \int \zeta(x) T(z|x; \rho) \lambda(d\rho) \\ &= \int \zeta(z) T(x|z; \rho) \lambda(d\rho) \\ &= \zeta(z) T(x|z). \end{aligned}$$

Proof (Proof of Lemma 2)

(i)

$$\begin{aligned} \zeta(x) T(z|x) &= \sum_{k=1}^G \omega_k \zeta_k(x) T_k(z|x) \\ &= \sum_{k=1}^G \omega_k \zeta_k(z) T_k(x|z) = \zeta(z) T(x|z). \end{aligned}$$

$$(ii) \quad \omega(k|x) = \omega_k \zeta_k(x) / \zeta(x) = \omega_k.$$

(iii) This follows from part (ii) and Lemma 1(i).

Proof (Proof of Corollary 2) The proof follows from part (ii) of Lemma 1 and Lemma 2.

Proof (Proof of Lemma 3) The proof follows because $T(z|x)/T(x|z) = \zeta(z)/\zeta(x)$.

Proof (Proof of Lemma 4) To prove Part (i), recall that $\phi_d(z; a, B)$ denotes a d -variate Gaussian density with mean vector a and covariance matrix B and let $\text{IG}(\lambda; \alpha, \beta)$ denote an inverse gamma density with shape parameter α and scale parameter β . We first consider

the case $\mu = 0$ and $\Sigma = I$. Define the following densities $g(x|\lambda) = \phi_d(x; 0, \lambda I)$, $p(\lambda) = \text{IG}(\lambda; \nu/2, \nu/2)$, and $T(z|x; \rho, \lambda) = \phi_d(z; \rho x, \lambda(1-\rho^2)I)$. Then, $p(\lambda|x) = \text{IG}(\lambda; (\nu+d)/2, (\nu+x')/2)$. Let $\zeta(x; \psi) = t_d(x; 0, I, \nu)$, it is straightforward to establish that

$$g(x|\lambda)p(\lambda) = p(\lambda|x)\zeta(x; \psi), \quad (19)$$

$$g(x|\lambda)T(z|x; \rho, \lambda) = g(z|\lambda)T(x|z; \rho, \lambda). \quad (20)$$

We define

$$T(z|x; \rho) = \int T(z|x; \rho, \lambda)p(\lambda|x)d\lambda = t_d(z; \tilde{\mu}(x), \tilde{\Sigma}(x), \tilde{\nu}),$$

where $\tilde{\mu}(x)$, $\tilde{\Sigma}(x)$ and $\tilde{\nu}$ are defined as in (4). We now establish reversibility.

$$\begin{aligned} \zeta(x; \psi)T(z|x; \rho) &= \zeta(x; \psi) \int T(z|x; \rho, \lambda)p(\lambda|x)d\lambda \\ &= \zeta(x; \psi) \int \frac{T(x|z; \rho, \lambda)g(z|\lambda)}{g(x|\lambda)} p(\lambda|x)d\lambda \end{aligned}$$

by equation (20),

$$= \zeta(x; \psi) \int \frac{T(x|z; \rho, \lambda)g(z|\lambda)}{p(\lambda|x)\zeta(x; \psi)} p(\lambda)p(\lambda|x)d\lambda$$

by equation (19)

$$\begin{aligned} &= \int T(x|z; \rho, \lambda)g(z|\lambda)p(\lambda)d\lambda \\ &= \zeta(z; \psi) \int T(x|z; \rho, \lambda)p(\lambda|z)d\lambda \end{aligned}$$

by equation (19),

$$= \zeta(z; \psi)T(x|z; \rho) \text{ as required.}$$

The result for general μ and Σ is obtained by noting that $t_d(x; \mu, \Sigma, \nu) = t_d(\Sigma^{-\frac{1}{2}}(x-\mu); 0, I, \nu)$.

The proof of Part (ii) follows from Part (iii) of Lemma 1.

Proof (Proof of Lemma 5) (i) follows from Lemma 2 (i); (ii) follows from Lemma 3.

Proof (Proof of Lemma 6) The proof follows from Lemma 1(ii) and Lemma 2(i).

Proof (Proof of Lemma 7) (i) Follows from Part (ii) of Lemma 2 because $g_M(z; \tilde{\psi}(\mathcal{H}^{n-1}))$ is the invariant density of both $T_{g_M}^{\text{CMH}}(z|x; \tilde{\psi}(\mathcal{H}^{n-1}))$ and $T_{g_M}^{\text{BS}}(z|x; \tilde{\psi}(\mathcal{H}^{n-1}))$; (ii) Follows from part (i) of Lemma 2; (iii) This follows from Part (iii) of Lemma 2. (iv) This follows from Lemma 3.