Efficient Sampling Using Metropolis Algorithms: Applications of Optimal Scaling Results

Mylène Bédard [*]

Abstract

We recently considered the optimal scaling problem of Metropolis algorithms for multidimensional target distributions with non-IID components. The results that were proven have wide applications and the aim of this paper is to show how practitioners can take advantage of them. In particular, we illustrate with several examples the case where the asymptotically optimal acceptance rate is the usual 0.234, and also the latest developments where smaller acceptance rates should be adopted for optimal sampling from the target distributions involved. We study the impact of the proposal scaling on the performance of the algorithm, and finally perform simulation studies exploring the efficiency of the algorithm when sampling from some popular statistical models.

Keywords: Asymptotically optimal acceptance rate, diffusion, hierarchical model, non-identically distributed components, target distribution, speed measure

1 Introduction

Metropolis-Hastings algorithms are a popular class of MCMC methods that are used to generate data from basically any given probability distribution (Metropolis et al. 1953, Hastings 1970). The underlying principle is to build a Markov chain $X(0), X(1), \ldots$ having the probability distribution of interest as its unique stationary distribution. This distribution of interest is referred to as the target distribution and its density is denoted by $\pi(\cdot)$, which we assume to be defined on a continuous state space. Necessary to the construction of the Markov chain is the selection of a proposal density $q(\cdot\,|\cdot)$, which is used to generate potential moves for the chain. The algorithm is then performed by applying the following steps. Given that the time-t position of the Markov chain is $X(t)$, we generate a new state $Y(t+1)$ from the proposal distribution, which is accepted with probability $\alpha(X(t), Y(t+1)) = 1 \wedge (\pi(Y(t+1))\, q(X(t)\,|Y(t+1))\,/\pi(X(t))\, q(Y(t+1)\,|X(t)))$. If the suggested move is accepted, then the chain moves from $X(t)$ to $Y(t+1)$; otherwise, $X(t+1) = X(t)$.

The acceptance probability $\alpha(X(t), Y(t+1))$ is chosen such that the Markov chain is reversible with respect to the target density $\pi(\cdot)$, ensuring that $\pi(\cdot)$ is stationary for the

[*] Department of Statistics, University of Warwick, Coventry, U.K., CV4 7AL.
Email: M.Bedard@warwick.ac.uk Original date: April 2006, revised January 2007

chain. To compel the convergence of the Markov chain to its stationary distribution, one might choose virtually any proposal density engendering a Markov chain that is irreducible and aperiodic. A popular choice for the proposal distribution is the normal family centered at the current state $X(t)$ and scaled according to σ^2. For an algorithm with such a proposal distribution to be efficient, one must however carefully select the variance term. Indeed, small values for σ^2 cause the chain to jump to nearby states, resulting in a lengthy exploration of the state space. On the other hand, large scaling values often suggest states located in low target density regions, encouraging the chain to refuse these moves and stand still for long periods of time.

In this paper, we consider the optimal scaling problem of Metropolis algorithms (Roberts et al. 1997, Breyers & Roberts 2000, Roberts & Rosenthal 2001, Neal & Roberts 2004) for targets with independent but not identically distributed components, and study how well existing asymptotic results serve the practical side. In particular, we consider the target model introduced recently in Bédard (2006a,b) and present various examples illustrating how the optimal value for the proposal variance and acceptance rate can be determined using the theorems proved in these papers. This demonstrates that, although of an asymptotic nature, the results can be used to facilitate the tuning of algorithms in finite-dimensional problems.

In Section 2 we shall introduce the target and proposal distributions considered, and briefly discuss measures of efficiency for the algorithm. The optimal scaling results that shall be presented in Section 3 are divided in three cases: the asymptotically optimal acceptance rate (AOAR) is the usual 0.234; the AOAR is smaller than 0.234 and its exact value is solved on a case per case basis; finally the AOAR is 0, in which case the efficiency of the algorithm cannot be optimized. In that situation, we must resort to inhomogeneous proposal distributions. It is worth mentioning that the Metropolis algorithm considered in the present paper cannot admit an AOAR greater than 0.234 since the proposal distribution selected is best suited for target distributions with independent and identically distributed (IID) components, for which the AOAR is the well-known 0.234. In each of the three parts, the theoretical results are first discussed and examples illustrating their application are then presented. This section is concluded with a discussion about the robustness of these results to the violation of the independence assumption among the target components. Section 4 aims to present simulation studies to investigate the efficiency of the algorithm applied to more complicated and widely used statistical models; specifically, we consider the normal hierarchical model, the variance components model and the gamma-gamma hierarchical model.

2 The Model

To illustrate the purpose of the following sections, suppose we wish to sample from

$$\pi\left(x_1,\ldots,x_d\right) \propto x_1^4 \exp\left(-x_1\right)\ x_2^4 \exp\left(-x_2\right) \prod_{i=3}^{d} x_i^4 \exp\left(-x_i/5\sqrt{d}\right). \tag{1}$$

For a relatively low-dimensional target density of this sort, say $d = 10$, the density of the last 8 components is spread out over $(0,\infty)$ while that of the first two have their mass concentrated within a much narrower interval of the state space. Choosing a proper variance for the proposal distribution is thus not an easy task: the last 8 components require a large proposal variance for appropriate exploration of their state space, but the selection of too large a value would result in frequent rejection of the proposed moves by the variables X_1 and X_2. A compromise between these requirements then becomes necessary.

An obvious alternative to the block updating algorithm would be to use a variable-at-a-time method for sampling from this target; however, optimal scaling results for such methods have yet to be derived. Although existing asymptotic results work well in low dimensions, their demonstrations rely heavily on laws of large numbers and results in finite dimensions have only been derived for normal target distributions (Sherlock, 2006).

For this example, the optimal proposal variance is close to $\sigma^2 = 61$ for any dimension d, resulting in an AOAR lying around 0.098, as shall be seen in Example 5. In fact, tuning the algorithm to accept a proportion of 23.4% of the proposed moves would reduce the efficiency of the algorithm (i.e. the mixing rate of the chain or the speed at which the algorithm converges to its stationary distribution) by about 20%; it then becomes important to determine the exact optimal scaling value, rather than blindly apply the 0.234 rule. Before discussing the optimal scaling results, we however start by introducing the general model for the target density.

2.1 The Target Distribution

Suppose we want to generate a sample from the d-dimensional target density

$$\pi\left(d, \mathbf{x}^{(d)}\right) = \prod_{j=1}^{d} \theta_j\left(d\right) f\left(\theta_j\left(d\right) x_j\right), \tag{2}$$

where f is a smooth density and $\theta_j^{-2}\left(d\right)$, $j = 1, \ldots, d$ are referred to as the scaling terms of the target distribution, which can be any function of d for which $\lim_{d\to\infty} \theta_j\left(d\right)$ exists. This model constitutes a natural extension of the d-dimensional density formed of IID components considered in the literature (Roberts et al. 1997, Roberts & Rosenthal 2001, Neal & Roberts 2004) and used to investigate optimal acceptance rates.

In order to establish optimal scaling results for the proposal distribution, we need to study the limiting behavior of the algorithm as $d \to \infty$. The results presented subsequently are then asymptotic ones, valid for sampling from infinite-dimensional versions of (2). Consequently, we need to consider a model where there is an infinite number of scaling terms. As d increases, some of the scaling terms will be repeated an infinite number of times; we assume that there are $0 < m < \infty$ such different terms. Other terms will not be replicated as $d \to \infty$; we suppose that there are $n < \infty$ of them. We let the first $n + m$ components of the vector $\mathbf{\Theta}^{-2}\left(d\right)$ consist of the n non-replicated scaling terms, followed by the m different scaling terms that shall be replicated. We further assume that the first n and next m components are respectively arranged according to an asymptotically increasing order.

Although the components $\theta_{n+1}\left(d\right), \cdots, \theta_{n+m}\left(d\right)$ appear infinitely often as $d \to \infty$, they might however not be replicated the same number of times for finite values of d. To determine the proportion of $\mathbf{\Theta}^{-2}\left(d\right)$ occupied by the i-th group ($i \in \{1, \ldots, m\}$), we define the cardinality functions

$$c\left(i,d\right)=\#\left\{j\in\left\{1,\ldots,d\right\};\theta_j\left(d\right)=\theta_{n+i}\left(d\right)\right\},\quad i=1,\ldots,m. \tag{3}$$

The following example should help clarify the notation just introduced.

Example 1. Consider a d-dimensional target density as in (2) with scaling terms $1/\sqrt{d}$, $4/\sqrt{d}$, 10 and the other ones equally divided among $2\sqrt{d}$ and $\left(d+1\right)/2$. As $d\to\infty$, the last two scaling terms are replicated so $n=3$, $m=2$ and $\mathbf{\Theta}^{-2}\left(d\right)=\Big(1/\sqrt{d},4/\sqrt{d},10,2\sqrt{d},\left(d+1\right)/2,2\sqrt{d},\left(d+1\right)/2,\ldots\Big)$. The cardinality functions for the scaling terms appearing infinitely often in the limit are

$$c\left(1,d\right)=\#\left\{j\in\left\{1,\ldots,d\right\};\theta_j\left(d\right)=\left(2\sqrt{d}\right)^{-1/2}\right\}=\left\lceil\frac{d-3}{2}\right\rceil$$

and $c\left(2,d\right)=\lfloor\left(d-3\right)/2\rfloor$, where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the ceiling and integer part functions respectively. Note however that such rigor is superfluous for applying the results and it is enough to affirm that both cardinality functions grow according to $d/2$.

The model just presented for the target is not the most general form under which the conclusions of the theorems are satisfied. Indeed, it would be sufficient to assume that scaling terms belonging to a common group $i\in\left\{1,\ldots,m\right\}$ are of the same order. For more details, we refer the reader to Bédard (2006a,b).

2.2 The Proposal Distribution and its Scaling

The proposal distribution we consider for sampling from the target density $\pi\left(\cdot\right)$ is such that $\mathbf{Y}^{(d)}\left(t+1\right)\sim N\left(\mathbf{X}^{(d)}\left(t\right),\sigma^2\left(d\right)I_d\right)$, where I_d is the d-dimensional identity matrix.

There exist two factors determining the form of the proposal variance as a function of d: the asymptotically smallest scaling term and the fact that some scaling terms appear infinitely often in the limit. If the asymptotically smallest scaling term were ignored, the proposed moves would possibly be too large for the corresponding component, resulting in high rejection rates and slow convergence of the algorithm. In addition, for large values of d, the algorithm proposes many moves in a single step; since these moves are generated independently from each others, the odds of proposing an improbable move for the algorithm augment as $d\to\infty$. In this case, a drop in the acceptance rate can be overturned by letting $\sigma^2\left(d\right)$ be a decreasing function of the dimension. Combining these two constraints, the proposal variance optimizing the mixing of the algorithm can be shown to take the form $\sigma^2\left(d\right)=\ell^2\varphi\left(d\right)$, with ℓ some positive constant and $\varphi\left(d\right)$ the largest order function such that

$$\lim_{d\to\infty}\theta_1^2\left(d\right)\varphi\left(d\right)<\infty\ \text{ and }\ \lim_{d\to\infty}c\left(i,d\right)\theta_{n+i}^2\left(d\right)\varphi\left(d\right)<\infty,\quad\text{for } i=1,\ldots,m. \tag{4}$$

Having determined $\varphi\left(d\right)$, we now know how the proposal variance varies as a function of d; the only unknown term in $\sigma^2\left(d\right)$ being ℓ^2, our goal has thus evolved in determining the value of ℓ for which the chain will converge the fastest to its stationary distribution.

Example 2. We now determine the optimal form for the proposal variance of the Metropolis algorithm in Example 1. According to (4), we have three limits to verify: the first one involves $\theta_1^{-2}(d)$, which is also the asymptotically smallest scaling term in the present case; the largest $\varphi(d)$ satisfying the finite property for $\lim_{d\to\infty}\sqrt{d}\varphi(d)$ is $1/\sqrt{d}$. For the second and third limits, we have $\lim_{d\to\infty}(d-3)\varphi(d)/4\sqrt{d}$ and $\lim_{d\to\infty}(d-3)\varphi(d)/(d+1)$; the largest $\varphi(d)$ satisfying the finite property is then $1/\sqrt{d}$ and 1 respectively. The function of largest order satisfying the constraint that all three limits be finite is $\varphi(d)=1/\sqrt{d}$.

2.3 Efficiency of the Algorithm

In order to optimize the mixing of our Metropolis algorithm, it would be convenient to determine criteria for measuring efficiency. For infinite-dimensional algorithms, this problem is relatively simple. Indeed, as $d\to\infty$, it can be shown that the path followed by any of the last $d-n$ components of an appropriately rescaled version of the Metropolis algorithm converges to a Langevin diffusion process with speed measure $\upsilon(\ell)$ (see Bédard 2006a,b). Since $\upsilon(\ell)$ is the only quantity depending on the proposal variance (i.e. on ℓ) in the limit, optimizing the mixing rate of any of the components $X_{n+1}^{(d)},\ldots,X_d^{(d)}$ is equivalent to choosing the diffusion process that goes the fastest (so the value of ℓ for which $\upsilon(\ell)$ is optimized). The components $X_1^{(d)},\ldots,X_n^{(d)}$ do not necessarily exhibit the same limiting behavior, and could converge to a discrete-time limiting process as $d\to\infty$. In this case, efficiency criteria are not all equivalent and attempting to select the proposal variance optimizing the speed at which $X_1^{(d)},\ldots,X_n^{(d)}$ converge to their stationary distribution could then result in conclusions that are specific to our choice of efficiency measure.

When it comes to the selection of an efficiency criterion for optimizing the speed of convergence of the algorithm to its stationary distribution, we thus face a dilemma. To be confident that the d-dimensional Markov chain has reached stationarity, we must be certain that $X_1^{(d)},\ldots,X_d^{(d)}$ have all reached stationarity. From Bédard (2006b) we know that the target components possessing a diffusion limit mix in $O(d)$ iterations, while those converging to a discrete-time process mix according to $O(1)$. It thus makes sense to rely on the quantity $\upsilon(\ell)$ to achieve the fastest convergence of the algorithm to its stationary distribution.

We now need to select a measure of efficiency that shall be used when dealing with finite-dimensional algorithms. In the subsequent examples, this shall be used to compare the efficiency of our finite-dimensional algorithms with the asymptotic efficiency obtained from the limiting process. When using MCMC methods, researchers and practitioners are most often interested in estimating the expectation of some function $g(\cdot)$. The natural way to estimate the function of interest is to compute the expectation of $g(\cdot)$ with respect to $\pi(\cdot)$. In practice, this is approximated by the sample average $\sum_{i=1}^{k} g\left(\mathbf{X}^{(d)}(i)\right)/k$, where $\mathbf{X}^{(d)}(1),\cdots,\mathbf{X}^{(d)}(k)$ is the generated Markov chain. Minimizing the asymptotic variance of $g(\cdot)$ would then be a good way to optimize efficiency. Since we do not want to lose generality by specifying such a quantity of interest, we instead choose the less specific first order efficiency criterion, as used by Roberts & Rosenthal (1998) and Pasarica & Gelman (2003). This measures the average squared jumping distance of the algorithm and is defined by

$$\begin{aligned} FOE_{n+1} &= \mathrm{E}\left[\left(X_{n+1}^{(d)}(t+1) - X_{n+1}^{(d)}(t)\right)^2\right] \\ &= \int\int \left(y_{n+1}^{(d)}(t+1) - x_{n+1}^{(d)}(t)\right)^2 q\left(\mathbf{y}^{(d)}(t+1)\left|\mathbf{x}^{(d)}(t)\right.\right) \\ &\qquad \times \alpha\left(\mathbf{x}^{(d)}(t), \mathbf{y}^{(d)}(t+1)\right) \pi\left(d, \mathbf{x}^{(d)}(t)\right) d\mathbf{y}^{(d)}(t+1)\, d\mathbf{x}^{(d)}(t)\,. \end{aligned}$$

In our case, we choose to base this criterion on the path followed by the $n+1$-st component of the algorithm (we could also have chosen the path of any of the last $d-n$ components).

Before concluding this section, we briefly discuss how the asymptotic efficiency measure $\upsilon(\ell)$ is related to the various efficiency criteria in finite dimensions. In fact, since the speed measure is the only sensible efficiency criterion for a diffusion, it can be shown that almost any efficiency criterion selected in finite dimensions becomes proportional to the speed measure as $d \to \infty$ (see Roberts & Rosenthal 2001). This means that, no matter if we use the asymptotic variance of a function of interest $g(\cdot)$, the mean squared error (MSE) of its Monte Carlo estimator, or simply FOE as our preferred measure of efficiency, the effect of selecting a particular efficiency criterion shall vanish as d gets larger. The only exceptions consist in efficiency measures based solely on the components converging to a discrete-time limit, or where the weight of these components does not vanish as $d \to \infty$. This is fortunately not a big issue, as such efficiency criteria are typically not of interest when using the assumed target model. If they were, then one could easily use an alternative product form target model to perform the analysis.

3 Optimal Scaling Results

3.1 The Familiar Asymptotic Behavior

Consider a Metropolis algorithm with proposal distribution $\mathbf{Y}^{(d)} \sim N\left(\mathbf{x}^{(d)}, \ell^2\varphi(d) I_d\right)$, where $\varphi(d)$ satisfies (4). Suppose this algorithm is used to sample from a target density as in (2) and supporting the model described in Section 2.1.

It is now an established fact that 0.234 is the AOAR for target distributions with IID components (Roberts et al. 1997). Roberts & Rosenthal (2001) also showed that the same conclusion applies for $\pi(\cdot)$ as in (2) but with $\theta_j(d)$'s independent of d. It is thus natural to wonder how big a discrepancy between the scaling terms is tolerated in order not to violate this established asymptotic behavior. It turns out that if

$$\lim_{d\to\infty} \frac{\theta_1^2(d)}{\sum_{j=1}^{d} \theta_j^2(d)} = 0, \tag{5}$$

then the optimal acceptance rate can be shown to converge towards 0.234. The optimal value $\hat{\ell}$ maximizes the equation

$$\upsilon(\ell) = 2\ell^2 \Phi\left(-\frac{\ell\sqrt{E_R}}{2}\right) = \ell^2 a(\ell)\,, \tag{6}$$

where $\Phi(z) = \int_{-\infty}^{z} \exp\left(-x^2/2\right)/\sqrt{2\pi}\,dx$, $a(\ell) = 2\Phi\left(-\ell\sqrt{E_R}/2\right)$ is the asymptotic acceptance rate of the algorithm, and

$$E_R = \lim_{d\to\infty} \sum_{i=1}^{m} c(i,d)\,\theta_{n+i}^2(d)\,\varphi(d)\,\mathrm{E}\left[\left(\frac{f'(X)}{f(X)}\right)^2\right], \qquad (7)$$

with $c(i,d)$ as in (3) and f as in (2). We then obtain $\hat{\ell} = 2.38/\sqrt{E_R}$, from where $a(\hat{\ell}) = 0.234$. For a formal version of the theorem, see Bédard (2006a).

This result provides valuable guidelines for practitioners. It reveals that when the target distribution has no scaling term that is significantly smaller than the others, the asymptotic acceptance rate optimizing the efficiency of the chain is 0.234. Alternatively, setting ℓ equal to $2.38/\sqrt{E_R}$ leads to greatest efficiency of the algorithm. In some situations, finding $\hat{\ell}$ will be easier while in others, tuning the algorithm according to the AOAR will be a more convenient way to achieve optimal efficiency. In the present case, since the AOAR does not depend on the target density, it is simpler in practice to monitor the acceptance rate and to tune it to be about 0.234.

While the AOAR is independent of the target distribution, $\hat{\ell}$ varies inversely proportionally to E_R. Two different factors influence this quantity: the function $f(\cdot)$ in (2) and the $\theta_j^{-2}(d)$'s. The latter can have an effect through their size or the proportion of the vector $\mathbf{\Theta}^{-2}(d)$ they occupy. Specifically, suppose that $c(i,d)\,\theta_{n+i}^2(d)$ is $O\left(\varphi^{-1}(d)\right)$ for some $i \in \{1,\ldots,m\}$, implying that the i-th group contributes to augment the value of E_R. The amount by which E_R increases is then proportional to the size of the $\theta_j(d)$'s, but inversely proportional to the quantity of scaling terms included in the group. The following examples shall clarify these concepts.

Example 3. Consider a d-dimensional target with independent, normally distributed components; we suppose that half of them have a variance of 1 and the other half a variance of $2d$. Applying (4), the proposal variance takes the form $\sigma^2(d) = \ell^2/d$ and Condition (5) is verified by computing $\lim_{d\to\infty}(d/2 + 1/4)^{-1} = 0$ (in fact this is trivial since $n = 0$). We can thus optimize the efficiency of the algorithm by setting the acceptance rate to be close to 0.234; equivalently, since $E_R = \lim_{d\to\infty}(1/2 + 1/4d) = 1/2$, we find $\hat{\ell} = 3.366$. Note that since only half of the components affect the accept/reject ratio in the limit, $\hat{\ell}$ is now larger than the baseline 2.38, which is the value for $\hat{\ell}$ when all components are IID standard normal.

In all of the examples, the starting values have been generated from the target distribution, and FOE_j has been estimated by $\sum_{i=1}^{k}\left(X_j^{(d)}(i) - X_j^{(d)}(i-1)\right)^2/k$, where $X_j^{(d)}(i)$ is the state of the j-th component at time i and k is the number of iterations performed. Furthermore, the acceptance rate has been estimated by the proportion of accepted moves in the algorithm.

The left graph in Figure 1 presents the relation between first order efficiency and ℓ^2. The dotted curve has been obtained by performing 100,000 iterations of the Metropolis algorithm with $d = 100$ for various values of ℓ (so for each of the 50 points on the curve). As expected the maximum is located very close to $\hat{\ell}^2 = 11.33$, and the simulations also agree with the theoretical curve (solid line) of $\upsilon(\ell)$ in (6) versus ℓ^2. For the second graph, we run the algorithm with various values of ℓ and plot FOE_1 as a function of the acceptance rate of the

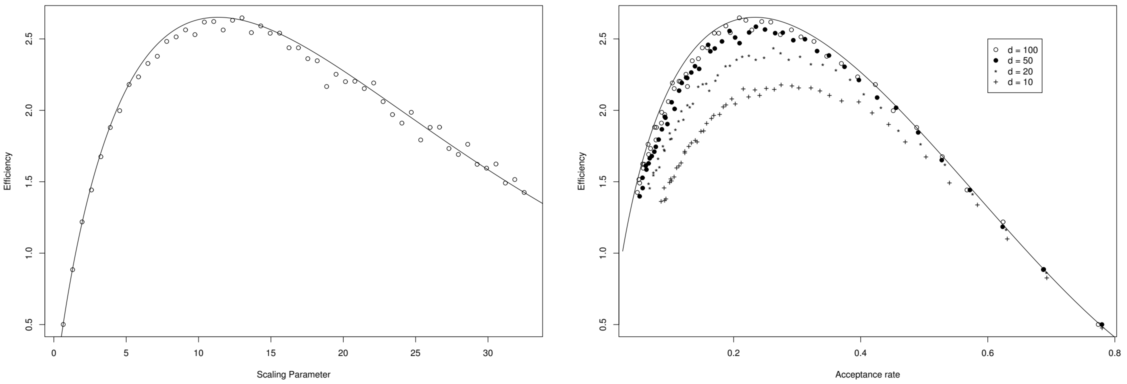

Figure 1: Left graph: efficiency of X_1 versus ℓ^2; the dotted line is the result of simulations with $d = 100$. Right graph: efficiency of X_1 versus the acceptance rate; the dotted lines come from simulations in dimensions 10, 20, 50 and 100. In both graphs, the solid line represents the theoretical curve $\upsilon(\ell)$.

algorithm. That is, each point in a given curve is the result of a simulation with a particular value for ℓ (for both graphs we chose $\ell^2 = (1\colon 50) * 0.65$), and each curve is the result of simulations in different dimensions ($d = 10, 20, 50, 100$). We again performed 100,000 iterations of the algorithm, obtaining four curves which illustrate the fact that the optimal acceptance rate converges very rapidly to its asymptotic counterpart. The theoretical curve of $\upsilon(\ell)$ versus $a(\ell)$ is represented by the solid line.

We note that efficiency is a relative measure in our case. For instance, choosing an acceptance rate around 0.05 or 0.5, would necessitate to run the chain 1.5 times as long to obtain the same precision for a particular estimate.

Although MCMC methods are not required to sample from normal distributions, this type of target is widely used in the literature to investigate the optimal scaling problem, and thus allows us to see how our results compare to others. Note however that we could also have used any smooth density $f(\cdot)$ with any $\mathbf{\Theta}^{-2}(d)$ satisfying Condition (5). While the convergence might get slightly slower as $f(\cdot)$ gets further from normality, we generally observe curves similar to those of Figure 1, as well as a rapid convergence to the asymptotic curve. The following example presents a particular situation where the convergence of some components towards the AOAR is extremely slow, a phenomenon due to the form of $\mathbf{\Theta}^{-2}(d)$.

Example 4. Consider a multivariate normal target with independent components and variances $\mathbf{\Theta}^{-2}(d) = (d^{-0.75}, 1, 1, 1, \ldots)$. We find $\sigma^2(d) = \ell^2/d$ and $\lim_{d\to\infty} d^{0.75}\left(d^{0.75} + (d-1)\right)^{-1}$ $= 0$, implying that Condition (5) is verified. The quantity E_R being equal to 1, the optimal value for ℓ is then the baseline 2.38.

The difficulty of this case resides in the size of $\theta_1^{-2}(d)$, which is somewhat smaller than the other terms and is not significant as $d \to \infty$. As a consequence, the dimension of the target distribution must be quite large before the asymptotics kick in. In small dimensions, the optimal acceptance rate is thus closer to the case where there exist significantly small

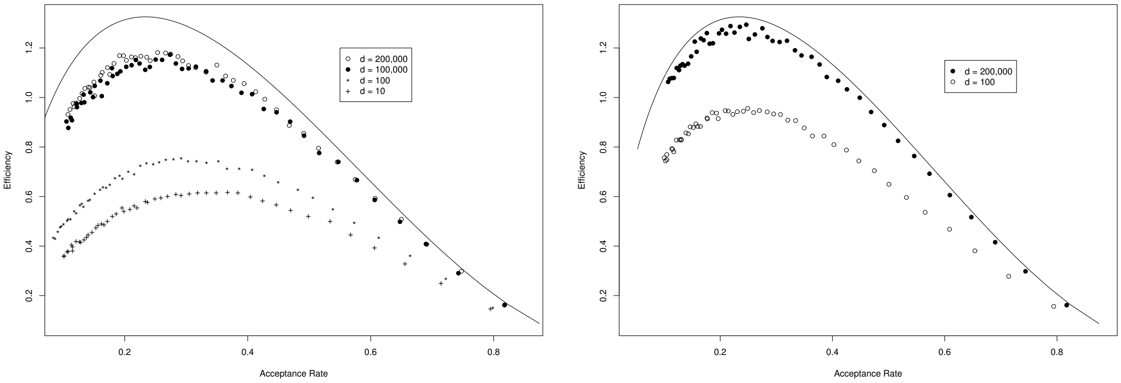

Figure 2: Left graph: efficiency of X_1 versus the acceptance rate; the dotted curves are the results of simulations in dimensions 10, 100, 100,000 and 200,000. Right graph: efficiency of X_2 versus the acceptance rate; the dotted curves come from simulations in dimensions 100 and 200,000. In both graphs, the solid line represents the theoretical curve $\upsilon(\ell)$.

scaling terms, which shall be studied in Section 3.2.

Both graphs in Figure 2 present the relationship between FOE and the acceptance rate of the algorithm, and have been obtained by using the method explained in Example 3. In particular, we performed simulations in different dimensions for various values of ℓ, but this time we used $\ell^2 = 30 + (1\colon 50) * 3.5$. The right graph demonstrates that even in small dimensions, the first order efficiency criterion based on any of the last $d-1$ components is very close to 0.234 so as not to make much difference in practice. When first order efficiency is based on X_1 however, setting $d = 100$ yields an optimal acceptance rate around 0.3 and the dimensions must be raised as high as 100,000 to get an optimal acceptance rate reasonably close to the asymptotic one (left graph). Relying on X_1 would then falsely suggest a higher optimal acceptance rate, as explained in Section 2.3.

3.2 A Reduction of the Asymptotically Optimal Acceptance Rate

In the presence of a finite number of significantly small scaling terms, choosing a correct proposal variance is a slightly more delicate task. We can think for instance of the density in (1), which seems to promote contradictory characteristics when it comes to the selection of an efficient proposal variance. In that example, the components X_1 and X_2 are said to rule the algorithm since they ensure that the proposal variance is not too big as a function of d. When dealing with such target densities, we realize that Condition (5) is violated, and thus

$$\lim_{d\to\infty} \frac{\theta_1^2(d)}{\sum_{j=1}^{d}\theta_j^2(d)} > 0. \tag{8}$$

The existence of scaling terms ruling the algorithm leads to one of two situations: if they are of extremely small order compared to the other scaling terms, the algorithm is inefficient; on the other hand, a reasonable difference of size among them can be handled. In particular, an AOAR exists if there is at least one $i \in \{1, \ldots, m\}$ satisfying

$$\lim_{d \to \infty} c\left(i, d\right) \theta_{n+i}^{2}\left(d\right) \varphi\left(d\right) > 0. \tag{9}$$

In words this condition requires that if we were ignoring $X_1, \ldots, X_n$, the form selected for $\varphi\left(d\right)$ based on only the last $d-n$ components would remain intact. This therefore ensures that the first n components are not solely affecting the selection of $\varphi\left(d\right)$.

In this setting, the optimal value for ℓ maximizes

$$\upsilon\left(\ell\right) = 2\ell^2 \mathrm{E}_{\mathbf{Y}^{(b)}, \mathbf{X}^{(b)}}\left[\Phi\left(\frac{\sum_{j=1}^{b} \log\left(f\left(\theta_j Y_j\right) / f\left(\theta_j X_j\right)\right) - \ell^2 E_R/2}{\sqrt{\ell^2 E_R}}\right)\right] = \ell^2 a\left(\ell\right), \tag{10}$$

where $b = \max\left(j \in \{1, \ldots, n\}; 0 < \lim_{d\to\infty} \theta_j\left(d\right) \theta_1^{-1}\left(d\right) < \infty\right)$ is the number of non-replicated components having a $O\left(\theta_1^{-2}\left(d\right)\right)$ scaling term (E_R and $c\left(i, d\right)$ are as in (7) and (3) respectively).

The equation $\upsilon\left(\ell\right)$ is affected by the b components which, even when dealing with very high-dimensional targets, have a scaling term small enough so as to always affect the accept-reject ratio of the algorithm. The AOAR is unfortunately not independent of the target distribution anymore, and varies according to the choice of $f\left(\cdot\right)$ in (2) and $\mathbf{\Theta}^{-2}\left(d\right)$. It is then simpler to optimize the efficiency of the algorithm by numerically determining $\hat{\ell}$ from (10) rather than monitoring the acceptance rate, since in any case finding the AOAR implies solving for $\hat{\ell}$. As before, $\hat{\ell}$ is inversely proportional to E_R but now also depends on $X_1, \ldots, X_b$, causing the algorithm to reject a greater proportion of moves. This provokes a reduction of $a(\hat{\ell})$, resulting in AOARs that are smaller than 0.234 and vary inversely proportionally to b. Correspondingly, the greater b is, the smaller $\hat{\ell}$ will be. For more details on this case, see Bédard (2006b).

The following example illustrates how to solve for the appropriate $\hat{\ell}$ and AOAR using (10). It presents a situation where tuning the acceptance rate to 0.234 results in an algorithm whose performance is substantially less than when using the correct AOAR.

Example 5. Consider the d-dimensional target density introduced in (1). Consistent with the notation of Sections 2.1 and 2.2, we find $\mathbf{\Theta}^{-2}\left(d\right) = \left(1, 1, 25d, 25d, 25d, \ldots\right)$ and so $\sigma^2\left(d\right) = \ell^2$. We remark that the first two scaling terms are significantly smaller than the balance since we have $\lim_{d\to\infty}\left(2 + \left(d-2\right)/25d\right)^{-1} = 25/51$. Even though $\theta_1^{-2}\left(d\right)$ and $\theta_2^{-2}\left(d\right)$ are substantially smaller than the other components, they still share the responsibility of selecting $\varphi\left(d\right)$ with the other $d-2$ components since $\lim_{d\to\infty} c\left(1, d\right) \theta_3^2\left(d\right) \varphi\left(d\right) = \lim_{d\to\infty}\left(d-2\right)/25d = 1/25$. Conditions (8) and (9) being satisfied, we thus use (10) to optimize the efficiency of the algorithm.

Generating values for $X_1, \ldots, X_b$ is generally complicated, from where the importance of using MCMC methods. In these cases, numerical integration can be used to evaluate the expectation in (10), as b is usually not very high-dimensional. In the present example, since values for X_1 and X_2 are easily generated, we turn to regular Monte Carlo methods to optimize $\upsilon\left(\ell\right)$. We first generate 500,000 values for the pairs $\left(X_1, Y_1\right)$ and $\left(X_2, Y_2\right)$ (recall that X_i is a target component and that $Y_i \sim N\left(X_i, \sigma^2\left(d\right) = \ell^2\right)$, $i = 1, 2$). We then replace the expectation in (10) by the sample average, using $E_R = 1/75$. We repeat this

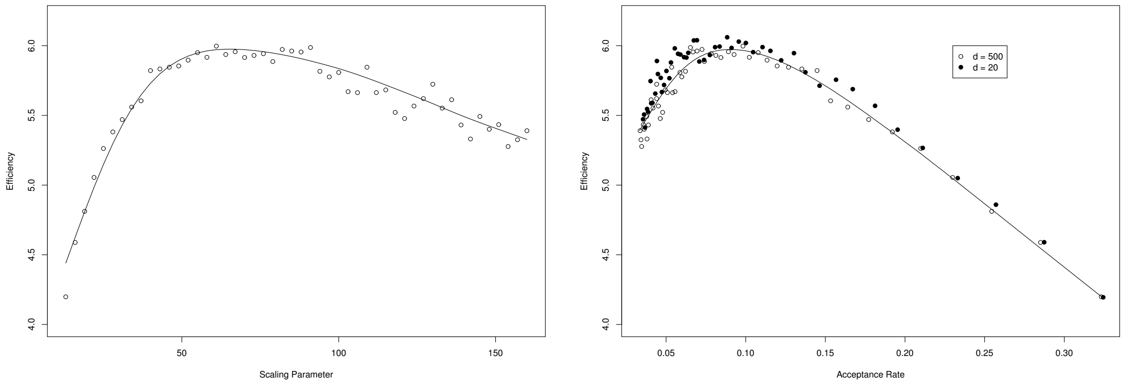

Figure 3: Left graph: efficiency of X_3 versus ℓ^2; the dotted curve represents the results of simulations with $d = 500$. Right graph: efficiency of X_3 versus the acceptance rate; the results of simulations with $d = 20$ and 500 are pictured by the dotted curves. In both cases, the theoretical curve $\upsilon(\ell)$ is depicted (solid line).

procedure for each value in the vector $\ell^2 = 10 + (1{:}\ 50) * 3$ and obtain the vector $\upsilon(\ell)$. A scan of this vector allows us to determine the maximum value for the speed measure $\upsilon(\ell)$ in (10), and thus the corresponding optimal scaling value $\hat{\ell}^2$; we find $\hat{\ell}^2 = 61$ and $a\left(\hat{\ell}\right) = \upsilon\left(\hat{\ell}\right)/\hat{\ell}^2 = 0.0980717$. Note that the term $E_R = 1/75$ causes an increase of $\hat{\ell}$, but the components X_1 and X_2 ($b = 2$) act in the opposite direction. This is why $\hat{\ell}^2 < 424.83$, which would be the optimal value for ℓ if X_1 and X_2 were ignored. The simulations performed also allow us to draw the theoretical efficiency curves which appear in Figure 3.

Figure 3 also illustrates the result of 500,000 iterations (again, for every point in the curves) of a Metropolis algorithm in dimensions 500 for the left graph and in dimensions 20 and 500 for the right one. On both graphs, obtained as in Example 3 but with $\ell^2 = 10 + (1{:}\ 50) * 3$, the maximum occurs close to the theoretical values mentioned previously. We note that the AOAR is now quite far from 0.234, and that tuning the proposal scaling so as to produce this acceptance rate would contribute to considerably lessen the performance of the method. In particular, this would generate a drop of at least 20% in the efficiency of the algorithm.

In Example 3, it did not matter which of the d components was selected to compute first order efficiency, as all of them would have yielded similar efficiency curves. In Example 4, the choice of the component became important since X_1 had a scaling term much smaller than the others, resulting in a lengthy convergence to the right optimal acceptance rate. In Examples 5, it is now crucial to choose this component judiciously since X_1 has an asymptotic distribution that remains discrete. The AOAR generated by this sole component is thus specific to the chosen measure of efficiency, which is not representative of the target distribution as a whole.

3.3 Inhomogeneous Proposal Distribution: An Alternative

We finally consider the remaining situation where there exist b components having scaling terms that are of extremely small order (where b is as in Section 3.2), meaning that they are the only ones to have an impact on the selection of the proposal variance. This is mathematically translated by the satisfaction of Condition (8) along with

$$\lim_{d\to\infty} c\left(i,d\right)\theta_{n+i}^{2}\left(d\right)\varphi\left(d\right)=0 \ \ \forall\, i\in\left\{1,\ldots,m\right\}. \tag{11}$$

This means that if we were basing our prognostic for $\varphi\left(d\right)$ on the last $d-n$ components only, we would opt for a larger order proposal variance. The b components thus become the only one to have an impact on the accept/reject ratio as the dimension of the target increases.

In these circumstances, the value $\hat{\ell}$ is maximizing the equation

$$\upsilon\left(\ell\right)=2\ell^{2}\mathrm{P}_{\mathbf{Y}^{(b)},\mathbf{X}^{(b)}}\left(\sum_{j=1}^{b}\varepsilon\left(X_{j},Y_{j}\right)>0\right)=\ell^{2}a\left(\ell\right). \tag{12}$$

Attempting to optimize $\upsilon\left(\ell\right)$ leads to an impasse, since this function is unbounded for basically any smooth density $f\left(\cdot\right)$. That is, $\upsilon\left(\ell\right)$ increases with ℓ, resulting in a null AOAR. This phenomenon can be explained by the fact that the scaling of the first b components are much smaller than the others, determining the form of $\sigma^{2}\left(d\right)$ as a function of d. However, the moves generated by a proposal distribution with such a variance will definitely be too small for the other components, forcing the parameter ℓ to increase in order to generate reasonable moves for them. In practice, it is thus impossible to find a proposal variance that is small enough for the first b components, but at the same time large enough so as to generate moves that are not compromising the convergence speed of the last $d-b$ components. In Section 3.2, the situation encountered was similar, except that it was possible to achieve an equilibrium between these two constraints. In the current circumstances, the discrepancy between the scaling terms is too large and the disparities are irreconcilable.

In practice, we then face an important difficulty as no choice of ℓ will lead to an efficient algorithm for large values of d. Therefore, modifying the proposal distribution becomes essential and we turn to inhomogeneous proposal variances. The approach chosen is to maintain the same parameter ℓ for each component and also the same form for the proposal variance of the first n terms, while adjusting the form of $\sigma^{2}\left(d\right)$ as a function of d for the last $d-n$ components. In particular, we let $\sigma_{j}^{2}\left(d\right)=\ell^{2}\varphi_{j}\left(d\right)$ with $\varphi_{j}\left(d\right)$ as in (4) for $j=1,\ldots,n$. For every component $j\in\left\{n+1,\ldots,d\right\}$ and belonging to the i-th of the m groups, $\varphi_{j}\left(d\right)$ is now defined to be the largest order function satisfying

$$0<\lim_{d\to\infty} c\left(i,d\right)\theta_{n+i}^{2}\left(d\right)\varphi_{j}\left(d\right)<\infty. \tag{13}$$

Under the inhomogeneity assumption for the proposal variances, we can then use the results presented in Section 3.2 with $E_{R}=\lim_{d\to\infty}\sum_{j=n+1}^{d}\theta_{j}^{2}\left(d\right)\varphi_{j}\left(d\right)\mathrm{E}\left[\left(\frac{f'(X)}{f(X)}\right)^{2}\right]$, and determine $\hat{\ell}$ by maximizing (10). Note that E_{R} is equivalent to the previous definition appearing in (7), but has now been adjusted to take the inhomogeneous proposal variances into account. To illustrate such a situation, consider the following example.

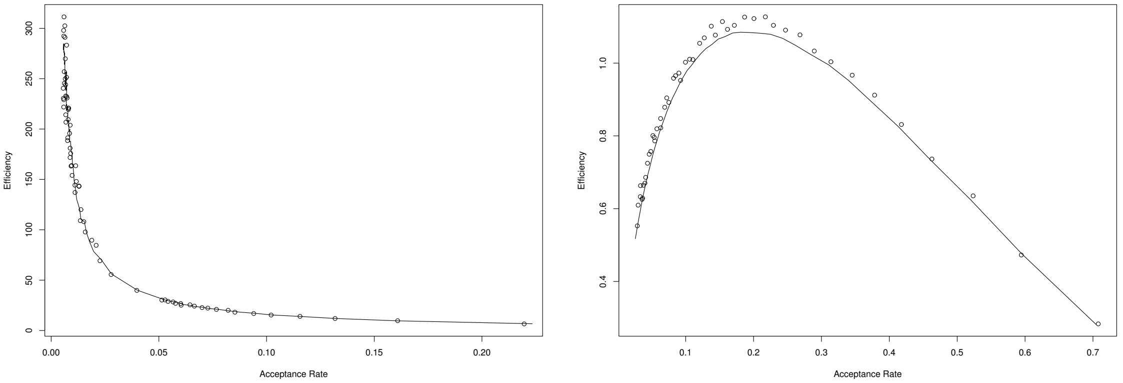

Figure 4: Efficiency of X_2 versus the acceptance rate for homogeneous and inhomogeneous proposal variances respectively. The solid line represents the theoretical curve $\upsilon(\ell)$ and the dotted line has been obtained by running a Metropolis algorithm in dimensions 101.

Example 6. Suppose $f(\cdot)$ in (2) is the standard normal density and consider the vector of variances $\mathbf{\Theta}^{-2}(d) = \left(d^{-5}, d^{-1/2}, 3, d^{-1/2}, 3\ldots\right)$. The particularity of this setting resides in the fact that $\theta_1^{-2}(d)$ is extremely small compared to the other scaling terms (so Condition (8) is satisfied). In the present case, $\sigma^2(d) = \ell^2/d^5$ and the proposal variance is totally governed by X_1; indeed, $\lim_{d\to\infty}(d-1)/2d^{4.5} = 0$ and $\lim_{d\to\infty}(d-1)/6d^5 = 0$, implying that Condition (11) is also verified. We must then use (12) to determine how to optimize the efficiency of the algorithm.

Once again, the graphs in Figure 4 have been obtained by applying the same methods as in Examples 3 and 5. As explained previously and as illustrated by the dotted curve of the left graph (for which we performed 100,000 iterations, using $d = 501$ and $\ell^2 = ((1{:}\ 20)*30, (1{:}\ 50)*1000))$, the optimal value for ℓ diverges, resulting in an optimal acceptance rate which converges to 0. The solid line has been obtained by generating 500,000 values for the pair (X_1, Y_1) (where $X_1 \sim N\left(0, d^{-5}\right)$, $Y_1 \sim \left(X_1, \ell^2/d^5\right)$ and where we fix $d = 5{,}000$), and recording the proportion for which $\log f(Y_1)/f(X_1) > 0$. From (12), doubling this proportion and repeating these steps for each value of the vector ℓ^2 allow us to draw the theoretical efficiency curve. Clearly, it is impossible to reach a satisfactory level of efficiency in the limit using the prescribed proposal distribution. To overcome this problem, we shall make use of inhomogeneous proposal distributions. The idea is to personalize the proposal variance of the last $d-1$ terms. The proposal variance for the first term just stays ℓ^2/d^5 and using (13), the vector $\sigma^2(d)$ becomes $\left(\ell^2/d^5, \ell^2/d^{1.5}, \ell^2/d, \ldots, \ell^2/d^{1.5}, \ell^2/d\right)$; we then deduce that $E_R = \lim_{d\to\infty}((d-1)/2d + (d-1)/6d) = 2/3$.

Figure 4 (right graph) has been obtained by running the Metropolis algorithm for 100,000 iterations in dimensions 101 for $\ell^2 = (1{:}\ 50) * 0.4$, where the solid line again represents the theoretical curve $\upsilon(\ell)$ in (10). The theoretical values for $\hat{\ell}^2$ and $a\left(\hat{\ell}\right)$ have been obtained by using the previous vector for ℓ^2 along with the method described in Example 5 with 500,000 generated values for (X_1, Y_1). By scanning the vector $\upsilon(\ell)$ we find $\hat{\ell}^2 = 6$ and $a\left(\hat{\ell}\right) = 0.1808251$ respectively, which agree with the simulations. The inhomogeneous proposal variances have then contributed to decrease $\hat{\ell}$ while raising the AOAR. Indeed, large values for $\hat{\ell}$ are now inappropriate since components with larger scaling terms now possess a proposal variance that is suited to their size, ensuring a reasonable speed of convergence for these components.

For a chosen efficiency measure (e.g. the MSE of the Monte Carlo estimator), it is sensible to wonder how much is lost in terms of performance if the AOAR is not attained. This can be answered by looking at the shapes of the curves obtained throughout the various examples presented previously. We know from Section 2.3 that measures of efficiency selected for finite-dimensional versions of the algorithm become proportional to the function $\upsilon(\ell)$ as $d \to \infty$, meaning that efficiency measures become equivalent for reasonably large values of d. From the theoretical efficiency curves displayed in the previous examples, it is clear that opting for an acceptance rate which is not optimal affects the MSE of the Monte Carlo estimator, or any other preferred efficiency measure. However, the importance of choosing the exact optimal acceptance rate should not be overstated. On all the graphs presented, the curves obtained are not very peaked, meaning that choosing an acceptance rate relatively close to the optimal one would not have a dramatic impact on the performance of the algorithm. In Example 3 for instance, tuning the acceptance rate somewhere between 0.1 and 0.35 would result in a loss of performance of at most 10% when compared to the AOAR.

Of course, efficiency measures do not necessarily all converge towards the speed measure $\upsilon(\ell)$ at the same rate. For relatively small values of d, the loss of performance engendered by not tuning the algorithm to the AOAR might slightly vary from that prescribed by the function $\upsilon(\ell)$. But then again in small dimensions, the asymptotics are not necessarily attained either, so in any case practitioners need to be cautious when applying the results to low-dimensional problems.

3.4 Robustness of Optimal Scaling Results

Throughout the paper, we assumed the independence of the target components $X_1^{(d)}, \ldots, X_d^{(d)}$. We believe that the independence assumption could be relaxed to a certain extent; that is, under weak dependence of the target components, the algorithm should still adopt the behavior prescribed by the theorems. It would however be interesting to gain a better idea of the meaning of weak dependence in our case, and in order to study the robustness of the results introduced, we shall use the concept of copulas. These allow, given the marginals of a multidimensional distribution, to introduce dependency amongst the various target components; specifically, we define the cumulative distribution function (CDF) of $X_1^{(d)}, \ldots, X_d^{(d)}$ by $F_{X_1^{(d)},\ldots,X_d^{(d)}}(x_1, \ldots, x_d) = C\left(F_{X_1^{(d)}}(x_1), \ldots, F_{X_d^{(d)}}(x_d)\right)$, where C is the selected copula (Joe 1997, Nelsen 2006). This concept allows us to first choose marginal distributions, then determine the way they are linked together, and finally select the degree of dependency among the components.

When dealing with copulas, a popular measure of correlation between two random variables is Kendall's tau: $\tau = \tau(X, Y) = \mathrm{P}[(X_2 - X_1)(Y_2 - Y_1) \geq 0] - \mathrm{P}[(X_2 - X_1)(Y_2 - Y_1) < 0]$, where (X_1, Y_1) and (X_2, Y_2) are two independent realizations of a joint distribution. In what follows, we consider the Cook-Johnson copula, belonging to the Archimedean family:

$$F^{(\alpha)}_{X_1^{(d)},\ldots,X_d^{(d)}}\left(x_1,\ldots,x_d\right) = \left\{\sum_{j=1}^{d}\left(F_{X_j^{(d)}}\left(x_j\right)\right)^{-1/\alpha} - d + 1\right\}^{-\alpha},$$

where $\alpha > 0$. A particularity of this copula is that all pairs of components possess the same Kendall's tau, which shall allow us to easily set the desired degree of correlation in the target distribution. Kendall's tau is closely related to the parameter α through the relation $\tau = (1+2\alpha)^{-1}$. When $\alpha = 0$, then $\tau = 1$ and the dependence among target components is maximal; as α increases, τ approaches 0 and the components become independent; the Cook-Johnson copula thus introduces positive correlation between each pair of components.

By performing simulation studies similar to those presented previously, we can get insight about the level of dependency tolerated by the theorems. Specifically, we assume $d = 100$ random variables to be distributed according to a Gamma with shape parameter 5 (as in Example 5) and scaling vector $\mathbf{\Theta}^{-2}(d)$; we then link these marginal distributions with the Cook-Johnson copula. For such a target distribution we find that, regardless of the scaling vector $\mathbf{\Theta}^{-2}(d)$, the results presented perfectly hold when $\alpha > 6$, corresponding to a Kendall's tau smaller than 0.075. For instance, if $\mathbf{\Theta}^{-2}(d) = (1,\ldots,1)$ then Condition (5) is satisfied, $\hat{\ell}^2 = 3\,(2.38)^2 = 17$, and the AOAR is 0.234. As the degree of dependence increases past this point (i.e. as $\alpha \searrow 0$), the results slowly get further from those prescribed by the theorems. Of course, the model used here to introduce dependency is quite specific and in practice, we do not necessarily deal with distributions possessing identical pairwise correlations. This however gives a good idea of what to expect when assumptions are relaxed.

When dealing with target distributions exhibiting weakly dependent components, it then suffices to ignore the dependency and apply the results using the vector of scaling terms drawn from the marginal distributions. In cases where the dependence structure is too strong, the prescribed comportment is violated and the results nonapplicable. As we shall see in Section 4, optimal scaling results for hierarchical models seem to exist, but do not agree with the results derived under the independence assumption. In that case, the limiting behavior of the target distribution appears to be considerably more complex and is presently under investigation.

In this section, we referred to Kendall's tau as the dependency measure because it is conveniently related to the parameter α of the Cook-Johnson copula. In practice, we might be tempted to compute the usual Pearson's correlation instead. One should keep in mind that the use of Pearson's correlation as a measure of dependence can sometimes be misleading and that Kendall's tau is generally more reliable. However, both dependency measures should be used cautiously. For instance, suppose that σ^2 is distributed according to an inverse gamma distribution and that $X_i\,|\sigma^2 \sim N\left(0,\sigma^2\right),\ i = 1,\ldots,d$; the resulting covariance matrix is diagonal with $O(1)$ terms. However, the random variables are clearly not independent, and using either Pearson's correlation coefficient or Kendall's tau abusively suggest an AOAR of 0.234. In that case, the measures of dependency considered do not provide an explanation of the relationships between the various components. In cases where the covariance matrix or the set of Kendall's taus seem to provide a reliable account of the dependence structure, the method presented in this paper seems to be robust to the inclusion of weak dependence in the target.

4 Simulation Studies for some Hierarchical Models

A nice feature of the results presented in this paper is their capability to optimize the efficiency of the Metropolis algorithm when sampling from any multivariate normal target distribution, no matter the correlation structure existing between its components. The normal hierarchical target model considered in Section 4.1 illustrates this property. The last two sections focus on empirically studying the optimal scaling problem for more general hierarchical models, engendering distributions that are not jointly normal.

4.1 Normal Hierarchical Model

Consider a model with location parameters $\mu_1 \sim N(0,1)$ and $\mu_2 \sim N(\mu_1, 1)$. Further suppose, assuming conditional independence, that $X_i \sim N(\mu_1, 1)$, $i = 1, \ldots, 9$ and $X_i \sim N(\mu_2, 1)$, $i = 10, \ldots, 18$. The joint distribution of μ_1, μ_2, X_1, ..., X_{18} is multivariate normal with null mean and covariance matrix Σ_{20}. Obtaining the covariances between each pair of components is easily achieved by using conditioning; the covariance matrix is then

$$\Sigma_{20} = \begin{pmatrix} 1 & 1 & & \cdots & & 1 & 1 & & \cdots & 1 \\ 1 & 2 & 1 & & \cdots & 1 & 2 & & \cdots & 2 \\ 1 & 1 & 2 & 1 & \cdots & 1 & 1 & \cdots & 1 & 1 \\ \vdots & & 1 & 2 & & \vdots & \vdots & \ddots & \vdots & \vdots \\ & \vdots & \vdots & & \ddots & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 2 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 1 & \cdots & 1 & 1 & 3 & 2 & \cdots & 2 \\ \vdots & & \vdots & \ddots & \vdots & \vdots & 2 & 3 & & \vdots \\ & \vdots & 1 & \cdots & 1 & 1 & \vdots & & \ddots & 2 \\ 1 & 2 & 1 & \cdots & 1 & 1 & 2 & \cdots & 2 & 3 \end{pmatrix}_{20\times 20}.$$

A useful property of multivariate normal distributions is their invariance under orthogonal transformations. It is therefore possible to transform Σ_{20} into a diagonal matrix where the diagonal elements consist in the eigenvalues of Σ_{20}. Since the target distribution is still normal but has now independent components, optimizing the efficiency of the Metropolis algorithm can be achieved by using the results presented previously.

In order to determine which one of (6), (10) or (12) should be used for determining $\hat{\ell}$, we need to know how the eigenvalues of Σ_d evolve as a function of d. Obtaining numerical values for the eigenvalues of Σ_d in any dimension is easily achieved with the help of any statistical software; this allows us to deduce that $d-4$ of the d eigenvalues are exactly equal to 1. Plots of the four remaining eigenvalues, $\lambda_i(d)$, $i = 1, \ldots, 4$, clearly show that the two smallest eigenvalues satisfy $a_i/d = \lambda_i(d)$; they also reveal a relation of the form $a_i d = \lambda_i(d)$ for the two largest eigenvalues. Fitting these linear equations using the eigenvalues of Σ_{600} (say), we obtain fitted values for a_i, $i = 1, 2, 3, 4$.

Optimizing the efficiency of the algorithm for sampling from this hierarchical model then reduces to optimize a 20-dimensional multivariate normal distribution with independent components, null mean and variances equal to $(1.983/20, 1.997/20, 0.193\,(20), 1.311\,(20), 1, \ldots, 1)$.

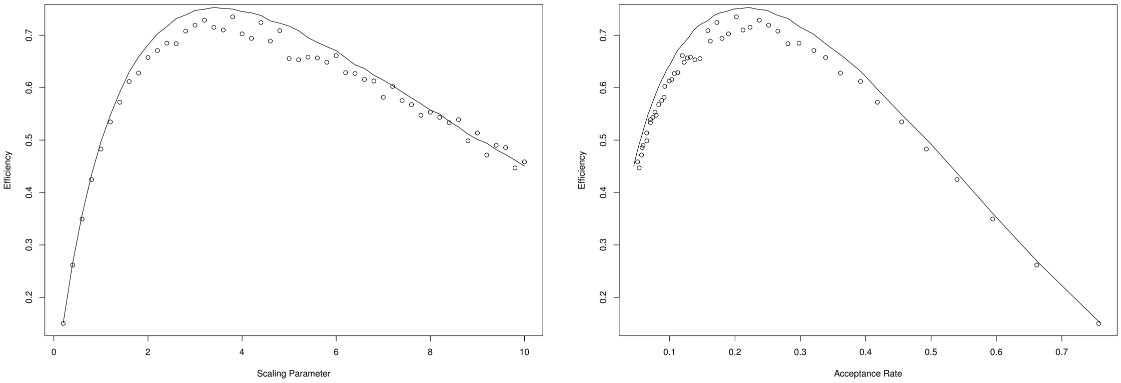

Figure 5: Left graph: efficiency of X_3 versus ℓ^2. Right graph: efficiency of X_3 versus the acceptance rate. The solid line represents the theoretical curve, while the dotted curve is the result of the simulation study.

It is easily verified that this vector satisfies Conditions (8) and (9), and leads to $\sigma^2(\ell) = \ell^2/d$. We then turn to equation (10) to optimize the efficiency of the algorithm; using $E_R = 1$, $d = 5,000$ (so $X_1 \sim N(0, 1.983/5,000)$, $X_2 \sim N(0, 1.997/5,000)$, $Y_i \sim N(X_i, \ell^2/5,000)$, $i = 1, 2$) and $\ell^2 = (1\colon 50) * 0.2$ along with the method described in Example 5, we estimate that $\hat{\ell}^2 = 3.4$, for which $a(\hat{\ell}) = 0.2214$. The value $\hat{\ell} = 1.84$ thus differs from the baseline 2.38 in Section 3.1, but still yields an AOAR that is close to 0.234.

Figure 5 presents graphs based on 100,000 iterations of the Metropolis algorithm, depicting how the first order efficiency of X_5 relates to ℓ^2 and the acceptance rate respectively (we again used $\ell^2 = (1\colon 50) * 0.2$). The curves obtained emphasize the rapid convergence of the algorithm in finite dimensions to its asymptotic counterpart, represented by the solid line. As before, the starting values of the algorithms have been sampled from the target distribution and the same methods as before have been used for estimating FOE and the acceptance rate.

Since non-normal target distributions are not necessarily invariant under orthogonal transformations, it would be misleading to use the eigenvalues of the covariance matrix to apply the results introduced in this paper. We must then stick to the analysis of weakly dependent non normal targets only, as discussed in Section 3.4.

Sometimes, the covariance matrix of the target distribution is not easily computed. Recall however that to perform Metropolis algorithms, we must possess information on the distribution we wish to sample from, up to the normalization constant. A solution would be to perform the Metropolis algorithm with the given target to obtain a preliminary empirical estimate of the covariance matrix (or of the set of Kendall's taus). This information could then be used to determine whether or not the dependence structure among the components is weak enough to apply the results presented in this paper. This method could also be used to estimate the scaling vector, and may be repeated for different dimensions of the target to determine how the scaling terms evolve as a function of d.

4.2 Variance Components Model

The second simulation study focuses on the variance components model. Let $\mu \sim N(0,1)$, $\sigma_\theta^2 \sim IG(3,1)$ and $\sigma_e^2 \sim IG(2,1)$. The means θ_i are conditionally IID given μ, σ_θ^2 and are distributed according to $\theta_i \sim N\left(\mu, \sigma_\theta^2\right)$ for $i = 1, \ldots, 30$. The 30 groups of data values are conditionally independent given the mean vector $(\theta_1, \ldots, \theta_{30})$ and the variance σ_e^2, while the values within each group are IID. In particular, $Y_{i,j} \sim N\left(\theta_i, \sigma_e^2\right)$ for $i = 1, \ldots, 30$ and $j = 1, \ldots, 10$.

We are interested in the posterior distribution of $\mu, \sigma_\theta^2, \sigma_e^2, \theta_1, \ldots, \theta_{30}$ given the data $Y_{i,j}$, $i = 1, \ldots, 30$, $j = 1, \ldots, 10$. Since the algorithm does not work well when generating moves from a normal proposal to mimic moves from an inverse gamma distribution, we use inverse transformations and instead deal with gamma distributions. The target is such that

$$\pi\left(\mu, \varphi_\theta, \varphi_e, \theta_1, \ldots, \theta_{30} \,|\mathbf{Y}\,\right) \propto$$
$$(\varphi_\theta)^{17} (\varphi_e)^{151} \exp\left(-\frac{\mu^2}{2\sigma_0^2} - \varphi_\theta - \varphi_e - \sum_{i=1}^{30} \frac{\varphi_\theta (\theta_i - \mu)^2}{2} - \sum_{i=1}^{30}\sum_{j=1}^{10} \frac{\varphi_e (\theta_i - Y_{i,j})^2}{2}\right), \quad (14)$$

where $\varphi_\theta = 1/\sigma_\theta^2$ and $\varphi_e = 1/\sigma_e^2$.

We run the Metropolis algorithm with a target as in (14), updating the variables μ, φ_θ, φ_e, θ_1, ..., θ_{30}. For the sake of the example, the data was simulated from the target. We performed 100,000 iterations and plotted FOE_4 versus ℓ^2 and the acceptance rate. We used the method described in Example 3 with $\ell^2 = (1{:}\ 50) * 0.01$, and then the function smooth.spline in R to obtain a smooth curve. The maximum is located around 0.17 for $\hat{\ell}^2$ and choosing an acceptance rate close to 0.2 optimizes the efficiency of the algorithm. Although the AOAR seems to lie close to 0.234, it is hard to tell its exact value from the graph. According to the previous results, we suspect that it might differ from 0.234, which might become clearer when simulating from target distributions possessing a greater number of non-normal components. Although the joint distribution is not normally distributed, it then seems possible to optimize not only hierarchical models where the mean of normally distributed variables is random, but also hierarchical models with more layers and random variances.

4.3 Gamma-Gamma Hierarchical Model

Let $\lambda \sim \Gamma(4,1)$ and, assuming conditional independence, $X_i \sim \Gamma(4, \lambda)$ for $i = 1, \ldots, 20$. The unconditional 21-dimensional target density satisfies

$$\pi(\lambda, x_1, \ldots, x_{20}) \propto \lambda^{83} \exp\left(-\lambda\left(1 + \sum_{i=1}^{20} x_i\right)\right) \prod_{i=1}^{20} x_i^3.$$

This time, 10,000,000 iterations of the algorithm were required to reduce Monte Carlo errors and obtain clear curves (we again used the cubic smoothing spline in R to draw the plots, but this time based on $\ell^2 = (1{:}\ 50) * 0.1$). Figure 7 shows the existence of a finite value $\hat{\ell}$ optimizing the efficiency of the method ($\hat{\ell}^2 \approx 1.6$), resulting in an optimal acceptance rate lying around 0.16. This small AOAR appears to corroborate the discussion at the end of

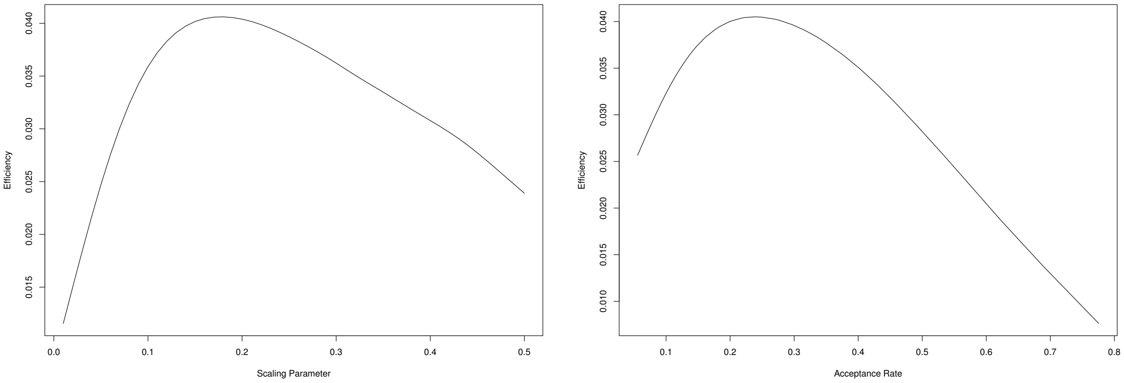

Figure 6: Left graph: efficiency of θ_1 versus ℓ^2. Right graph: efficiency of θ_1 versus the acceptance rate.

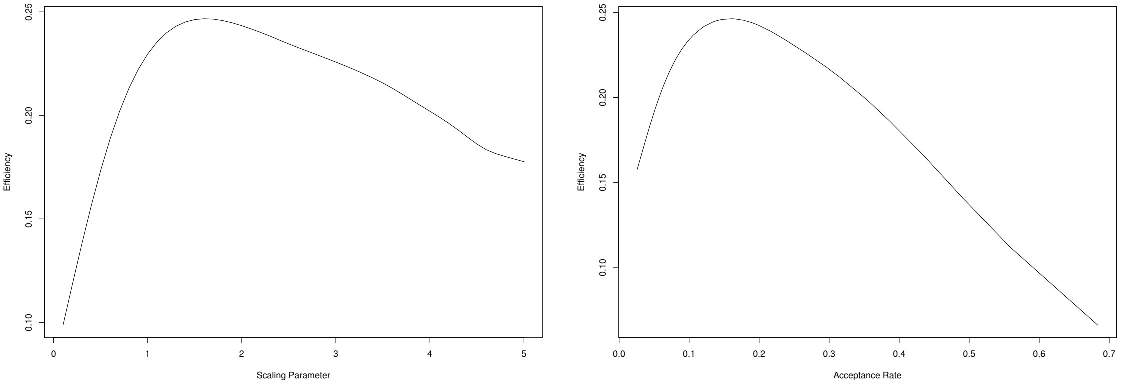

Figure 7: Left graph: efficiency of X_1 versus ℓ^2 . Right graph: efficiency of X_1 versus the acceptance rate.

last section. That is, it seems feasible to optimize the efficiency of Metropolis algorithms for general hierarchical target models and this will yield AOARs that are smaller than 0.234.

5 Discussion

The results presented in this paper permit to optimize the efficiency for sampling from target densities as described in Section 2.1 using Metropolis algorithms with proposal distributions as in Section 2.2. This has been illustrated with numerous examples throughout the paper, which aimed to outline the fact that applying these asymptotic results even to relatively low-dimensional target distributions produces satisfactory conclusions. They also provided evidence that the acceptance rate might considerably differ from 0.234, from where the importance of solving for the correct AOAR. A drastic variation in the AOAR seems to be more common with target distributions that are not normally distributed. In general, AOARs for multivariate normal targets appear to lie close to 0.234, regardless of the correlation structure existing among the components. As discussed in Section 3.3 however, even for the most regular target distributions, an extremely small scaling term causes the algorithm to be inefficient and forces us to resort to inhomogeneous proposal distributions. The AOAR obtained under this method is then not necessarily close to 0.234. In general, the results presented also hold under weak dependence of the target components, as discussed in Section 3.4.

As mentioned previously, since our results can be used to optimize any multivariate normal target distribution, this includes cases where the target is any level normal hierarchical model, meaning that the variances of the distributions are fixed while the mean is random and normally distributed. This raises the question as to whether Metropolis algorithms can be optimized when the variance of the normal is also random, or more generally if similar results can be derived for broader hierarchical models. The examples presented in Sections 4.2 and 4.3 seem to answer this question positively, but AOARs appear to differ from 0.234 with increasing significance as the distribution gets further from normality. The optimization problem for general hierarchical models is presently under investigation (see Bédard, 2006c).

Acknowledgments

This work is part of my Ph.D. thesis and has been supported by NSERC of Canada. Special thanks are due to my supervisor, Professor Jeffrey S. Rosenthal, without who this work would not have been complete. His expertise, guidance and encouragements have been precious throughout my studies. I am also grateful to the Associate Editor and anonymous referees for constructive comments.

References

Bédard, M. (2006a), "Weak Convergence of Metropolis Algorithms for Non-IID Target Distributions," *to appear in Annals of Applied Probability.*

Bédard, M. (2006b), "Optimal Acceptance Rates for Metropolis Algorithms: Moving Beyond 0.234," *submitted for publication in Annals of Statistics.*

Bédard, M. (2006c), "On the Optimization of Metropolis Algorithms for Hierarchical Target Distributions," in preparation.

Breyer, L. A., Roberts, G. O. (2000), "From Metropolis to Diffusions: Gibbs States and Optimal Scaling," *Stochastic Processes and their Applications,* **90**, 181-206.

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

Joe, H. (1997), "Multivariate Models and Multivariate Dependence Concepts (Monographs on Statistics and Applied Probability)" *Chapman & Hall*, New York.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953), "Equations of state calculations by fast computing machines," *Journal of Chemical Physics,* **21**, 1087-92.

Neal, P., Roberts, G. O. (2004), "Optimal Scaling for Partially Updating MCMC Algorithms," *to appear in Annals of Applied Probability.*

Nelsen, R. B. (2006), "An Introduction to Copulas," 2nd edition, *Springer*, New York.

Pasarica, C., Gelman A. (2003), "Adaptively scaling the Metropolis algorithm using expected squared jumped distance," technical report, Department of Statistics, Columbia University.

Roberts, G. O., Gelman, A., Gilks, W. R. (1997), "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," *Annals of Applied Probability,* **7**, 110-20.

Roberts, G. O., Rosenthal, J. S. (1998), "Optimal Scaling of Discrete Approximations to Langevin Diffusions," *Journal of the Royal Statistical Society, Series B,* **60**, 255-68.

Roberts, G. O., Rosenthal, J. S. (2001), "Optimal Scaling for various Metropolis-Hastings algorithms," *Statistical Science,* **16**, 351-67.

Sherlock, C. (2006), "Methodology for Inference on the Markov Modulated Poisson Process and Theory for Optimal Scaling of the Random Walk Metropolis," Ph.D. dissertation, Lancaster University, Department of Mathematics and Statistics.