NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Control variates for the Metropolis-Hastings algorithm

by

Hugo Hammer and Håkon Tjelmeland

PREPRINT
STATISTICS NO. 8/2005

NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL `http://www.math.ntnu.no/preprint/statistics/2005/S8-2005.ps`

Hugo Hammer has homepage: `http://www.math.ntnu.no/~hammer`
E-mail: `hammer@stat.ntnu.no`
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology,
N-7491 Trondheim, Norway.

Control variates for the Metropolis-Hastings algorithm

HUGO HAMMER AND HÅKON TJELMELAND [1]

Abstract

We propose new control variates for variance reduction in the Metropolis–Hastings algorithm. We use variates that are functions of both the current state of the Markov chain and the proposed new state. This enable us to specify control variates which have known mean values for general target and proposal distributions. We develop the ideas for both the standard Metropolis–Hastings algorithm and the generalized reversible jump version. We present simulation results for four simulation examples. The variance reduction varies depending on the target distribution and proposal mechanisms used, the typical relative variance reduction is between 15% and 35%.

Key words: Control variate, Markov chain Monte Carlo, rejected states, variance reduction.

1 Introduction

Suppose we want to estimate the mean, μ, of a function $f(\mathbf{x})$ when $\mathbf{x} \in \mathbb{R}^n$ is distributed according to a target distribution $\pi(\cdot)$. If $\mathbf{x}$ is of high dimension and $\pi(\cdot)$ is complex, Markov chain Monte Carlo (MCMC) techniques are typically the only viable alternatives for evaluating μ. The most commonly used MCMC algorithm is the Metropolis-Hastings algorithm (Hastings, 1970), in which each iteration consists of two steps. Letting $\mathbf{x}$ denote the current state of the Markov chain, one first proposes a new state $\mathbf{y}$ for the Markov chain. Second, one accepts the proposed state $\mathbf{y}$ with a certain probability, otherwise keeping $\mathbf{x}$ as the current state of the Markov chain. After discarding a burn in period, the current states are essentially from the target distribution $\pi(\cdot)$ and can be used to estimate μ. The most natural and traditional estimator of μ is the empirical mean of $f(\mathbf{x})$.

It is possible to do better than the empirical mean when it comes to variance of the estimator. Liu (2001) points out some variance reduction methods. The antithetic variates method is due to Hammersley and Morton (1956). The idea is to make samples with negative correlation. Then the variance of the empirical mean is less than for independent samples. The Rao-Blackwellization is based on the facts that $\mathrm{E}[\mathrm{E}(f(\mathbf{x})|z)] = \mathrm{E}(f(\mathbf{x}))$ and $\mathrm{Var}[\mathrm{E}(f(\mathbf{x})|z)] \leq \mathrm{Var}(f(\mathbf{x}))$ for any random variable z, implying that if we can compute $\mathrm{E}(f(\mathbf{x})|z)$ analytically, this is a better estimator for μ then $f(\mathbf{x})$. The control variates method construct estimators that are linear combinations of the original unbiased estimator for μ and other random variables, called control variates. The control variates have zero expectation and are correlated with the original estimator. In this way it is possible to construct unbiased estimators with less variance.

Variance reduction techniques can also be used in a Metropolis–Hastings setting. Casella and Robert (1996) develop the Rao-Blackwellization technique for this situation. However, to evaluate the resulting estimator is quadratic in the number of Metropolis-Hastings iterations. Pinto and Neal (2001) use the same sequence of random numbers for two Markov chains, one sampling from the distribution of interest and the other from a Gaussian approximation, and the latter chain is used to construct a control variate. This gives large variance reductions if the target distribution can be reasonably well approximated with a Gaussian. Mira et al. (2003) use the zero-variance principle introduced in the physics literature to construct another estimator $\tilde{f}(\mathbf{x})$ that is the sum of the original estimator $f(\mathbf{x})$ and an additional term with expectation zero. Thus, the additional term can be considered a control variate. The method is only examined for simple low dimensional examples, so the potential is not known for more complicated problems. Atchadé

[1] Hugo Hammer is a PhD student and Håkon Tjelmeland an Assosiate Professor, Department of Mathematical Sciences, Norwegian University of Science and Technology, 7034 Trondheim, Norway (Email: Hugo.Hammer@stat.ntnu.no, Haakon.Tjelmeland@stat.ntnu.no)

and Perron (2005) construct control variates that is a function of all proposed states. However, to be able to know the mean of the control variate they must limit the attention to the independent proposal Metropolis–Hastings algorithm.

In the present paper we also use the control variate technique, but consider control variates that are functions of both the current states, $\mathbf{x}$, and the proposals, $\mathbf{y}$. This enable us to define several zero-mean control variates for general target and proposal distributions. We develop the ideas both for the standard Metropolis–Hastings procedure and the generalised reversible jump algorithm. The amount of variance reduction obtained varies depending on the target distribution and the proposal distribution used.

The paper is organised as follows. In Section 2 we give a brief introduction to the Metropolis-Hastings algorithm, and in Section 3 we describe the control variate technique. The new control variates are developed in Section 4 and simulation examples are presented in Section 5. Finally, in Section 6 we give some closing remarks.

2 MCMC simulation

Our aim is to make samples from a target distribution $\pi(\cdot)$. The idea behind MCMC algorithms is to construct a Markov chain, with transition kernel P say, which have $\pi(\cdot)$ as it's limiting distribution, see for example Liu (2001). An MCMC algorithm typically consists of two steps in each iteration. Letting $\mathbf{x}$ denote the current state of the Markov chain, one first proposes a new state $\mathbf{y}$ for the Markov chain and, second, one accepts $\mathbf{y}$ with a certain probability, otherwise one keeps $\mathbf{x}$ as the current state. Let $\mathbf{x}^1, \ldots, \mathbf{x}^N$ and $\mathbf{y}^1, \ldots, \mathbf{y}^N$ denote N states of the Markov chain and corresponding proposals in the MCMC algorithm, respectively, after having discarded a "burn-in" period. Then $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are samples from $\pi(\cdot)$. Thus, with

$$\mu = E(f(\mathbf{x})) = \int f(\mathbf{x})\pi(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{1}$$

the most frequently used estimator for μ is the sample mean,

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i). \tag{2}$$

As discussed above, it is possible to construct better estimators than the sample mean by using also the information in the rejected states and we return to this in Section 4. First we describe in more detail some MCMC algorithms. We start with the standard Metropolis-Hastings scheme.

2.1 Standard Metropolis-Hastings algorithm

Assume the target distribution to be continuous on $\mathbb{R}^n$ and let $\pi(\cdot)$ denote its density. A transition kernel P will then define a Markov chain with $\pi(\cdot)$ as it's limiting distribution if

$$\int_A \pi(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\mathbb{R}^n} \mathrm{P}(A|\mathbf{x})\pi(\mathbf{x})\mathrm{d}\mathbf{x} \quad \forall A \in \mathcal{F} \tag{3}$$

where $\mathcal{F}$ is the Borel σ-algebra on $\mathbb{R}^n$. It is hard to construct a suitable kernel P directly from (3). Instead it is common to restrict attention to time reversible chains, i.e. require the the detailed-balance condition,

$$\int_A \pi(\mathbf{x})\mathrm{P}(B|\mathbf{x})\mathrm{d}\mathbf{x} = \int_B \pi(\mathbf{x})\mathrm{P}(A|\mathbf{x})\mathrm{d}\mathbf{x} \quad \forall A, B \in \mathcal{F}. \tag{4}$$

If the chain satisfies (4) then the chain also satisfy (3). The transition kernel in the Metropolis-Hastings algorithm is given by

$$\mathrm{P}(A|\mathbf{x}) = \int_A q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{y} + I(\mathbf{x} \in A)r(\mathbf{x}),$$

where $q(\cdot|\mathbf{x})$ is a proposal density, $\alpha(\mathbf{y}|\mathbf{x})$ is the probability for accepting a proposal $\mathbf{y}$, $I(\cdot)$ is the indicator function and $r(\mathbf{x})$ the probability for remaining at $\mathbf{x}$. Equation (4) is satisfied if (Hastings, 1970)

$$\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x}) = \pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y}), \tag{5}$$

which has the most general solution

$$\alpha(\mathbf{y}|\mathbf{x}) = \frac{s(\mathbf{x},\mathbf{y})}{1 + \frac{\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})}{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})}}, \tag{6}$$

where $s(\mathbf{x},\mathbf{y}) \geq 0$ can be any symmetric function $s(\mathbf{x},\mathbf{y}) = s(\mathbf{y},\mathbf{x})$, that gives $\alpha(\mathbf{y}|\mathbf{x}) \leq 1 \;\; \forall \mathbf{x},\mathbf{y} \in \mathbb{R}^n$. In terms of asymptotic variance, the optimal choice for the acceptance probability is given by (Peskun, 1973)

$$\alpha(\mathbf{y}|\mathbf{x}) = \min\{1, R(\mathbf{x},\mathbf{y})\} \quad \text{where} \quad R(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})}. \tag{7}$$

2.2 Mode jumping algorithm

A variant of the Metropolis-Hastings algorithm is the mode jumping algorithm introduced in Tjelmeland and Hegstad (2001). This algorithm uses two proposal densities $q_0(\cdot|\mathbf{x})$ and $q_1(\cdot|\mathbf{x})$. Still letting $\mathbf{x}$ denote the current state of the Markov chain, the algorithm works as follows. First, we draw $k = 0$ or 1 at random and generate a new state $\mathbf{y}$ for the Markov chain from the proposal distribution $q_k(\cdot|\mathbf{x})$. Second, we accept the proposed state $\mathbf{y}$ with probability

$$\alpha(\mathbf{y}|\mathbf{x}) = \min\{1, R(\mathbf{x},\mathbf{y})\} \quad \text{where} \quad R(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q_{1-k}(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})q_k(\mathbf{y}|\mathbf{x})}. \tag{8}$$

Otherwise we keep $\mathbf{x}$ as the current state. One should note that the acceptance probability in (8) results from (6) for a specific choice of the function $s(\mathbf{x},\mathbf{y})$.

2.3 Reversible jump algorithm

The reversible jump algorithm is a generalization of the standard Metropolis-Hastings algorithm and is also based on the construction of a time reversible Markov chain. This short description is based on Waagepetersen and Sørensen (2001). The reversible jump algorithm is most often used when the target distribution have a sample space of varying dimension.

The target distribution $\pi(\cdot)$ is now the joint probability distribution of $\mathbf{x} = (m, \mathbf{z})$, where $m \in \{1, 2, \ldots, M\}$ is a model indicator and $\mathbf{z}$ is a real stochastic vector of possibly varying dimension. The vector $\mathbf{z}$ takes values in a set C defined as a union $C = \cup_{m=1}^{M} C_m$ of spaces $C_m = \mathbb{R}^{n_m}$, $n_m \geq 1$. Given we are inside a model m, $\mathbf{z}$ can only take values in C_m.

Letting $\mathbf{x} = (m, \mathbf{z})$ with $\mathbf{z} \in C_m$ be the current state of the Markov chain, each iteration of the algorithm works as follows. Propose a new state $\mathbf{y} = (m', \mathbf{z}')$ with $\mathbf{z}' \in C_{m'}$ by first proposing m' and a stochastic variable $\mathbf{u} \in \mathbb{R}^{n_{mm'}}$ from a proposal distribution $q(m', \mathbf{u}|\mathbf{x}) = p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})$, where $p_{mm'}$ is the proposal distribution for m' and $q_{mm'}(\mathbf{u}|\mathbf{z})$ the proposal distribution for $\mathbf{u}$. Note that $\mathbf{u}$ can be of varying dimension dependent on the value of m'.

Next, a one-to-one deterministic functional relation is assumed between $(\mathbf{z}, \mathbf{u})$ and $(\mathbf{z}', \mathbf{u}')$ for some $\mathbf{u}'$. Write $(\mathbf{z}', \mathbf{u}') = \phi_{mm'}(\mathbf{z}, \mathbf{u}) \Leftrightarrow (\mathbf{z}, \mathbf{u}) = \phi_{mm'}^{-1}(\mathbf{z}', \mathbf{u}')$. The proposed new value of the Markov chain is $\mathbf{y} = (m', \mathbf{z}')$ and it is accepted with probability

$$\alpha(\mathbf{y}|\mathbf{x}) = \min\{1, R(\mathbf{x},\mathbf{y})\} \quad \text{where} \quad R(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(m, \mathbf{u}'|\mathbf{y})}{\pi(\mathbf{x})q(m', \mathbf{u}|\mathbf{x})}|J| \tag{9}$$

and J is the Jacobi determinant for the transformation from $(\mathbf{z}, \mathbf{u})$ to $(\mathbf{z}', \mathbf{u}')$. More details of the algorithm is given in Appendix A.1.

3 Control variates

Suppose we have samples from the target distribution $\pi(\cdot)$ generated using one of the algorithms presented in Section 2 and we are interested in estimating μ defined in (1). We can then use the control variates technique (Liu, 2001) to reduce the variance in our estimate. More generally we can describe the control variates technique as follows. Let $\mathbf{x}^1, \ldots, \mathbf{x}^N$ be samples from the target distribution and suppose we use the sample to estimate μ with an unbiased estimator μ^*. Typically μ^* is the sample mean, equation (2). Suppose we have another random variable v, the control variate, with known expectation δ and which is correlated with μ^*. Without loss of generality we assume $\delta = 0$. Then for any value c we can also use the unbiased estimator

$$\tilde{\mu} = \mu^* + c \cdot v. \tag{10}$$

This gives,

$$\text{Var}(\tilde{\mu}) = \text{Var}(\mu^*) + c^2 \text{Var}(v) + 2c\text{Cov}(\mu^*, v),$$

which is minimized for

$$c = -\frac{\text{Cov}(\mu^*, v)}{\text{Var}(\mu^*)}. \tag{11}$$

For this optimal value of c, the relative variance reduction by using $\tilde{\mu}$ in stead of μ^* is

$$\frac{\text{Var}(\mu^*) - \text{Var}(\tilde{\mu})}{\text{Var}(\mu^*)} = [\text{Corr}(\mu^*, v)]^2. \tag{12}$$

Thus, the aim is to construct a control variate which has a high squared correlation with μ^*. Note that it is possible to use more then one control variate. Suppose we have the control variates $v_1, \ldots, v_m$. Then we can use the estimator

$$\tilde{\mu} = \mu^* + \sum_{k=1}^{K} c_i v_i$$

and find constants $c_1, \ldots, c_m$ which minimizes the variance. In the next section we discuss ways to construct control variates for Metropolis-Hastings algorithms.

Remark 1. Suppose we achieve a relative variance reduction a in the estimate of μ when we use a control variate compared with the sample mean. Alternatively, we can obtain the same variance reduction by running the chain for more iterations and estimate μ by the sample mean in the longer chain. Suppose the original chain was run for N_1 iterations and the longer chain for N_2 iterations. Then the longer chain gives a relative variance reduction of $a = (1/N_1 - 1/N_2)/(1/N_1) = (N_2 - N_1)/N_1$. Defining $r_a = N_2/N_1$, we get

$$r_a = \frac{1}{1-a}. \tag{13}$$

Thus, to get a relative variance reduction of a we need to increase the number of iterations with a factor of $1/(1-a)$.

4 Metropolis-Hastings and control variates

As in Section 2, let $\mathbf{x}^1, \ldots, \mathbf{x}^N$ and $\mathbf{y}^1, \ldots, \mathbf{y}^N$ denote the states of the Markov chain and the corresponding proposals in an MCMC algorithm, respectively. In this section we construct control variates for the Metropolis-Hastings algorithm.

4.1 A first control variate

Consider a control variate of the form

$$v = \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}^i, \mathbf{y}^i) \tag{14}$$

where $g(\cdot,\cdot)$ is a function to be specified. Recall the standard Metropolis-Hastings algorithm from Section 2.1. The estimator $\tilde{\mu}$ in (10) is then unbiased if $E(g(\mathbf{x},\mathbf{y})) = 0$ where $(\mathbf{x},\mathbf{y}) \sim \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})$. Define the function

$$g_0(\mathbf{x},\mathbf{y}) = w_1(\mathbf{x},\mathbf{y})f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y})f(\mathbf{y}) \tag{15}$$

where $w_1(\cdot,\cdot)$ and $w_2(\cdot,\cdot)$ are weight functions to be specified. We can then use $g_0(\cdot,\cdot)$ as $g(\cdot,\cdot)$ in (14) if

$$E(g_0(\mathbf{x},\mathbf{y})) = \iint [w_1(\mathbf{x},\mathbf{y})f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y})f(\mathbf{y})]\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} = 0. \tag{16}$$

To find weight functions that satisfy (16), split the integral in two parts and change the order of integration in the last integral. This gives the requirement

$$\iint w_1(\mathbf{x},\mathbf{y})f(\mathbf{x})\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} + \iint w_2(\mathbf{y},\mathbf{x})f(\mathbf{x})\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} = 0 \tag{17}$$

Thus a sufficient condition for $E(g_0(\mathbf{x},\mathbf{y})) = 0$ is

$$w_1(\mathbf{x},\mathbf{y})\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x}) = -w_2(\mathbf{y},\mathbf{x})\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y}), \tag{18}$$

which is satisfied for

$$w_1(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x}) + \pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})} \quad \text{and} \quad w_2(\mathbf{x},\mathbf{y}) = -w_1(\mathbf{x},\mathbf{y}). \tag{19}$$

Substituting (19) into (15) gives

$$g_0(\mathbf{x},\mathbf{y}) = \frac{R(\mathbf{x},\mathbf{y})}{1+R(\mathbf{x},\mathbf{y})}(f(\mathbf{x}) - f(\mathbf{y})) \tag{20}$$

which again can be substituted into equation (14) and we have defined a control variate that uses all the accepted and proposed states in the Metropolis-Hastings algorithm. We end this section with some remarks concerning the control variate just defined.

Remark 2. With similar calculations like above we get the same control variate for the mode jumping and the reversible jump algorithms. In Appendix A.1 we give a proof of (20) in the reversible jump setting.

Remark 3. The two weight functions in (19) may be multiplied with any function $s(\mathbf{x},\mathbf{y})$ that is symmetric in $\mathbf{x}$ and $\mathbf{y}$ and (18) will still hold. Note that this function corresponds essentially to the symmetric function that can be chosen in Hastings (1970) original acceptance probability, our equation (6). However, an important difference is that in (6) the choice of $s(\mathbf{x},\mathbf{y})$ must ensure that the acceptance probability is not larger than one, whereas no such requirement exists for the weight functions.

Remark 4. The connection between the Metropolis–Hastings acceptance probability and the weights in (15) can also be seen from the following. With (15) the estimator for μ becomes

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N}[1 + w_1(\mathbf{x}^i,\mathbf{y}^i)]f(\mathbf{x}^i) + w_2(\mathbf{x}^i,\mathbf{y}^i)f(\mathbf{y}^i).$$

Thus, the total weights given to the current state $\mathbf{x}$ and the proposal $\mathbf{y}$ are $1 + w_1(\mathbf{x},\mathbf{y})$ and $w_2(\mathbf{x},\mathbf{y})$, respectively. If we require these to sum to unity we get $w_2(\mathbf{x},\mathbf{y}) = -w_1(\mathbf{x},\mathbf{y})$. In turn substituting this into (18) we get a requirement for $w_1(\mathbf{x},\mathbf{y})$ which exactly corresponds to (5) for the Metropolis–Hastings acceptance probability. In particular, our choice in (19) corresponds to the Barker (1965) acceptance probability.

Remark 5. To calculate the estimate $\tilde{\mu}$ all we need is the current state $\mathbf{x}$, the proposal $\mathbf{y}$ and the acceptance ratio $R(\mathbf{x},\mathbf{y})$. This is quantities that are already calculated to run the Metropolis-Hastings algorithm. Thus, to calculate $\tilde{\mu}$ is essentally of the same computational complexity as calculating $\hat{\mu}$.

4.2 Control variates including the acceptance indicator

Now we go one step further and define control variates not only depending on the current and proposed states for each iteration but also the information whether we get acceptance or not. Define the acceptance indicator $\gamma \in \{0,1\}$. As before, let $\mathbf{x}$ and $\mathbf{y}$ denote the current state of the Markov chain and the proposal respectively. We let $\gamma = 1$ if the proposal is accepted and $\gamma = 0$ otherwise. Thus

$$\mathrm{P}(\gamma = r|\mathbf{x},\mathbf{y}) = [\alpha(\mathbf{y}|\mathbf{x})]^r[1-\alpha(\mathbf{y}|\mathbf{x})]^{(1-r)}, \quad r = 0,1. \tag{21}$$

Consider control variates of the form

$$v = \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}^i,\mathbf{y}^i,\gamma^i), \tag{22}$$

where $g(\cdot,\cdot,\cdot)$ is some function to be specified and $\gamma^1,\ldots,\gamma^N$ are the acceptance indicators for each iteration of the Markov chain. We then have the following result that holds for the standard Metropolis-Hastings, the mode jumping and the reversible jump algorithms

Theorem 1. *Let* $(\mathbf{x},\mathbf{y},\gamma) \sim \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})P(\gamma|\mathbf{x},\mathbf{y})$ *and define the functions*

$$g_1(\mathbf{x},\mathbf{y},\gamma) = \begin{cases} \alpha(\mathbf{y}|\mathbf{x})f(\mathbf{x}) & \text{if } \gamma = 0,\\ -[1-\alpha(\mathbf{y}|\mathbf{x})]f(\mathbf{x}) & \text{if } \gamma = 1, \end{cases} \tag{23}$$

$$g_2(\mathbf{x},\mathbf{y},\gamma) = \begin{cases} \alpha(\mathbf{y}|\mathbf{x})f(\mathbf{x}) & \text{if } \gamma = 0,\\ -[1-\alpha(\mathbf{x}|\mathbf{y})]f(\mathbf{y}) & \text{if } \gamma = 1, \end{cases} \tag{24}$$

$$g_3(\mathbf{x},\mathbf{y},\gamma) = \begin{cases} \alpha(\mathbf{y}|\mathbf{x})f(\mathbf{y}) & \text{if } \gamma = 0,\\ -[1-\alpha(\mathbf{x}|\mathbf{y})]f(\mathbf{x}) & \text{if } \gamma = 1. \end{cases} \tag{25}$$

$$g_4(\mathbf{x},\mathbf{y},\gamma) = \begin{cases} \alpha(\mathbf{y}|\mathbf{x})f(\mathbf{y}) & \text{if } \gamma = 0,\\ -[1-\alpha(\mathbf{y}|\mathbf{x})]f(\mathbf{y}) & \text{if } \gamma = 1, \end{cases} \tag{26}$$

Then $E(g_l(\mathbf{x},\mathbf{y},\gamma)) = 0$ *for* $l = 1,\ldots,4$.

This is proved in much the same way that we calculated (20). A proof for the standard Metropolis-Hastings algorithm is given in Appendix A.2. Then we can set $g(\mathbf{x},\mathbf{y},\gamma) = g_l(\mathbf{x},\mathbf{y},\gamma)$ for any of $l = 1,\ldots,4$ in (22) and we have constructed four more control variates.

Remark 6. To calculate the control variates presented in this section we also only need the current state $\mathbf{x}$, the proposal $\mathbf{y}$ and the acceptance ratio $R(\mathbf{x},\mathbf{y})$, similar to what we observed in Remark 5.

4.3 Calculation of the control variate estimate

When calculating the estimate $\tilde{\mu}$ in (10) for one of the control variates defined in Sections 4.1 and 4.2 we want to use the optimal value for the constant c. The optimal value in (11) can not be directly used as both $\mathrm{Cov}(\mu^*,v)$ and $\mathrm{Var}(\mu^*)$ are unknown. The two quantities may be estimated via time series analysis methods, see Priestly (1981), Geier (1992) and Han and Green (1992). We adopt a very simple method based on dividing the chain into batches. More precisely, we divide the N iterations into M batches of equal lengths, where N and M must be chosen so that N/M is much larger than the correlation length of the simulated Markov chain. We estimate the optimal value for c by

$$\hat{c} = -\frac{\widehat{\text{Cov}(\hat{\mu}, v)}}{\widehat{\text{Var}(\hat{\mu})}} = -\frac{\frac{1}{M}\sum_{j=1}^{M}(\hat{\mu}^j - \hat{\mu})(v^j - v)}{\frac{1}{M}\sum_{j=1}^{M}(\hat{\mu}^j - \hat{\mu})^2}, \tag{27}$$

where $\hat{\mu}^j$ and v^j are the sample mean and the control variate for batch number j, respectively, and $\hat{\mu} = (1/M)\sum_{j=1}^{M}\hat{\mu}^j$ and $v = (1/M)\sum_{j=1}^{M}v^j$. Finally, we substitute $\hat{c}$ for c in (10).

One should note that when using the same Markov chain run first to estimate the optimal value for c and thereafter computing $\hat{\mu}$ with $c = \hat{c}$, the resulting $\hat{\mu}$ is no longer unbiased. If the unbiasedness is considered really important, a better alternative is to do two independent Markov chain runs, runs A and B say. One may then estimate the optimal value for c from each of the two runs, getting $\hat{c}_A$ and $\hat{c}_B$ say, and thereafter computing $\hat{\mu}_A$ from run A with $c = \hat{c}_B$ and $\hat{\mu}_B$ from run B with $c = \hat{c}_A$. Then both $\hat{\mu}_A$ and $\hat{\mu}_B$ are unbiased estimators for μ, and so is $\frac{1}{2}(\hat{\mu}_A + \hat{\mu}_B)$.

5 Simulation examples

In this section we try the control variates in four simulation examples. First we present the results for a toy Gaussian example. The three other examples is based on previously analysed and published data sets and we use our control variates with the exact same models and simulation algorithms previously presented.

5.1 Toy Gaussian example

Let $\pi(\cdot)$ be a ten dimensional standard Gaussian distribution. We estimate the mean of the function $f(\mathbf{x}) = x_1$, where x_1 denotes the first component of the vector $\mathbf{x}$, of course $\mu = E(x_1) = 0$. We simulate using the Metropolis-Hastings algorithm and try two different proposal distributions. The first is the normal random walk proposal (Liu, 2001)

$$q_{RW}(\mathbf{y}|\mathbf{x}) = N_{10}(\mathbf{y}|\mathbf{x}, \sigma^2\mathbf{I}), \tag{28}$$

where $N_n(\cdot|\nu, \mathbf{\Sigma})$ is the density function of an n-dimensional Gaussian distribution with mean ν and covariance matrix $\mathbf{\Sigma}$, σ^2 is the proposal variance and I the identity matrix. The second is the Langevin proposal (Roberts and Rosenthal, 1998)

$$q_L(\mathbf{y}|\mathbf{x}) = N_{10}(\mathbf{y}|\mathbf{x} + \frac{\sigma^2}{2}\nabla\log\{\pi(\mathbf{x})\}, \sigma^2\mathbf{I}). \tag{29}$$

We present results for the control variates defined by (20) and (23), and denote them v_0 and v_1, respectively. We also tried the other control variates discussed in Section 4, but these produced less promising results. We consider the two estimators

$$\tilde{\mu}_{(0)} = \hat{\mu} + c \cdot v_0, \tag{30}$$
$$\tilde{\mu}_{(0,1)} = \hat{\mu} + c_1 \cdot v_0 + c_2 \cdot v_1. \tag{31}$$

The results are summarised in Figure 1. The left and right columns contain results for the normal random walk and Langevin proposals, respectively. For eight different values of σ, the first row gives the correlation between the sample mean and v_0, the second row the correlation between the sample mean and v_1, the third row gives the correlation between v_0 and v_1, and in the forth row the upper and lower curves show the relative variance reduction using the estimators $\tilde{\mu}_{(0)}$ and $\tilde{\mu}_{(0,1)}$ compared with the sample mean, respectively. The maximal relative variance reduction is slightly above and slightly below 30% for the random walk and Langevin proposals, respectively. This corresponds to an r_a value of about 1.43, i.e. the variance reduction is the same as one would have obtained by increasing the number of iterations by 43%. One can note that the largest variance reductions are obtained when σ is close to the optimal values identified in Roberts et al.

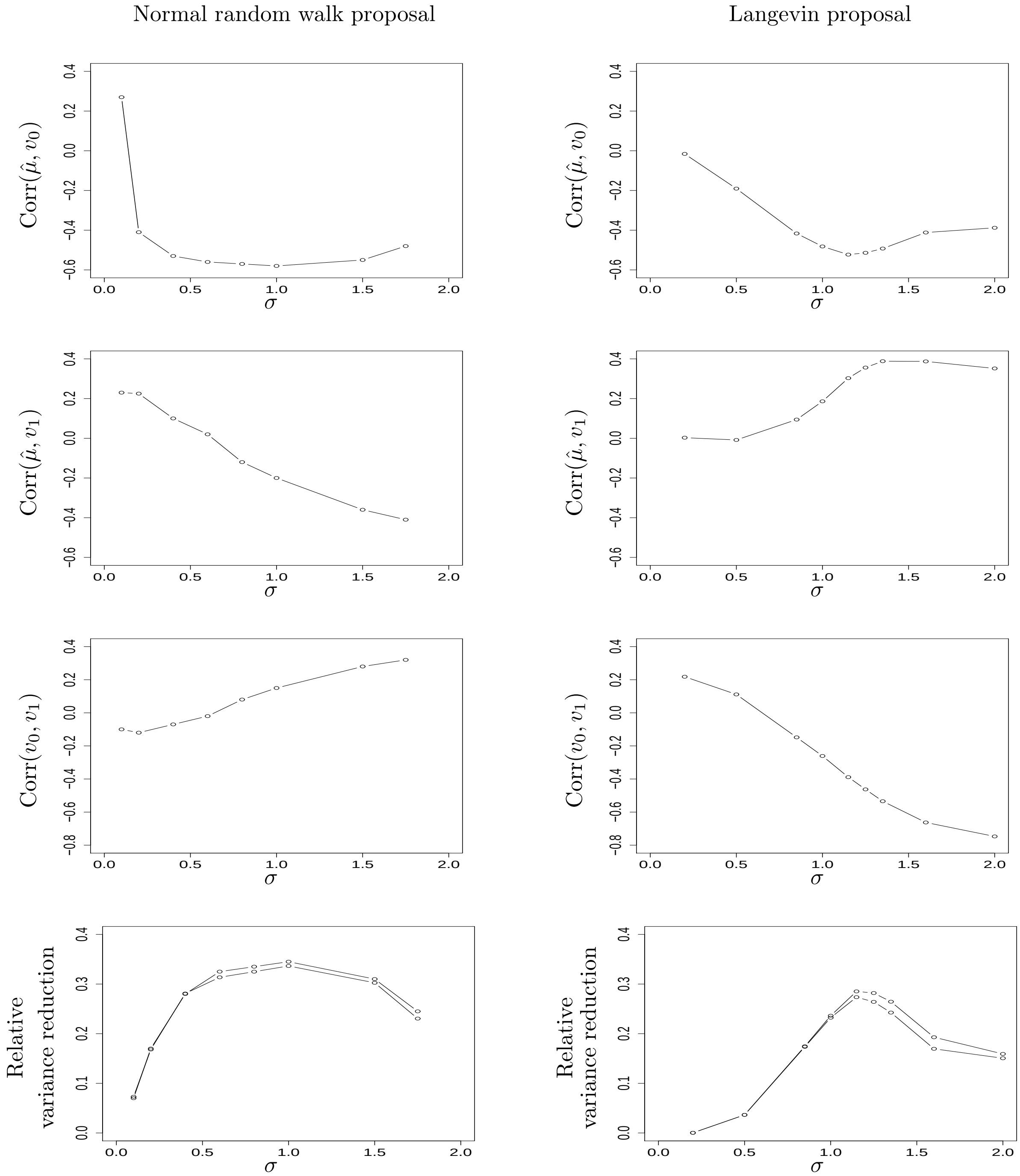

Figure 1: Toy Gaussian example: The left and right columns contain results for the normal random walk and Langevin proposals, respectively. As a function of σ we have: First row, $\mathrm{Corr}(\hat{\mu}, v_0)$. Second row, $\mathrm{Corr}(\hat{\mu}, v_1)$. Third row, $\mathrm{Corr}(\hat{\mu}, v_0)$. Forth row, relative variance reduction using the estimators $\tilde{\mu}_{(0)}$ (upper curve) and $\tilde{\mu}_{(0,1)}$ (lower curve) compared with the sample mean.

(1997), for random walk proposals, and in Roberts and Rosenthal (1998), for Langevin proposals. We also notice that the improvement using $\tilde{\mu}_{(0,1)}$ is minimal compared with $\tilde{\mu}_{(0)}$ even if $\text{Corr}(\hat{\mu}, v_1)$ is significantly different from zero for large values of σ. This can be understood from the plots in the third row. When $\text{Corr}(\hat{\mu}, v_1)$ is large, $\text{Corr}(v_0, v_1)$ is also large in absolute value, so v_0 and v_1 are kind of similar and v_1 do not give much additional information in the estimation.

5.2 Gaussian Markov random field example

In this section we consider a simple but much analysed binomial time series taken from Kitagawa (1987). Each day during the years 1983 and 1984 it was recorded whether there was more than one mm rainfall in Tokyo. We are interested in estimating the underlying probability p_t of rainfall at calendar day $t = 1, \ldots, 366$. We use the likelihood function (Kitagawa, 1987)

$$f(\mathbf{d}|\chi) = \prod_{t=0}^{n-1} \binom{n_t}{d_t} p(\chi_t)^{d_t} (1 - p(\chi_t))^{(n_t - d_t)} \tag{32}$$

where $p(\cdot)$ is the logit link and $\mathbf{d} = (d_1, \ldots, d_n)^T$ is the number of times it rained more than 1 mm on the different calendar days. Further, $n_t = 2$ for $t \neq 60$ and $n_{60} = 1$ which corresponds to February 29. We use two prior models for $\chi = (\chi_1, \ldots, \chi_n)$, circular Gaussian Markov random fields with first and second order neighbourhood with precision $\kappa \sim \Gamma(a, b)$, the gamma distribution. The circularity in the priors is based on Rue and Held (2004) and they refer to the priors as RW1 and RW2, respectively. With $\mathbf{x} = (\chi, \kappa)$ the target distribution of interest is the posterior distribution $\pi(\mathbf{x}|\mathbf{d})$ and as an estimator for p_t we use $E(p(\chi_t)|\mathbf{d})$. Thus, in the notation introduced in Section 2 we have $f(\mathbf{x}) = p(\chi_t)$.

To simulate from the posterior distribution we use an algorithm from Rue and Held (2004). We simulate from the posterior $\pi(\chi, \kappa|\mathbf{d})$ by first proposing a new precision value $\kappa' = f\kappa$ where f has density $\pi(f) \propto 1 + 1/f$ on the interval $[1/F, F]$. This conveniently makes

$$\frac{\pi(\kappa'|\kappa)}{\pi(\kappa|\kappa')} = 1. \tag{33}$$

Next we propose a new value χ' conditioned on κ' for the Markov random field from a second order Taylor approximation to the posterior distribution and simultaneously accept or reject the proposal (χ', κ'). We simulate using the library `GMRFlib` (Rue and Follestad, 2002).

We consider all the five control variates defined in Section 4. We denote the variates defined by (20), (23), (24), (25) and (26) with v_0, v_1, v_2, v_3 and v_4, respectively. We present results for the corresponding five estimators,

$$\tilde{\mu}_{(l)} = \hat{\mu} + c \cdot v_l \quad \text{for } l = 1, \ldots, 4. \tag{34}$$

In the simulations we adopt the hyper-parameter values $a = 1.0$, $b = 0.000289$ used in Rue and Held (2004). Further, we chose $F = 7$ which gave an acceptance rate around 30%. For each model, we ran the algorithm for 300000 iterations (after convergence) and used the realisations to estimate the relative variance reduction for each calendar day $t = 1, \ldots, 366$. We also used bootstrapping, the percentile method (Efron, 1981), to estimate corresponding confidence intervals. The results are summarised in Figure 2 and Table 1. In Figure 2, the left and right columns show results for the RW1 and RW2 priors, respectively. From top to bottom the five rows show, for each calendar day, confidence intervals for the relative variance reduction when using $\mu_{(0)}$ to $\mu_{(4)}$. One can note that the first control variate, v_0, is the only one that substantially contribute to any variance reduction. For v_0 the reduction is between 20 and 30%. In Table 1 we give more details for three arbitrary chosen calendar days, $t = 64$, 184 and 308. For each of the two models, the table give confidence intervals for p_t based on $\widehat{\mu}$ and $\tilde{\mu}_{(0)}$, respectively, together with corresponding estimated standard deviations and r_a. We note that the gain of using our control variates is somewhat lower here than in the Gaussian toy example.

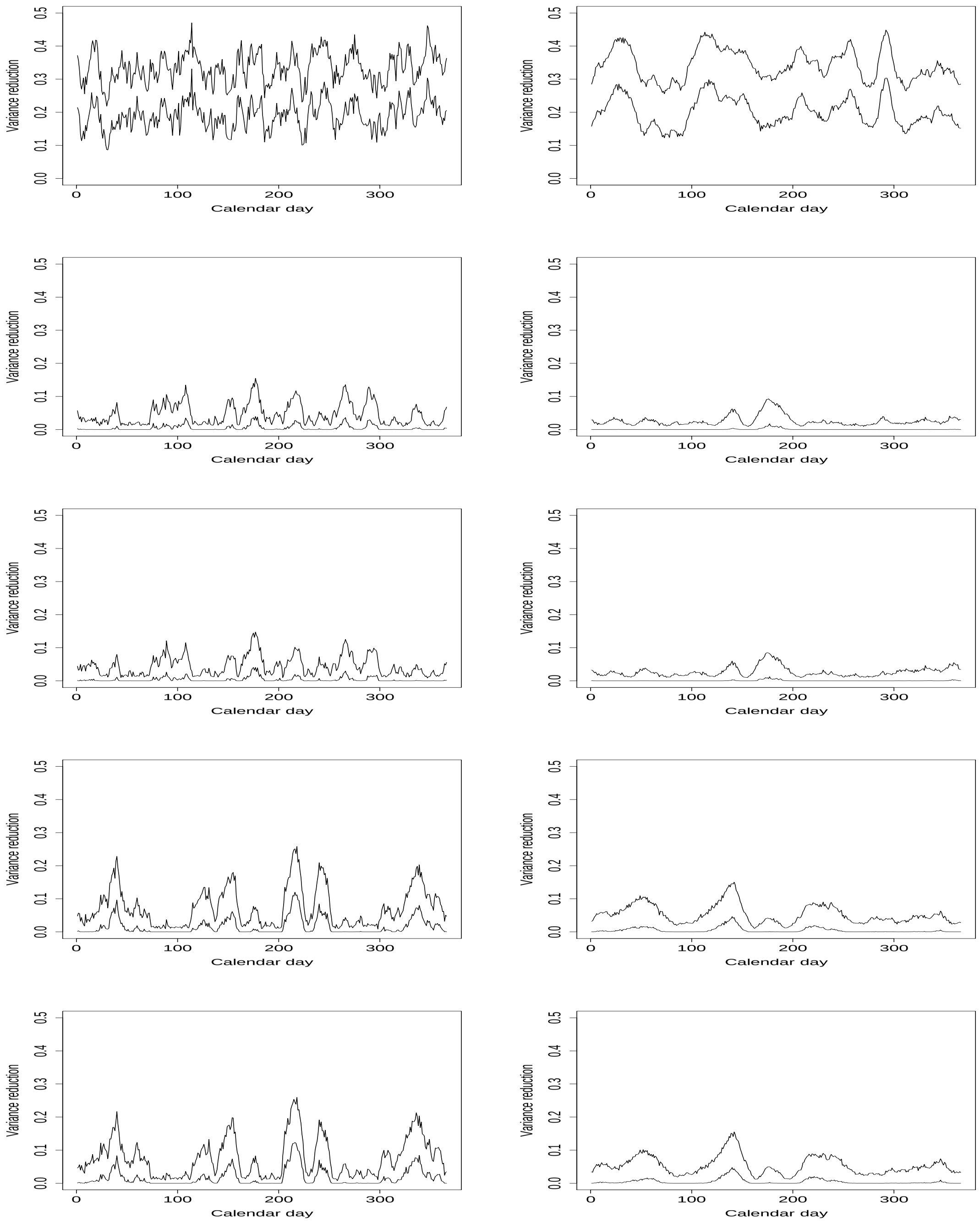

Figure 2: Gaussian Markov random field example: Confidence intervals for relative variance reduction for p_t, $t = 1, \ldots, 366$ using $\tilde{\mu}_{(i)}$, $i = 0, \ldots, 4$ compared with just using the sample mean. The left and right column contains results for RW1 and RW2 respectively. In row i we consider estimator $\tilde{\mu}_{(i-1)}$.

Table 1: Gaussian Markov random field example: Confidence intervals, standard deviations for p_t and corresponding r_a for $t = 64, 184$ and 308, when using the estimators $\hat{\mu}$ and $\tilde{\mu}_{(0)}$.

RW1	Conf.int. using $\hat{\mu}$	Conf.int. using $\tilde{\mu}_{(0)}$	Sd($\hat{\mu}$)	Sd($\tilde{\mu}_{(0)}$)	r_a
p_{64}	(0.21401, 0.21514)	(0.21381, 0.21480)	$2.89 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	1.30
p_{184}	(0.41674, 0.41820)	(0.41706, 0.41832)	$3.72 \cdot 10^{-4}$	$3.22 \cdot 10^{-4}$	1.33
p_{308}	(0.17317, 0.17415)	(0.17309, 0.17395)	$2.51 \cdot 10^{-4}$	$2.18 \cdot 10^{-4}$	1.33

RW2	Conf.int. using $\hat{\mu}$	Conf.int. using $\tilde{\mu}_{(0)}$	Sd($\hat{\mu}$)	Sd($\tilde{\mu}_{(0)}$)	r_a
p_{64}	(0.23700, 0.23788)	(0.23694, 0.23772)	$2.25 \cdot 10^{-4}$	$1.97 \cdot 10^{-4}$	1.30
p_{184}	(0.48155, 0.48264)	(0.48170, 0.48265)	$2.77 \cdot 10^{-4}$	$2.42 \cdot 10^{-4}$	1.31
p_{308}	(0.18988, 0.19064)	(0.18989, 0.19056)	$1.93 \cdot 10^{-4}$	$1.70 \cdot 10^{-4}$	1.29

Table 2: Mode jumping example: For the empirical mean, $\widehat{\mu}$, and each of $\tilde{\mu}_{(0)}, \ldots, \tilde{\mu}_{(4)}$, confidence interval for the amount of probability mass contained in the smaller posterior mode, corresponding standard deviation, estimated r_a-value and confidence interval for the relative variance reduction.

Estimator	Confidence interval	Standard.dev.	r_a	Var. red.
$\hat{\mu}$	(0.01477, 0.01891)	$1.06{\cdot}10^{-03}$		
$\tilde{\mu}_{(0)}$	(0.01489, 0.01795)	$7.81{\cdot}10^{-04}$	1.83	(0.407, 0.496)
$\tilde{\mu}_{(1)}$	(0.01467, 0.01877)	$1.05{\cdot}10^{-03}$	1.02	(0.001, 0.053)
$\tilde{\mu}_{(2)}$	(0.01467, 0.01874)	$1.04{\cdot}10^{-03}$	1.03	(0.007, 0.072)
$\tilde{\mu}_{(3)}$	(0.01504, 0.01834)	$8.43{\cdot}10^{-04}$	1.57	(0.316, 0.411)
$\tilde{\mu}_{(4)}$	(0.01470, 0.01809)	$8.39{\cdot}10^{-04}$	1.58	(0.325, 0.416)

5.3 Mode jumping example

In this section we reconsider the example in Section 4.2 of Tjelmeland and Hegstad (2001) where a mixture model is used for a data set originally presented in Brooks et al. (1997) concerning fetal deaths in litters of mice. The model is a mixture of beta-binomial and binomial distribution

$$p(\lambda|\eta) = \gamma \left[\binom{\eta}{\lambda} \prod_{r=0}^{\lambda-1} \frac{\mu + r\theta}{1 + r\theta} \prod_{r=0}^{\eta-\lambda-1} \frac{1 - \mu - r\theta}{1 + r\theta} \right] + (1 - \gamma) \left[\binom{\eta}{\lambda} \nu^{\lambda} (1 - \nu)^{\eta-\lambda} \right], \qquad (35)$$

where λ is the number of deaths and η the number of implants or fetuses. The model parameters are $\gamma \in [0, 1]$, $\mu \in [0, 1]$, $\theta \geq 0$ and $\nu \in [0, 1]$, to which independent vague prior distributions are assigned. The target distribution of interest is the posterior distribution of the parameters given the data. This posterior turns out to have two distinct modes and our focus here is the probability mass contained in the smaller mode.

Given a current state $\mathbf{x} = (\gamma, \mu, \theta, \nu)$, a potential new state $\mathbf{y}$ is generated as follows. First, a vector φ is generated from a very wide Gaussian distribution and $k = 0$ or 1 is chosen with equal probabilities. Second, a local optimisation algorithm is run, starting in $\mathbf{x} + \varphi$ if $k = 0$ and in $\mathbf{x} - \varphi$ if $k = 1$. Third, a Gaussian or t-distribution is fitted to the local optimum located and, finally, the potential new state y is generated from the fitted distribution. For more details of the algorithm used, see the reference given above.

We consider the same control variates and estimators that we considered in Section 5.2. In Table 2 we summarise the results. For the empirical mean, $\hat{\mu}$, and each of $\tilde{\mu}_{(0)}, \ldots, \tilde{\mu}_{(4)}$ we give the resulting confidence interval for the probability mass contained in the smaller mode, corresponding standard deviation, the r_a-value and confidence intervals for the relative variance reduction. The confidence interval for the relative variance reduction is again obtained using bootstrapping. We see that we obtain very satisfying results for three of the estimators, namely $\tilde{\mu}_{(0)}$, $\tilde{\mu}_{(3)}$ and $\tilde{\mu}_{(4)}$. However, v_0, v_3 and v_4 are highly correlated so very little extra can be gained by using more than one control variate.

Table 3: Reversible jump example: Confidence intervals, standard deviation and corresponding r_a for P($m = 2$), P($m = 5$) and P($m = 8$) for the enzyme and the acidity data using the estimators $\hat{\mu}$ and $\tilde{\mu}_{(0)}$.

Enzyme	Conf.int. using $\hat{\mu}$	Conf.int. using $\tilde{\mu}_{(0)}$	Sd($\hat{\mu}$)	Sd($\tilde{\mu}_{(0)}$)	r_a
P($m = 2$)	(0.02312, 0.02387)	(0.02349, 0.02415)	$1.92 \cdot 10^{-4}$	$1.69 \cdot 10^{-4}$	1.29
P($m = 5$)	(0.20807, 0.20950)	(0.20815, 0.20950)	$3.65 \cdot 10^{-4}$	$3.43 \cdot 10^{-4}$	1.13
P($m = 8$)	(0.01590, 0.01632)	(0.01582, 0.01621)	$1.06 \cdot 10^{-4}$	$1.01 \cdot 10^{-4}$	1.11

Acidity	Conf.int. using $\hat{\mu}$	Conf.int. using $\tilde{\mu}_{(0)}$	Sd($\hat{\mu}$)	Sd($\tilde{\mu}_{(0)}$)	r_a
P($m = 2$)	(0.07549, 0.07663)	(0.07550, 0.07655)	$2.91 \cdot 10^{-4}$	$2.67 \cdot 10^{-4}$	1.19
P($m = 5$)	(0.18158, 0.18254)	(0.18142, 0.18230)	$2.46 \cdot 10^{-4}$	$2.25 \cdot 10^{-4}$	1.20
P($m = 8$)	(0.03598, 0.03655)	(0.03603, 0.03658)	$1.46 \cdot 10^{-4}$	$1.41 \cdot 10^{-4}$	1.08

5.4 Reversible jump example

In this example we consider the model and data sets presented in Richardson and Green (1997). For three different data sets, Richardson and Green (1997) use a mixture of Gaussian densities with a stochastic number of mixture components. Thus, the resulting posterior is defined on space of varying dimension and a reversible jump algorithm is used to generate samples from it. We use the same three data sets as in Richardson and Green (1997) and also adopt exactly the same model and simulation algorithm. The reversible jump algorithm used to simulate from the posterior uses three groups of proposals. First, Gibbs updates is used for allocation of observations into the different mixture components and for the model parameters. Second, reversible jump moves are used either to split one mixture component into two or merging two mixture components into one. The last update type is to propose to remove an existing or to add a new empty mixture component. An empty mixture component is a component to which no data is allocated. For more details on the model and the simulation algorithm used we refer to the reference given above.

Our focus here is on the number of mixture components, denoted by m, and in particular its posterior distribution. Thus, for each possible value of m we let $f(\mathbf{x})$ be an indicator function that is one whenever the number of mixture components equals the specified value, and zero otherwise. In the estimation process we only include updates of the two last types described above. The first update type does not change the value of m and therefore does not give any additional relevant information. We consider the same five control variates as in Sections 5.2 and 5.3. However, we now define two control variates of each type, one for each of the two update types involving a change in the value of m. For $l = 0, 1, \ldots, 4$ let $v_{l,2}$ be the control variate defined from equations (20), (23), (24), (25) and (26), respectively, for the second update type described above, and let $v_{l,3}$ be the corresponding control variate based on the third update type. We then consider the five estimators

$$\tilde{\mu}_{(l)} = \widehat{\mu} + c_2 v_{l,2} + c_3 v_{l,3} \quad \text{for } l = 0, 1, \ldots, 4. \tag{36}$$

We ran the algorithm for 32 million sweeps (as defined in Richardson and Green (1997)) after convergence and used the realisations to estimate the relative variance reduction for each value of m. The results are summarised in Figure 3 and Table 3. In Figure 3, the left and right columns show results for two different data sets, the "enzyme" and the "acidity" data sets. The results for the third data set, the "galaxy" data, were less promising. From top to bottom the five rows show 95% confidence intervals for the relative variance reduction for each value of m, when using $\tilde{\mu}_{(0)}$ to $\tilde{\mu}_{(4)}$. As in the previous examples, bootstrapping is used to generate the confidence intervals. The first control variate is again the most effective one, but estimators $\tilde{\mu}_{(3)}$ and $\tilde{\mu}_{(4)}$ also give noticeable variance reductions. Note that $\tilde{\mu}_{(3)}$ and $\tilde{\mu}_{(4)}$ are the same control variates that work well in the mode jumping example. In Table 3 we give more detailed results for $m = 2$, 5 and 8. For each of the two data sets considered, the table give confidence intervals for the posterior probability based on $\widehat{\mu}$ and $\tilde{\mu}_{(0)}$, respectively, together with corresponding estimated standard deviations and r_a. One can note that the estimated posterior probabilities given in Richardson and Green (1997) fall

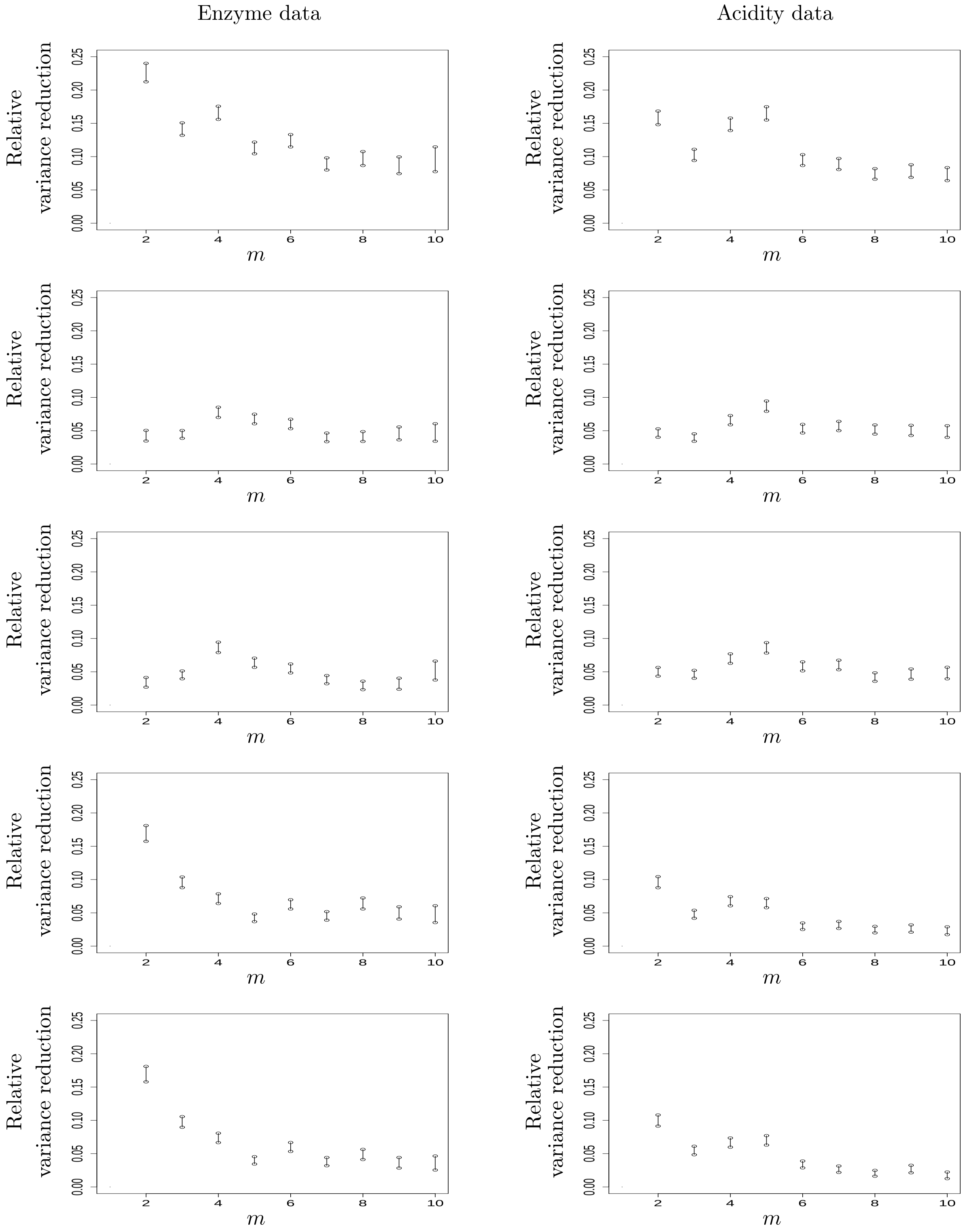

Figure 3: Reversible jump example: Confidence intervals for relative variance reduction in estimation of $\mathrm{P}(m = i)$, $i = 2, \ldots, 10$ compared with just using the sample mean. This covers the most plausible values of m. The left and right column contains results for the Enzyme and the Acidity data, respectively. In row i we consider estimator $\tilde{\mu}_{(i-1)}$.

outside our confidence intervals. This is not unreasonable as we are using a much larger number of iterations.

6 Closing remarks

In this paper we introduce five new control variates that can be used together with the Metropolis-Hastings algorithm. We consider functions of both the current state of the Markov chain and the proposed new state, and this enables us to construct control variates with known mean values for general target and proposal distributions. We work out the ideas for both the standard Metropolis–Hastings setting and for the more general reversible jump situation.

To apply the new control variates require very little extra effort, both in terms of implementation and computation time. The use of the control variates is best implemented as a program post-prosessing the simulation output. The only change necessary in the simulation code is to store the proposed new value $\mathbf{y}$ and the acceptance ratio $R(\mathbf{x}, \mathbf{y})$ in addition to the current state $\mathbf{x}$. The post-processing step can be implemented as a general program, the only part that need to be recoded for each problem is the function of interest, $f(\mathbf{x})$. The extra computation time is proportional to the number of iterations run and will usually be neglectable compared to the simulation time. In contrast, using the Rao-Blackwellization method in a Metropolis–Hastings setting give an extra cost that goes as the square of the number of iterations and will therefore dominate the total compution time when the number of iterations is large.

In four simulation examples we have explored what variance reduction can be obtained by using the new control variates. Two of the variates, v_1 and v_2 seem to be of little practical use. The best results are consistently obtained for the simplest control variate, v_0, but also v_3 and v_4 give promising results in two of the examples. However, as v_3 and v_4 typically seem to be highly correlated with v_0, we expect it to be most reasonable to use only v_0.

The relative variance reduction obtained varies significantly depending on both the target distribution of interest, the proposal distributions used, and the expectation of interest. The largest reduction in our examples was 45%, which corresponds to $r_a = 1.83$. Thus, to obtain a similar variance reduction, the run length would have to be increased by 83%. This number is from our mode jumping example where the simulation run took several days of computation time, so in this situation such a variance reduction is definitely of practical use.

Comparing the variance reductions in our four examples, the largest reductions seem to be obtained with proposal distributions that propose large changes. This is certainly true for the mode jumping example. In the Gaussian toy example the variance reductions are also largest when σ is reasonably large, but not too large as then most of the proposals will be far out in the tail of the distribution. The variance reductions are smaller for the reversible jump example where, loosely speaking, the proposed changes are rather small. This conclusion is also intuitively reasonable, with an algorithm proposing very small changes the correlation between $\mathbf{x}$ and $\mathbf{y}$ becomes very high and one can not expect to gain much by using also the rejected proposals.

Acknowledgements

We thank Håvard Rue for letting us use his computer code for the Gaussian Markov random field example.

References

Atchadé, Y. F. and Perron, F. (2005). Improving on the independent Metropolis-Hastings algorithm, *Statistica Sinica, **15**: 3-18* .

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma, *Australian Journal of Physics **18**: 119-133* .

Brooks, S. P., Morgan, B. J., Rideout, M. S. and Pack, S. E. (1997). Finite mixture models for proportions, *Biometrics **53**: 1097-1115* .

Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes, *Biometrica **83**: 81-94* .

Efron, B. (1981). Censored data and the bootstrap, *Journal of the American Statistical Association **76**: 312-319* .

Geier, C. (1992). Practical Markov chain Monte Carlo (with discussion), *Statistical Science **7**: 473-511* .

Hammersley, J. M. and Morton, K. W. (1956). A new Monte Carlo technique: Antithetic variates, *Proceedings of the Cambridge Philosophical Society **52**: 449-475* .

Han, X. L. and Green, P. J. (1992). Metropolis methods, Gaussian proposals, and antithetic variables *in* P. Barone, A. Frigessi and M. Piccioni (eds), *Stochastic Models, Statistical Methods and Algorithms in Image Analysis, number 74 in Lecture Notes in Statistics, Springer, Berlin pp. 142-164* .

Hastings, W. (1970). Monte Carlo sampling using Markov chains and their applications, *Biometrica, **57**, 97-109* .

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series (with discussion), *Journal of the American Statistical Association **82(400)**: 1032-1063* .

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*, Springer, Berlin.

Mira, A., Tenconi, P. and Bressanini, D. (2003). Variance reduction in MCMC, *Technical report no 29/2003, Department of Economics, University of Insubria, Italy* .

Peskun, P. (1973). Optimal Monte-Carlo sampling using Markov chains, *Biometrica, **60**, 607-612* .

Pinto, R. L. and Neal, R. M. (2001). Improving Markov chain Monte Carlo estimators by coupling to an approximating chain, *Technical Report No. 0101, Department of Statistics, University of Toronto* .

Priestly, M. B. (1981). Spectral analysis and time series, *Academic, London* .

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components, *Journal of the Royal Statistical Society, Series B **59**: 731-792* .

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms, *Annals of Applied Probability **7**: 110-120* .

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions, *Journal of the Royal Statistical Society. Series B **60**: 255-268* .

Rue, H. and Follestad, T. (2002). *GMRFlib: A C-library for fast and exact simulation of Gaussian Markov random fields*, Statistics Report No. 1, Department of Mathematical Sciences, Norwegian University of Science and Technology, Norway.

Rue, H. and Held, L. (2004). *Gaussian Markov Random Fields, Theory and Applications*, Chapman and Hall/CRC.

Tjelmeland, H. and Hegstad, B. K. (2001). Mode jumping proposals in MCMC, *Scandinavian Journal of Statistics,* ***28****, 205-223* .

Waagepetersen, R. and Sørensen, D. (2001). A tutorial on reversible jump MCMC with a view toward application in QTL-mapping, *International Statistical Review* , ***69****, 49-61* .

A Some proofs of the control variates in Section 4

A.1 Proof of the first control variate for the Reversible jump algorithm

Recall the algorithm described in Section 2.3. Here comes some more details related to the algorithm which is needed to prove the first control variate. The functional relation between $(\mathbf{z},\mathbf{u})$ and $(\mathbf{z}',\mathbf{u}')$ can in more detail be written as

$$(\mathbf{z}',\mathbf{u}') = \phi_{mm'}(\mathbf{z},\mathbf{u}) = (\phi_{1mm'}(\mathbf{z},\mathbf{u}),\phi_{2mm'}(\mathbf{z},\mathbf{u}))$$

and

$$(\mathbf{z},\mathbf{u}) = \phi^{-1}_{mm'}(\mathbf{z}',\mathbf{u}') = (\phi_{1m'm}(\mathbf{z}',\mathbf{u}'),\phi_{2m'm}(\mathbf{z}',\mathbf{u}')).$$

The deterministic mappings must be of dimension $\phi_{1mm'}:\mathbb{R}^{n_m+n_{mm'}}\to\mathbb{R}^{n_{m'}}$ and $\phi_{2mm'}:\mathbb{R}^{n_m+n_{mm'}}\to\mathbb{R}^{n_{m'm}}$. When considering a move from state $(m,\mathbf{z})$ to $(m',\mathbf{z}')=(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))$ and the reversed move $(m',\mathbf{z}')$ to $(m,\mathbf{z})=(m,\phi_{1m'm}(\mathbf{z}',\mathbf{u}'))$ the crucial so called dimensional matching condition $n_m+n_{mm'}=n_{m'}+n_{m'm}$ must be satisfied to obtain reversebility. It means that the vectors $(\mathbf{z},\mathbf{u})$ and $(\mathbf{z}',\mathbf{u}')$ must be of the same dimension. We write the Jacobi determinant in (9) as

$$J = \left|\frac{\phi_{mm'}(\mathbf{z},\mathbf{u})}{\partial\mathbf{z}\partial\mathbf{u}}\right|.$$

Remember that we have $\mathbf{x}=(m,\mathbf{z})$, $\mathbf{y}=(m',\mathbf{z}')$ and $\mathbf{z}'=\phi_{1mm'}(\mathbf{z},\mathbf{u})$. Thus, $g(\mathbf{x},\mathbf{y})$ defined in (15) can be written on the form

$$g(\mathbf{x},\mathbf{y}) = w_1(m,\mathbf{z},m',\mathbf{u})f(m,\mathbf{z}) + w_2(m,\mathbf{z},m',\mathbf{u})f(m',\phi_{1mm'}(\mathbf{z},\mathbf{u})). \tag{37}$$

This gives

$$\begin{aligned}E(g(\mathbf{x},\mathbf{y})) &= \sum_m\sum_{m'}\iint [w_1(m,\mathbf{z},m',\mathbf{u})f(m,\mathbf{z}) + w_2(m,\mathbf{z},m',\mathbf{u})f(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))]\\ &\qquad \pi(m,\mathbf{z})p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{u}\\ &= \sum_m\sum_{m'}\iint w_1(m,\mathbf{z},m',\mathbf{u})f(m,\mathbf{z})\pi(m,\mathbf{z})p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{u}\\ &+ \sum_m\sum_{m'}\iint w_2(m,\mathbf{z},m',\mathbf{u})f(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))\pi(m,\mathbf{z})p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{u}\end{aligned}$$

Performing the substitution $(\mathbf{z}',\mathbf{u}')=\phi_{mm'}(\mathbf{z},\mathbf{u})$ in the last term above, and thereafter interchanging the variable symbols $(m,\mathbf{z},\mathbf{u})$ and $(m',\mathbf{z}',\mathbf{u}')$ in the same term we get

$$\begin{aligned}E(g(\mathbf{x},\mathbf{y})) &= \sum_m\sum_{m'}\iint w_1(m,\mathbf{z},m',\mathbf{u})f(m,\mathbf{z})\pi(m,\mathbf{z})p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{u}\\ &+ \sum_m\sum_{m'}\iint w_2(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}),m,\phi_{2mm'}(\mathbf{z},\mathbf{u}))f(m,\mathbf{z})\pi(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))\\ &\qquad p_{m'm}q_{m'm}(\phi_{mm'}(\mathbf{z},\mathbf{u}))\left|\frac{\phi_{mm'}(\mathbf{z},\mathbf{u})}{\partial\mathbf{z}\partial\mathbf{u}}\right|\mathrm{d}\mathbf{z}\mathrm{d}\mathbf{u}.\end{aligned} \tag{38}$$

Thus, we get a sufficient condition for $E(g(\mathbf{x},\mathbf{y}))=0$ by setting the sum of the two integrands identical to zero. This gives

$$\begin{aligned}&w_1(m,\mathbf{z},m',\mathbf{u})\\ &= \frac{\pi(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))p_{m'm}q_{m'm}(\phi_{mm'}(\mathbf{z},\mathbf{u}))\left|\frac{\phi_{mm'}(\mathbf{z},\mathbf{u})}{\partial\mathbf{z}\partial\mathbf{u}}\right|}{\pi(m',\phi_{1mm'}(\mathbf{z},\mathbf{u}))p_{m'm}q_{m'm}(\phi_{mm'}(\mathbf{z},\mathbf{u}))\left|\frac{\phi_{mm'}(\mathbf{z},\mathbf{u})}{\partial\mathbf{z}\partial\mathbf{u}}\right| + \pi(m,\mathbf{z})p_{mm'}q_{mm'}(\mathbf{u}|\mathbf{z})}\end{aligned} \tag{39}$$

$$w_2(m,\mathbf{z},m',\mathbf{u}) = -w_1(m,\mathbf{z},m',\mathbf{u}) \tag{40}$$

and we get the first control variate for the reversible jump setting by substituting (39) and (40) into (37). Finally, we get (20) using the expression for the acceptance ratio $R(\mathbf{x},\mathbf{y})$.

A.2 Proof of theorem 1 for the Metropolis-Hastings algorithm

Let $g(\mathbf{x},\mathbf{y},\gamma) = w_1(\mathbf{x},\mathbf{y},\gamma)f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y},\gamma)f(\mathbf{y})$. To construct control variates we must be able to evaluate

$$\begin{aligned} E(g(\mathbf{x},\mathbf{y},\gamma)) = &\iint [w_1(\mathbf{x},\mathbf{y},0)f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y},0)f(\mathbf{y})]\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})(1-\alpha(\mathbf{y}|\mathbf{x}))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\ + &\iint [w_1(\mathbf{x},\mathbf{y},1)f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y},1)f(\mathbf{y})]\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \end{aligned} \tag{41}$$

Split each of the integrals in (41) in two integrals and change the order of integration in the integrals containing w_2 to get

$$\begin{aligned} E(g(\mathbf{x},\mathbf{y})) = &\iint w_1(\mathbf{x},\mathbf{y},0)f(\mathbf{x})\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})(1-\alpha(\mathbf{y}|\mathbf{x}))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\ + &\iint w_2(\mathbf{y},\mathbf{x},0)f(\mathbf{x})\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})(1-\alpha(\mathbf{x}|\mathbf{y}))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\ + &\iint w_1(\mathbf{x},\mathbf{y},1)f(\mathbf{x})\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\ + &\iint w_2(\mathbf{y},\mathbf{x},1)f(\mathbf{x})\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}. \end{aligned}$$

Thus, a sufficient condition for $E(g(\mathbf{x},\mathbf{y})) = 0$ is

$$\begin{aligned} w_1(\mathbf{x},\mathbf{y},0)\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})(1-\alpha(\mathbf{y}|\mathbf{x})) + w_2(\mathbf{y},\mathbf{x},0)\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})(1-\alpha(\mathbf{x}|\mathbf{y})) + \\ w_1(\mathbf{x},\mathbf{y},1)\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x}) + w_2(\mathbf{y},\mathbf{x},1)\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y}) = 0. \end{aligned} \tag{42}$$

There are six natural ways to fulfill this requirement and in the following we discuss each in turn. **(i)** Equating to zero the sum of the first two terms in (42) and setting $w_1(\mathbf{x},\mathbf{y},1) = w_2(\mathbf{x},\mathbf{y},1) = 0$. This gives $g(\mathbf{x},\mathbf{y},0) = w_1(\mathbf{x},\mathbf{y},0)f(\mathbf{x}) + w_2(\mathbf{x},\mathbf{y},0)f(\mathbf{y})$ and $g(\mathbf{x},\mathbf{y},1) = 0$, where the weight functions must fulfill

$$w_1(\mathbf{x},\mathbf{y},0)\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})(1-\alpha(\mathbf{y}|\mathbf{x})) = -w_2(\mathbf{y},\mathbf{x},0)\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})(1-\alpha(\mathbf{x}|\mathbf{y})). \tag{43}$$

However, whenever $\mathbf{y}$ is rejected, i.e. $\gamma = 0$, we must have $\alpha(\mathbf{y}|\mathbf{x}) < 1$ and thereby we also have $\alpha(\mathbf{x}|\mathbf{y}) = 1$. Thus, the right hand side of (43) is always zero when $\gamma = 0$ and thereby the same equation imply $w_1(\mathbf{x},\mathbf{y},0) = 0$ and we get $g(\mathbf{x},\mathbf{y},\gamma) \equiv 0$. So this choice is of no interest.
(ii) Equating to zero the sum of the first and third terms in (42) and setting $w_2(\mathbf{x},\mathbf{y},0) = w_2(\mathbf{x},\mathbf{y},1) = 0$. Equation (42) is then fulfilled by setting $w_1(\mathbf{x},\mathbf{y},0) = \alpha(\mathbf{y}|\mathbf{x})$ and $w_1(\mathbf{x},\mathbf{y},1) = 1-\alpha(\mathbf{y}|\mathbf{x})$ which gives the $g(\mathbf{x},\mathbf{y},\gamma)$ function in (23) in Theorem 1.
(iii) Equating to zero the sum of the first and last terms in (42) and setting $w_2(\mathbf{x},\mathbf{y},0) = w_1(\mathbf{x},\mathbf{y},1) = 0$. Equation (42) is then fulfilled by setting

$$w_1(\mathbf{x},\mathbf{y},0) = \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x},\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y}) + \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})[1-\alpha(\mathbf{y}|\mathbf{x})]},$$

$$w_2(\mathbf{x},\mathbf{y},1) = -\frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})[1-\alpha(\mathbf{x}|\mathbf{y})]}{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y}) + \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})[1-\alpha(\mathbf{y}|\mathbf{x})]},$$

which can be simplified to $w_1(\mathbf{x},\mathbf{y},0) = \alpha(\mathbf{y}|\mathbf{x})$ and $w_2(\mathbf{x},\mathbf{y},1) = -[1-\alpha(\mathbf{x}|\mathbf{y})]$. This gives the $g(\mathbf{x},\mathbf{y},\gamma)$ function in (24) in Theorem 1.
(iv) Equating to zero the sum of the second and third terms in (42) and setting $w_1(\mathbf{x},\mathbf{y},0) = w_2(\mathbf{x},\mathbf{y},1) = 0$. Equation (42) is then fulfilled by setting

$$w_1(\mathbf{x},\mathbf{y},1) = \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})[1-\alpha(\mathbf{x}|\mathbf{y})]}{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})[1-\alpha(\mathbf{x}|\mathbf{y})] + \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x})},$$

$$w_2(\mathbf{x}, \mathbf{y}, 0) = -\frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})[1-\alpha(\mathbf{x}|\mathbf{y})] + \pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x})},$$

which, corresponding to case (*iii*) above, can be simplified to $w_1(\mathbf{x}, \mathbf{y}, 1) = -[1 - \alpha(\mathbf{x}|\mathbf{y})]$ and $w_2(\mathbf{x}, \mathbf{y}, 0) = \alpha(\mathbf{y}|\mathbf{x})$. This gives the $g(\mathbf{x}, \mathbf{y}, \gamma)$ function in (25) in Theorem 1.

(v) Equating to zero the sum of the second and last terms in (42) and setting $w_1(\mathbf{x}, \mathbf{y}, 0) = w_1(\mathbf{x}, \mathbf{y}, 1) = 0$. Equation (42) is then fulfilled by setting $w_2(\mathbf{x}, \mathbf{y}, 0) = \alpha(\mathbf{y}|\mathbf{x})$ and $w_2(\mathbf{x}, \mathbf{y}, 1) = -[1 - \alpha(\mathbf{y}|\mathbf{x})]$ which gives the $g(\mathbf{x}, \mathbf{y}, \gamma)$ function in (26) in Theorem 1.

(vi) Finally equating to zero the sum of the third and last terms in (42) and setting $w_1(\mathbf{x}, \mathbf{y}, 0) = w_2(\mathbf{y}, \mathbf{x}, 0) = 0$. This gives $g(\mathbf{x}, \mathbf{y}, 0) = 0$ and $g(\mathbf{x}, \mathbf{y}, 1) = w_1(\mathbf{x}, \mathbf{y}, 1)f(\mathbf{x}) + w_2(\mathbf{x}, \mathbf{y}, 1)f(\mathbf{y})$, where the weight functions must fulfill

$$w_1(\mathbf{x}, \mathbf{y}, 1)\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})\alpha(\mathbf{y}|\mathbf{x}) = -w_2(\mathbf{y}, \mathbf{x}, 1)\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{x}|\mathbf{y}). \tag{44}$$

Equation (44) can be simplified to $w_2(\mathbf{x}, \mathbf{y}, 1) = -w_1(\mathbf{y}, \mathbf{x}, 1)$, which gives that $g(\mathbf{x}, \mathbf{y}, 1) = w_1(\mathbf{x}, \mathbf{y}, 1)f(\mathbf{x}) - w_1(\mathbf{y}, \mathbf{x}, 1)f(\mathbf{y})$. Inserting this in (22) we see that the resulting control variate is defined by a telescope sum. So, as for case (*i*), this case is of no interest.