Statistica Sinica **20** (2010), 343-364

ADAPTIVELY SCALING THE METROPOLIS ALGORITHM USING EXPECTED SQUARED JUMPED DISTANCE

Cristian Pasarica and Andrew Gelman

J. P. Morgan and Columbia University

Abstract: A good choice of the proposal distribution is crucial for the rapid convergence of the Metropolis algorithm. In this paper, given a family of parametric Markovian kernels, we develop an adaptive algorithm for selecting the best kernel that maximizes the expected squared jumped distance, an objective function that characterizes the Markov chain. We demonstrate the effectiveness of our method in several examples.

Key words and phrases: Acceptance rates, Bayesian computation, iterative simulation, Markov chain Monte Carlo, multiple importance sampling.

1. Introduction

1.1. Adaptive MCMC algorithms: motivation and difficulties

The algorithm of Metropolis, Rosenbluth, Rosenbluth and Teller (1953) is an important tool in statistical computation, especially in calculation of posterior distributions arising in Bayesian statistics. The Metropolis algorithm evaluates a (typically multivariate) target distribution $\pi(\theta)$ by generating a Markov chain whose stationary distribution is π. Practical implementations often suffer from slow mixing and therefore inefficient estimation, for at least two reasons: the jumps are too short and therefore simulation moves very slowly to the target distribution; or the jumps end up in low probability areas of the target density, causing the Markov chain to stand still most of the time. In practice, adaptive methods have been proposed in order to tune the choice of the proposal, matching some criteria under the invariant distribution (e.g., Kim, Shephard and Chib (1998), Haario, Saksman and Tamminen (1999), Laskey and Myers (2003), Andrieu and Robert (2001), and Atchadé and Rosenthal (2003)). These criteria are usually defined based on theoretical optimality results, for example, for a d-dimensional target with i.i.d. components the optimal scaling of the jumping kernel is $c_d = 2.38/\sqrt{d}$ (Roberts, Gelman and Gilks (1997)). These results are based on the asymptotic limit of infinite-dimensional iid target distributions only, but in practice can be applied to dimensions as low as 5 (Gelman, Roberts and Gilks (1996)). Extensions of these results appear in Roberts and Rosenthal (2001).

Another approach is to coerce the acceptance probability to a preset value (e.g., 44% for one-dimensional target). This can be difficult to apply due to the complicated form of the target distribution which makes the optimal acceptance probability value difficult to compute. In practice, problems arise for distributions (e.g., multimodal) for which the normal-theory optimal scaling results do not apply, and for high-dimensional targets where initial optimization algorithms cannot find easily the global maximum of the target function, yielding a proposal covariance matrix different from the covariance matrix under the invariant distribution. A simple normal-normal hierarchical model example is considered in Bédard (2006); here the target distribution apparently breaks the assumption of independent components, but can be transformed to a d-dimensional multivariate independent normal target distribution with variance $(1/(d+1), d+1, 1, \ldots, 1)$. In this case, the optimal acceptance rate(with respect to the best possible mixing of states and fast convergence to stationarity) is 21.6%, slightly different than the theoretical optimal acceptance rate of 23.4% that holds for inhomogeneous target distributions $\pi(x^{(d)}) = \Pi_{i=1}^{d} C_i f(C_i x_i)$ (see Roberts and Rosenthal (2001)). When the target distribution moves further away from normality, as for example with the gamma hierarchical model, Bédard (2006) finds an optimal acceptance rate of 16%. More general optimal acceptance rates are based on the asymptotic behavior of the target distribution and can be found in some special cases (see Bédard (2006)).

In general, the adaptive proposal Metropolis algorithms do not simulate exactly the target distribution: the Markovian property or time-homogeneity of the transition kernel is lost, and ergodicity can be proved only under some conditions (see Tierney and Mira (1999), Haario, Saksman and Tamminen (2001), Holden (2000), Atchadé and Rosenthal (2003), Haario, Laine, Mira and Saksman (2006), and Roberts and Rosenthal (2006)). Adaptive methods that preserve the Markovian properties by using regeneration times have the challenge of estimating regeneration times; this is difficult for algorithms other than independence chain Metropolis (see Gilks, Roberts, and Sahu (1998)). In practice, we follow a two-stage finite adaptation approach: a series of adaptive optimization steps followed by an MCMC run with fixed kernel. We also consider an infinite adaptation version of the algorithm.

1.2. Our proposed method based on expected squared jumped distance

In this paper we propose a general framework which allows for the development of new MCMC algorithms that, in order to explore the target, are able to learn automatically the best strategy among a set of proposed strategies $\{J_\gamma\}_{\gamma\in\Gamma}$, where Γ is some finite-dimensional domain, in order to explore the target distribution π. Measures of efficiency in low-dimensional Markov chains are not unique

(see Gelman, Roberts and Gilks (1996) for discussion). A natural measure of efficiency is the asymptotic variance of the sample mean $\bar{\theta} = (1/T)\sum_{t=1}^{T}\theta_t$. The asymptotic efficiency of Markov chain sampling for $\bar{\theta}$ is defined as

$$\text{eff}_{\bar{\theta}}(\gamma) = \frac{\text{Var}_{(\pi)}(\bar{\theta})}{\text{Var}_{(J_\gamma)}(\bar{\theta})} = \Big[1 + 2(\rho_1 + \rho_2 + \cdots)\Big]^{-1}, \quad \gamma \in \Gamma,$$

where $\text{Var}_{(\pi)}$ denotes the variance under independent sampling, $\text{Var}_{(J_\gamma)}$ denotes the limiting scale sample variance from the MCMC output, and $\rho_k = (1/(T-k))\sum\theta_t\theta_{t-k}$ denotes the autocorrelation of the Markov chain at lag k. Our measure and alternative measures of efficiency in the MCMC literature are related to the eigenvalue structure of the transition kernel (see, for example, Besag and Green (1993)). Fast convergence to stationarity in total variation distance is attained by having a low second eigenvalue modulus. Maximizing asymptotic efficiency is a criterion proposed in Andrieu and Robert (2001) but the difficulty lies in estimating the higher order autocorrelations $\rho_2, \rho_3, \ldots$, since these involve estimation of an integral with respect to the Dirac measure. We maximize the expected squared jumping distance (ESJD):

$$\text{ESJD}(\gamma) \stackrel{\triangle}{=} \mathbf{E}_{J_\gamma}\Big[\|\theta_{t+1} - \theta_t\|^2\Big] = 2(1-\rho_1)\cdot\text{Var}_{(\pi)}(\theta_t),$$

for a one-dimensional target distribution π. Clearly, $\text{Var}_{(\pi)}(\theta_t)$ is a function of the stationary distribution only, thus choosing a transition rule to maximize ESJD is equivalent to minimizing the first-order autocorrelation ρ_1 of the Markov chain (and thus maximizing the efficiency if the higher order autocorrelations are monotonically increasing with respect to ρ_1). Nevertheless, it is easy to imagine a bimodal example in several dimensions in which the monotonicity of the higher order autocorrelations does not hold and jumps are always between the modes, giving a negative lag-1 autocorrelation, but a positive lag-2 autocorrelation. For this situation we can modify the efficiency criteria to include higher order autocorrelations as the method we present is a case of a general framework. However, our method will work under any other objective function and d-dimensional target distribution (see Section 2.4).

We present here an outline of our procedure.

1. Start the Metropolis algorithm with some initial kernel; keep track of both the Markov chain θ_t and proposals θ_t^*.
2. After every T iterations, update the covariance matrix of the jumping kernel using the sample covariance matrix, with a scale factor that is computed by optimizing an importance sampling estimate of the ESJD.
3. After some number of the above steps, stop the adaptive updating and run the MCMC with a fixed kernel, treating the previous iterations up to that point as a burn-in.

Importance sampling techniques for Markov chains, unlike the methods for independent variables, typically require the whole path for computing the importance sampling weights, thus making them computationally expensive. We take advantage of the properties of the Metropolis algorithm to use the importance weights that depend only on the current state, and not of the whole history of the chain. The multiple importance sampling techniques introduced in Geyer and Thompson (1992, reply to discussion) and Geyer (1996) help stabilize the variance of the importance sampling estimate over a broad region, by treating observations from different samples as observations from a mixture density. We study the convergence of our method by using the techniques of Geyer (1994) and Roberts and Rosenthal (2006).

This paper describes our approach, in particular, the importance sampling method used to optimize the parameters of the jumping kernel $J_\gamma(\cdot,\cdot)$ after a fixed number of steps, and illustrates it with several examples. We also compare our procedure with the Robbins-Monro stochastic optimization algorithm (see, for example, Kushner and Yin (2003)). We describe our algorithm in Section 2, and in Section 3 we discuss implementation with Gaussian kernels. Section 4 includes several examples, and we conclude with discussion and open problems in Section 5.

2. The Adaptive Optimization Procedure

2.1. Notation

To define Hastings's (1970) version of the algorithm, suppose that π is a target density absolutely continuous with respect to Lebesgue measure, and let $\{J_\gamma(\cdot,\cdot)\}_{\gamma\in\Gamma}$ be a family of proposal kernels. For fixed $\gamma \in \Gamma$ define

$$\alpha_\gamma(\theta,\theta^*) = \min\left\{\frac{J_\gamma(\theta^*,\theta)\pi(\theta^*)}{\pi(\theta)J_\gamma(\theta,\theta^*)},1\right\}.$$

If we define the off-diagonal density of the Markov process,

$$p_\gamma(\theta,\theta^*) = \begin{cases} J_\gamma(\theta,\theta^*)\alpha_\gamma(\theta,\theta^*), & \theta \neq \theta^* \\ 0, & \theta = \theta^* \end{cases} \tag{2.1}$$

and set

$$r_\gamma(\theta) = 1 - \int p_\gamma(\theta,\theta^*)d\theta^*,$$

then the Metropolis transition kernel can be written as

$$K_\gamma(\theta,d\theta^*) = p_\gamma(\theta,\theta^*)d\theta^* + r_\gamma(\theta)\delta_\theta(d\theta^*).$$

Throughout this paper we use the notation θ_t^* for the proposal generated by the Metropolis chain under proposal kernel $J_\gamma(\cdot, \theta_t)$ and denote by $\Delta_t \stackrel{\triangle}{=} \theta_t^* - \theta_t$, the proposed jumping distance.

2.2. Optimization of the proposal kernel after one set of simulations

Following Andrieu and Robert (2001), we define the objective function that we seek to maximize adaptively as

$$h(\gamma) \stackrel{\triangle}{=} \mathbf{E}\Big[H(\gamma, \theta_t, \theta_t^*)\Big] = \int \int_{\mathbf{R}^d \times \mathbf{R}^d} H(\gamma, x, y)\pi(x)J_\gamma(x, y)dxdy. \quad (2.2)$$

We start our procedure by choosing an initial proposal kernel $J_{\gamma_0}(\cdot, \cdot)$ and running the Metropolis algorithm for T steps. We use the T simulation draws θ_t and the proposals θ_t^* to construct the importance sampling estimator of $h(\gamma)$,

$$\hat{h}_T(\gamma|\gamma_0) \stackrel{\triangle}{=} \frac{\sum_{t=1}^T H(\gamma, \theta_t, \theta_t^*) \cdot w_{\gamma|\gamma_0}(\theta_t, \theta_t^*)}{\sum_{t=1}^T w_{\gamma|\gamma_0}(\theta_t, \theta_t^*)}, \ \forall \gamma \in \Gamma, \quad (2.3)$$

or the mean estimator

$$\hat{\hat{h}}_T(\gamma|\gamma_0) \stackrel{\triangle}{=} \frac{1}{T}\sum_{t=1}^T H(\gamma, \theta_t, \theta_t^*)w_{\gamma|\gamma_0}(\theta_t, \theta_t^*), \ \forall \gamma \in \Gamma, \quad (2.4)$$

where

$$w_{\gamma|\gamma_0}(\theta, \theta^*) \stackrel{\triangle}{=} \frac{J_\gamma(\theta, \theta^*)}{J_{\gamma_0}(\theta, \theta^*)}, \quad (2.5)$$

are the importance sampling weights. On the left side of (2.3) the subscript T emphasizes that the estimate comes from T simulation draws, and we explicitly condition on γ_0 because the importance sampling weights require J_{γ_0}.

We typically choose as objective function the expected squared jumped distance $H(\gamma, \theta, \theta^*) = \|\theta - \theta^*\|_{\Sigma^{-1}}^2 \alpha_\gamma(\theta, \theta^*) = (\theta - \theta^*)^t \Sigma^{-1}(\theta - \theta^*)\alpha_\gamma(\theta, \theta^*)$, where Σ is the covariance matrix of the target distribution π, because maximizing this distance is equivalent to minimizing the first-order autocorrelation in covariance norm. We return to this issue and discuss other choices of objective function in Section 2.4. We optimize the empirical estimator (2.3) using a numerical optimization algorithm such as Brent's (see, e.g., Press, Teukolski, Vetterling and Flannery (2002)) as we further discuss in Section 2.6. In Section 4 we discuss the computation time needed for the optimization.

2.3. Iterative optimization of the proposal kernel

If the starting point, γ_0, is not in the neighborhood of the optimum, then an effective strategy is to iterate the optimization procedure, both to increase

the amount of information used in the optimization and to use more effective importance sampling distributions. The iteration allows us to get closer to the optimum and not rely too strongly on our starting distribution. We explore the effectiveness of the iterative optimization in several examples in Section 4. In our algorithm, the "pilot data" used to estimate h will come from a series of different proposal kernels. The function h can be estimated using the method of multiple importance sampling (see Hesterberg (1995)), yielding the following algorithm based on adaptively updating the jumping kernel after steps $T_1, T_1 + T_2, T_1 + T_2 + T_3, \ldots$. For $k = 1, 2, \ldots$.

1. Run the Metropolis algorithm for T_k steps according to proposal rule $J_{\gamma_k}(\cdot, \cdot)$. Save the sample and proposals, $(\theta_{k1}, \theta^*_{k1}), \ldots, (\theta_{kT_k}, \theta^*_{kT_k})$.
2. Find the maximum γ_{k+1} of the empirical estimator $h(\gamma|\gamma_{k-1}, \ldots, \gamma_0)$, defined as

$$\hat{h}(\gamma|\gamma_{k-1}, \ldots, \gamma_0) = \frac{\sum_{i=1}^{k} \sum_{t=1}^{T_i} H(\gamma, \theta_{it}, \theta^*_{it}) \cdot w_{\gamma|\gamma_{k-1},\ldots,\gamma_0}(\theta_{it}, \theta^*_{it})}{\sum_{i=1}^{k} \sum_{t=1}^{T_i} w_{\gamma|\gamma_{k-1},\ldots,\gamma_0}(\theta_{it}, \theta^*_{it})}, \quad (2.6)$$

   where the *multiple importance sampling weights* are

$$w_{\gamma|\gamma_j,\ldots,\gamma_0}(\theta, \theta^*) \stackrel{\triangle}{=} \frac{J_\gamma(\theta, \theta^*)}{\sum_{i=1}^{j} T_i J_{\gamma_{i-1}}(\theta, \theta^*)}, \quad j = 1, \ldots, k. \quad (2.7)$$

We are treating the samples as having come from a mixture of j components $j = 1, \ldots, k$. The weights satisfy the condition $\sum_{i=1}^{k} \sum_{t=1}^{T_i} w_{\gamma|\gamma_{k-1},\ldots,\gamma_0}(\theta_{it}, \theta^*_{it}) = 1$, and are derived from the individual importance sampling weights by substituting $J_\gamma = \omega_{\gamma|\gamma_j} J_{\gamma_j}$ in the numerator of (2.7). With independent multiple importance sampling, these weights are optimal in the sense that they minimize the variance of the empirical estimator (see Veach and Guibas (1995, Theorem 2)), and our numerical experiments indicate that this greatly improves the convergence of our method. It is not always necessary to keep track of the whole chain and proposals, quantities that can become computationally expensive for high-dimensional distributions. For example, in the case of random walk Metropolis and ESJD objective function, it is enough to keep track of the jumped distance in covariance norm and acceptance probability to construct the adaptive empirical estimator. We discuss these issues further in Section 3.

2.4. Choices of the objective function

We focus on optimizing the expected squared jumped distance (ESJD), which in one dimension is defined as

$$\begin{aligned} \text{ESJD}(\gamma) &= \mathbf{E}_{J_\gamma}\left[|\theta_{t+1} - \theta_t|^2\right] = \mathbf{E}_{J_\gamma}\left[\mathbf{E}_{J_\gamma}\left[|\theta_{t+1} - \theta_t|^2 \middle| (\theta_t, \theta^*_t)\right]\right] \\ &= \mathbf{E}_{J_\gamma}\left[\Delta_t^2 \cdot \alpha_\gamma(\theta_t, \theta^*_t)\right] = 2(1 - \rho_1) \cdot \text{Var}_\pi(\theta_t) \end{aligned}$$

and corresponds to the objective function $H(\gamma, \theta_t, \theta_t^*) = \Delta_t^2 \cdot \alpha_\gamma(\theta_t, \theta_t^*)$. Maximizing the ESJD is equivalent to minimizing first-order autocorrelation, which is a convenient approximation to maximizing efficiency as we have discussed in Section 1.2.

For d-dimensional targets, we scale the expected squared jumped distance by the covariance norm and define the ESJD as

$$\text{ESJD}(\gamma) \stackrel{\triangle}{=} \mathbf{E}_{J_\gamma}\Big[\|\theta_{t+1} - \theta_t\|^2_{\Sigma^{-1}}\Big] = \mathbf{E}\Big[\|\Delta_t\|^2_{\Sigma^{-1}} \alpha_\gamma(\theta_t, \theta_t^*)\Big].$$

This corresponds to the objective function, $H(\gamma, \theta_t, \theta_t^*) = \|\Delta_t\|^2_{\Sigma^{-1}} \alpha_\gamma(\theta, \theta^*) = (\Delta_t)^t \Sigma^{-1} \Delta_t \alpha_\gamma(\theta_t, \theta_t^*)$, where Σ is the covariance matrix of the target distribution π. The adaptive estimator (2.6) then becomes

$$\hat{h}(\gamma|\gamma_k, \gamma_{k-1}, \ldots, \gamma_1) \stackrel{\triangle}{=} \frac{\sum_{i=1}^k \sum_{t=1}^{T_i} \|\Delta_{it}\|^2_{\Sigma^{-1}} \alpha_{\gamma_i}(\theta_{it}, \theta_{it}^*) \cdot w_{\gamma|\gamma_k,\ldots,\gamma_1}(\theta_{it}, \theta_{it}^*)}{\sum_{i=1}^k \sum_{t=1}^{T_i} w_{\gamma|\gamma_k,\ldots,\gamma_1}(\theta_{it}, \theta_{it}^*)}. \quad (2.8)$$

Maximizing the ESJD in covariance norm is equivalent to minimizing the lag-1 correlation of the d-dimensional process in covariance norm,

$$\text{ESJD}(\gamma) = \mathbf{E}_{J_\gamma}\Big[\|\theta_t\|^2_{\Sigma^{-1}}\Big]. \quad (2.9)$$

When Σ is unknown, we can use a current estimate in defining the objective function at each step. We illustrate this in Sections 4.2 and 4.4.

For other choices of the objective function already studied in the MCMC literature, see Andrieu and Robert (2001). In this paper we consider two optimization rules: (1) maximizing the ESJD (because of its property of minimizing the first-order autocorrelation) and (2) coercing the acceptance probability (because of its simplicity).

2.5. Convergence properties

For fixed proposal kernel, under conditions on π and J_γ such that the Markov chain (θ_t, θ_t^*) is ϕ irreducible and aperiodic (see Meyn and Tweedie (1993)), the ratio estimator $\hat{h}_T$ converges to h with probability 1. For example, if $J_\gamma(\cdot;\cdot)$ is positive and continuous on $R^d \times R^d$, and π is finite everywhere, then the algorithm is π-irreducible. No additional assumptions are necessary to ensure aperiodicity. In order to prove convergence of the maximizer of $\hat{h}_T$ to the maximizer of h, some stronger properties are required.

Proposition 1. *Let $\{(\theta_t, \theta_t^*)\}_{t=1:T}$ be the Markov chain and set of proposals generated by the Metropolis algorithm under transition kernel $J_{\gamma_0}(\cdot,\cdot)$. If the chain $\{(\theta_t, \theta_t^*)\}$ is ϕ-irreducible, and $\hat{h}_T(\cdot|\gamma_0)$ and h are concave and twice differentiable*

everywhere, then $\hat{h}_T(\cdot|\gamma_0)$ *converges to* h *uniformly on compact sets with probability 1, and the maximizers of* $\hat{h}_T(\cdot|\gamma_0)$ *converge to the unique maximizer of* h.

Proof. The proof is a consequence of well-known theorems of convex analysis stating that convergence on a dense set implies uniform convergence and consequently convergence of the maximizers; this can be found in Geyer and Thompson (1992).

In general, it is difficult to check the concavity assumption for the empirical ratio estimator, but we can prove convergence for the mean estimator.

Proposition 2. *Let* $\{(\theta_t, \theta_t^*)\}_{t=1:T}$ *be the Markov chain and set of proposals generated by the Metropolis algorithm under transition kernel* $J_{\gamma_0}(\cdot,\cdot)$. *If the chain* $\{(\theta_t, \theta_t^*)\}$ *is irreducible, and the mapping* $\gamma \to H(\gamma, x, y)J_\gamma(x, y)$, $\forall\gamma \in \Gamma$ *is continuous, and for every* $\gamma \in \Gamma$ *there is a neighborhood* B *of* γ *such that*

$$\mathbf{E}_{J_{\gamma_0}}\left[\sup_{\phi\in B} H(\phi, \theta_t, \theta_t^*)\frac{J_\phi(\theta_t, \theta_t^*)}{J_{\gamma_0}(\theta_t, \theta_t^*)}\right] < \infty, \tag{2.10}$$

then $h_T(\cdot|\gamma_0)$ *converges to* h *uniformly on compact sets with probability 1.*

Proof. See the Appendix.

The convergence of the maximizer of h_T to the maximizer of h is attained under the additional conditions of Geyer (1994).

Theorem. (Geyer (1994, Theorem 4)) *Assume that* $(\gamma_T)_T$, γ_* *are the unique maximizers of* $(h_T)_T$ *and* h, *respectively, and they are contained in a compact set. If there exist a sequence* $\epsilon_T \to 0$ *such that* $h_T(\gamma_T|\gamma_0) \geq \sup_T(h_T(\gamma_T|\gamma_0)) - \epsilon_T$, *then* $\gamma_T \to \gamma_*$.

Proposition 3. *If the chain* $\{(\theta_t, \theta_t^*)\}$ *is* ϕ*-irreducible and the objective function is the expected squared jumped distance,* $H(\gamma, x, y) = \|y - x\|^2_{\Sigma^{-1}}\alpha_\gamma(x, y)$, *then the mean empirical estimator* $h_T(\gamma|\gamma_0)$ *converges uniformly on compact sets for the case of random walk Metropolis algorithm with proposal kernel* $J_{\gamma,\Sigma}(\theta_*, \theta) \approx \exp(-\|\theta - \theta^*\|^2_{\Sigma^{-1}}/(2\gamma^2))$.

Proof. See the Appendix.

Remark. We used both the mean and the ratio estimator for our numerical experiments, but the convergence appeared to be faster than Andrieu and Robert, and the estimates more stable for the ratio estimator (see Remark 1 below for more details).

The infinite adaptation version can be proved to converge under some additional restriction, Haario, Saksman and Tamminen (2001) and Roberts and Rosenthal (2006).

Proposition 4. *Let π be a distribution function with a compact support $S \in R^d$. Consider the infinite adaptation version of the algorithm with Gaussian proposal kernel (see Section 3) with $T_n \uparrow \infty$. Then the Markov chain is ergodic and the Weak Law of Large Numbers holds for any bounded measurable function.*

Proof. The proof is a consequence of Proposition 1 and Roberts and Rosenthal (2006) Corollary 11 and remarks. By definition π has a compact support and Γ is a compact set, and we can take λ as the finite Lebesgue measure restricted to the the product space $S \times \Gamma$. The proposal kernels are multivariate normals, which ensures that for fixed $\gamma \in \Gamma$, P_γ is ergodic for $\pi(\cdot)$, and that the density is continous and bounded. By Proposition 1 for large n, γ_n converges to γ^* a.s., which ensures that the diminishing adaptation condition of Theorem 5 is satisfied. Since empirical estimates of the covariance matrix change at the nth iteration by only $O(1/n)$, it follows that the diminishing adaptation condition is satisfied for the covariance matrix. Compact support and convergence of the parameter also ensures that simultaneous uniform ergodicity condition from Theorem 5 holds which, in conjunction with Theorem 23 (Weak Law of Large Numbers), concludes the proof. Clearly, the ESJD is uniformly bounded, thus the Law of Large Numbers applies and, for T_n sufficiently large, γ_n converges to γ^*.

Remark. One can also choose the adaptation times T_n to be the regeneration times of the Markov chain (identified by enlarging the state space with an atom) as demonstrated in Brockwell and Kadane (2005). This guarantees the convergence of the estimators at the $\sqrt{n}$ rate.

2.6. Practical optimization issues

Remark 1. The ratio estimator (2.3) preserves the range of the objective function $H(\cdot)$, and has a lower variance than the mean estimator if the correlation between the numerator and denominator is sufficiently high (see Cochran (1977)). Other choices for the empirical estimator include the mean estimator h_T and estimators that use control variates that sum to 1 to correct for the bias (see, for example, the regression and difference estimators of Hesterberg (1995)).

Multiple importance sampling attempts to give high weights to individual proposal kernels that are close to the optimum. For more choices for the multiple importance sampling weights, see Veach and Guibas (1995).

Remark 2. For the usual symmetric kernels (e.g., normal, Student-t, Cauchy) and objective functions it is straightforward to derive analytical first and second order derivatives and run a few steps of a maximization algorithm that incorporates the knowledge of the first and second derivative (see, e.g., Press et al. (2002), for C code) or as already incorporated in the R function optim(). If

analytic derivatives do not exist or they are expensive to compute, then a grid maximization centered on the current optimal estimated is recommended.

Remark 3. Guidelines that ensure fast convergence of the importance sampling estimator $I_n(h) = \sum_{i=1}^n h(X_i)[(g_\gamma(X_i))/(g_{\gamma_0}(X_i))]$ of $I(h) = \mathbf{E}_{g_\gamma}[h(X)]$ based on the proposal distribution $g_{\gamma_0}(\cdot)$ are presented in Robert and Casella (1998): the importance sampling distribution g_{γ_0} should have heavier tails than the true distribution; minimizing the variance of importance weights minimizes the variance of $I_n(h)$.

3. Implementation with Gaussian Kernel

For the case of a random walk Metropolis with Gaussian proposal density $J_{\gamma,\Sigma}(\theta_*, \theta) \approx \exp[-(\|\theta - \theta^*\|^2_{\Sigma^{-1}})/(2\gamma^2)]$, the adaptive empirical estimator (2.8) of the ESJD is

$$\hat{h}(\gamma|\gamma_k, \gamma_{k-1}, \ldots, \gamma_1) \triangleq \frac{\sum_{i=1}^k \sum_{t=1}^{T_i} \|\Delta_{it}\|^2_{\Sigma_i^{-1}} \alpha(\theta_{it}, \theta^*_{it}) \cdot w_{\gamma|\gamma_k,\ldots,\gamma_1}(\|\Delta_{it}\|^2_{\Sigma_i{}^{-1}})}{\sum_{i=1}^k \sum_{t=1}^{T_i} w_{\gamma|\gamma_k,\ldots,\gamma_1}(\|\Delta_{it}\|^2_{\Sigma_i^{-1}})}$$

where

$$w_{\gamma|\gamma_k,\ldots,\gamma_1}(x) = \frac{\exp(-x/(2\gamma^2))/\gamma^d}{\sum_{i=1}^k T_i \exp(-x/2\gamma_i^2)/\gamma_i^d}. \tag{3.1}$$

For computational purposes, we program the Metropolis algorithm so that it gives as output the proposed jumping distance in covariance norm $\|\Delta_{it}\|_{\Sigma_i^{-1}}$ and the acceptance probability. This reduces the memory allocation for the optimization problem to one dimension, and the reduction is extremely important for high dimensions where the alternative is to store $d \times T$ arrays. We give here a version of our optimization algorithm that keeps track only of the jumped distance in covariance norm, the acceptance probability, and the sample covariance matrix.

1. Choose a starting covariance matrix Σ_0 for the Metropolis algorithm, for example a numerical estimation of the covariance matrix of the target distribution.
2. Choose starting points for the simulation and some initial scaling for the proposal kernel, for example $c_d = 2.38/\sqrt{d}$. Run the algorithm for T_1 iterations, saving the simulation draws θ_{1t}, the proposed jumping distances $\|\Delta_{1t}\|_{\Sigma_0^{-1}}$ in covariance norm, and the acceptance probabilities $\alpha(\theta_{1t}, \theta^*_{1t})$. Optionally, construct a vector consisting of the denominator of the multiple importance sampling weights and discard the sample θ_{1t}.

3. For $k > 1$, run the Metropolis algorithm using proposal kernel $J_{\gamma_k \Sigma_k}$. Update the covariance matrix using the iterative procedure

$$\Sigma_{k+1}(i,j) = \left(1 - \frac{T_k}{T_{total}}\right)\Sigma_k(i,j) + \frac{1}{T_{total}}\left((T_{total} - T_k)\bar{\theta}_{k-1,i}\bar{\theta}_{k-1,j} - T_{total}\bar{\theta}_{ki}\bar{\theta}_{kj} + \sum_{t=1}^{T_k}\theta_{kt}\theta_{jt}\right)$$

where $T_{total} = T_1 + \cdots + T_k$, and update the scaling using the adaptive algorithm. We must also keep track of the d-dimensional mean, but this is not difficult since it satisfies a simple recursion equation. Optionally, iteratively update the denominator of the multiple sampling weights.

4. Discard the sample θ_{kt} and repeat the above step.

In updating the covariance matrix we can also use the greedy-start procedure using only the accepted jumps (see Haario et al. (1999)). For random walk Metropolis, analytic first and second order derivatives are helpful in the implementation of the optimization step (2) (e.g., using a optimization method), and can be derived analytically. In our examples, we have had success updating the proposal kernel every 50 iterations of the Metropolis algorithm, until approximate convergence.

4. Examples

In our first three examples we use targets and proposals for which optimal proposal kernels have been proposed in the MCMC literature to demonstrate that our optimization procedure is reliable. We then apply our method on two applications of Bayesian inference using Metropolis and Metropolis within Gibbs updates.

4.1. Independent normal target distribution, $d = 1, \ldots, 100$

We begin with the multivariate normal target distribution in d dimensions with identity covariance matrix, for which the results from Gelman, Roberts and Gilks (1996) and Roberts, Gelman and Gilks (1997) regarding the choice of optimal scaling apply. This example provides some guidelines regarding the speed of convergence, the optimal sample size, and the effectiveness of our procedure for different dimensions. In the experiments we have conducted, our approach outperforms the stochastic Robbins-Monro algorithm, as implemented by Atchadé and Rosenthal (2003). These are well-known recursive algorithms, used to solve an equation $h(\gamma) = 0$ where h is unknown, but can be estimated with a noise.

Figure 6.1 shows the convergence to the maximum of the objective function of the adaptive optimization procedure for dimensions $d = 1$, 10, 25, 50, and

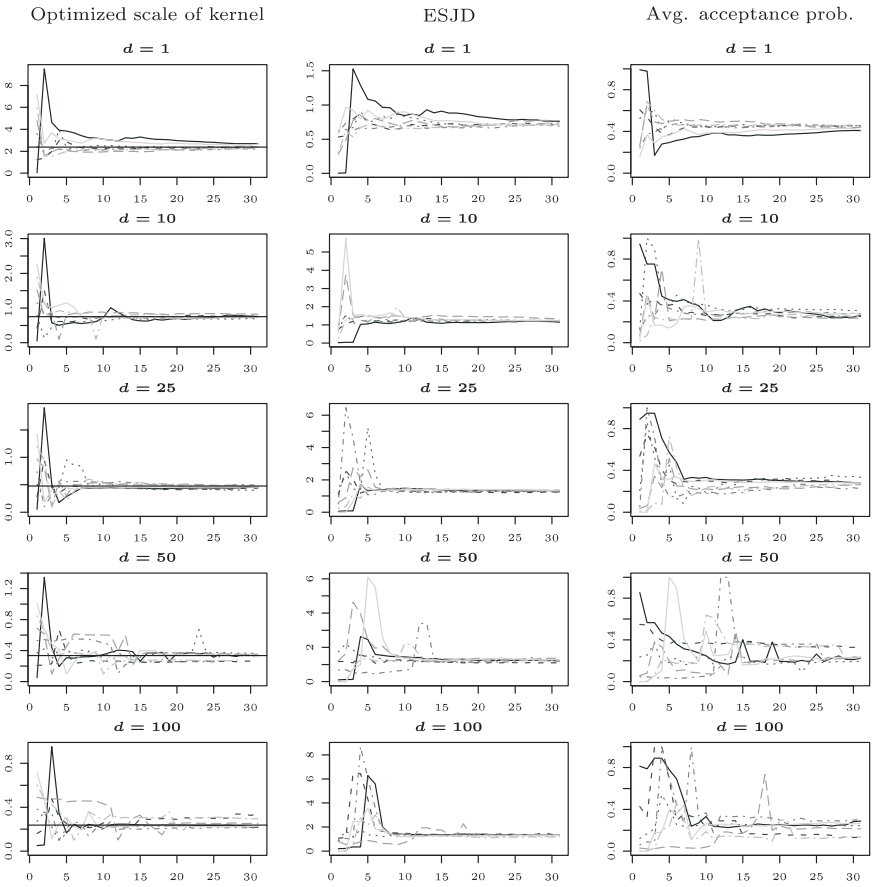

Figure 6.1. Convergence to the optimal value (solid horizontal line) at each step k of the adaptive optimization procedure, given seven equally spaced starting points in the interval $[0, 3 * 2.38/\sqrt{d}]$, $T = 50$ iterations per step, for the random walk Metropolis algorithm with multivariate standard normal target of dimensions $d = 1$, 10, 25, 50, and 100. The second and third column of figures show the multiple importance sampling estimator of ESJD and average acceptance probability, respectively.

100, as well as the corresponding values of the multiple importance sampling estimator of ESJD and average acceptance probability.

When starting from very small values, the estimated optimal scale shows some initial high upward jumps. In order to eliminate high amplitude jumps and slow convergence, the optimization could be restricted to the set where the variance of the importance sampling weights is finite:

$$\left\{\gamma > 0 | \gamma^2 < 2 \max_{i=1:k} \gamma_i^2\right\}. \tag{4.1}$$

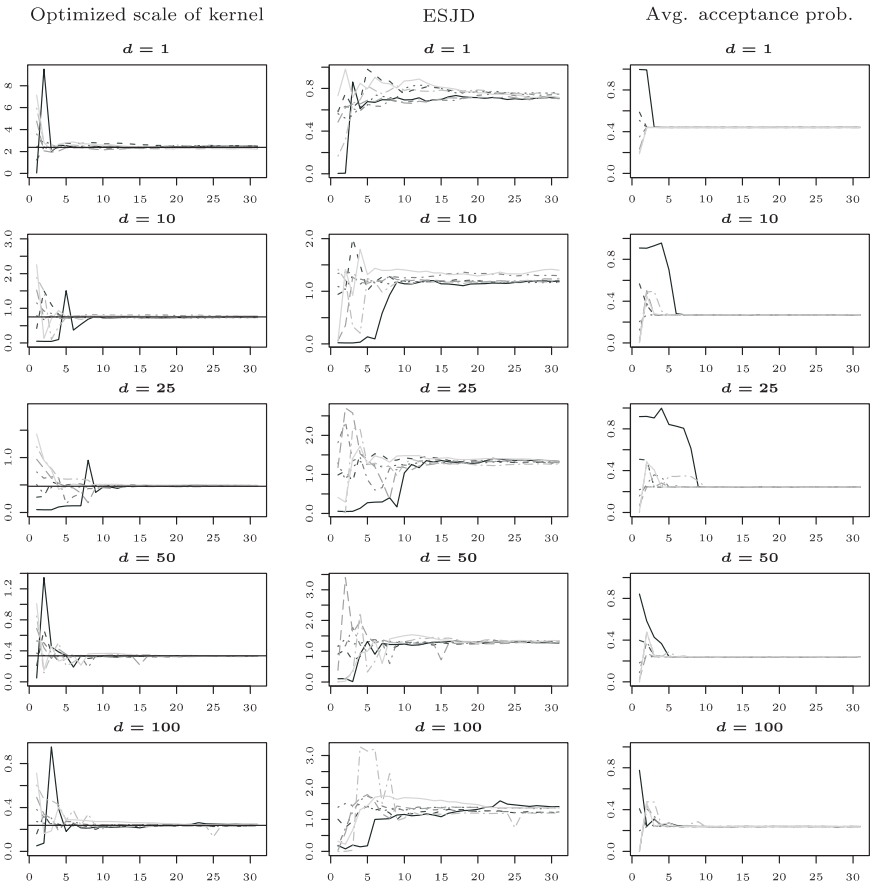

Figure 6.2. Convergence of the adaptive optimization procedure using as objective the coerced average acceptance probability (to the optimal acceptance value from Figure 6.1). The second and third column show the multiple importance sampling estimator of the ESJD and average acceptance probability, respectively. Convergence of the optimal scale is faster than optimizing ESJD, although not necessarily to the most efficient jumping kernel (see Figure 6.3).

In order to compare our algorithm with the stochastic Robbins-Monro algorithm, we have also coerced the acceptance probability by estimating the average acceptance probability using the objective function $H(x, y) = \alpha_\gamma(x, y)$ and then minimizing a quadratic loss function $h(\gamma) = (\int\int \alpha_\gamma(x, y) J_\gamma(x, y) dy dx - \alpha_*)^2$, where α_* is defined as the acceptance rate corresponding to the Gaussian kernel that minimizes the first-order autocorrelation.

The convergence of the algorithm coercing the acceptance probability method is faster than maximizing ESJD, which we attribute to the fact that the accep-

tance probability is less variable than ESJD, thus easier to estimate.

A comparison of our method with the stochastic Robbins-Monro algorithm implemented by Atchadé and Rosenthal (2003, Graph 2), shows that our method converges faster and does not encounter the problems of the stochastic algorithm which always goes in the first steps to a very low value and then converges from below to the optimal value. It is generally better to overestimate than to underestimate the optimal scaling. We have also successfully tested the robustness of our method for extreme starting values(see online Appendix Section 2) and correlated normal distribution (see online Appendix Section 1).

4.2. Mixture target distribution

We consider now a mixture of Gaussian target distributions with parameters $\mu_1 = -5.0$, $\sigma_1^2 = 1.0$, $\mu_2 = 5.0$, $\sigma_2^2 = 2.0$ and weights $(\lambda = 0.2, 1-\lambda)$. The purpose of this example is two-fold: first to illustrate that for a bimodal distribution, where the optimal scaling is different from $c_d = 2.38/\sqrt{d}$ (which holds for targets with i.i.d. components), our method of tuning ESJD is computationally feasible and produces better results. Second, to compare our method with the stochastic Robbins-Monro algorithm of Andrieu and Robert (2001, Sec. 7.1) where the acceptance probability is coerced to 40%.

We compare the results of our method given two objective functions, coercing the acceptance probability to 44% and maximizing the ESJD, in terms of convergence and efficiency. We also compare the speed of the stochastic Robbins-Monro algorithm with the convergence speed of our adaptive optimization procedure.

The convergence to the "optimal" acceptance probability for the coerced probability method is attained in 1,000 iterations for all starting values, an improvement over the approximately 10,000 iterations required under the stochastic optimization algorithm (see Andrieu and Robert (2001, Figure 6)). Maximizing ESJD yields an optimal scaling of $\gamma = 9$, and a comparison of the correlation structure ρ_t (the bottom two graphs of Figure 6.3), at the optimal scale determined by the two objective functions, shows that the autocorrelation decreases much faster for the optimal scale that maximizes ESJD, thus making the ESJD a more appropriate efficiency measure.

4.3. 16-dimensional nonlinear model

We next consider an applied example, a model for serial dilution assays from Gelman, Chew and Shnaidman (2004),

$$y_i \sim N\left(g(x_i,\beta), \left(\frac{g(x_i,\beta)}{A}\right)^{2\alpha}\sigma_y^2\right)$$
$$x_i = d_i \cdot x_j^{init}(i),$$

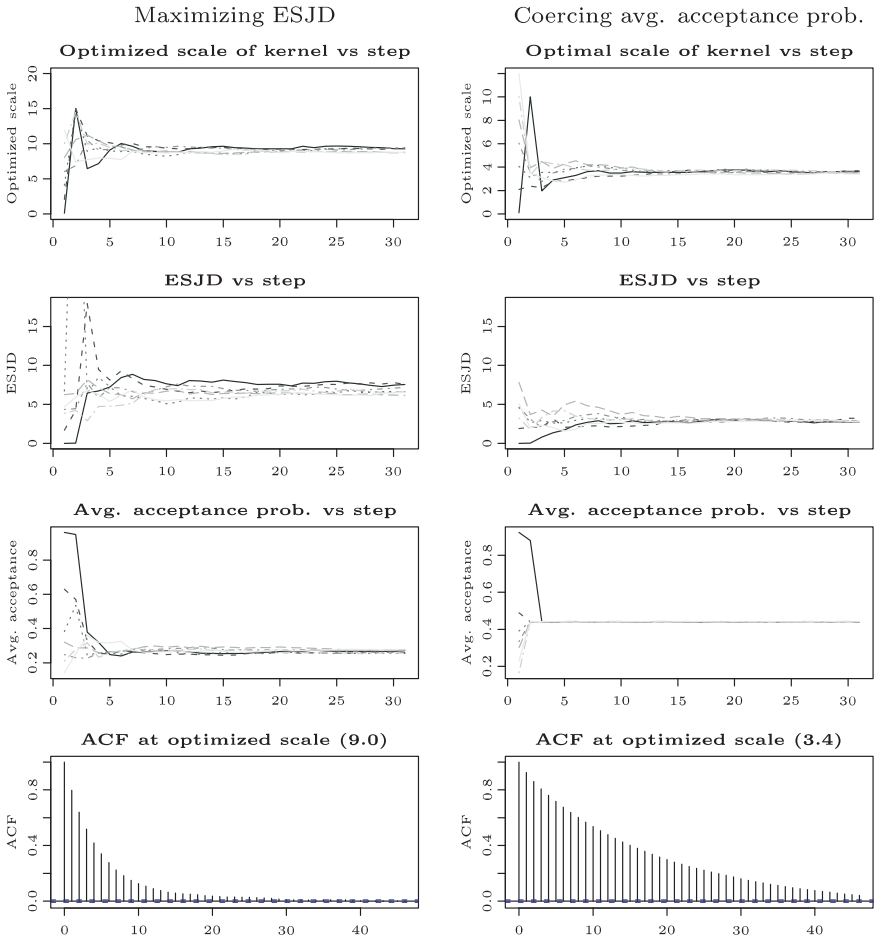

Figure 6.3. Comparison of two objective functions for the 2-component mixture target of Andrieu and Robert (2001) using our adaptive optimization algorithm: maximizing ESJD (left column of plots), and coercing the acceptance probability to 44% (right column of plots), with 50 iterations per step. The coerced acceptance probability method converges slightly faster but to a less efficient kernel (see ACF plot).

where y_i are data points(intensities of color changes in a laboratory essay), x_i are concentrations of a compound of interest, and $g(x, \beta) = \beta_1 + [\beta_2/(1+(x/\beta_3)^{-\beta_4})]$. For each sample j, we model

$$\log x_j^{init} \sim N\Big(\log(d_j^{init} \cdot \theta_j), (\sigma^{init})^2\Big) \quad \text{for the calibration sample,} \quad j = 0$$

$$x_j^{init} = \theta_j \quad \text{for the samples with unknown concentrations,} \quad j = 1, \ldots, 10.$$

The constant A is arbitrary and is set to some value in the middle of the range of

the data. The parameter σ^{init} is assumed known, and a vague prior distribution is applied to σ_y and β. We estimate the unknown concentrations θ_j using data y_i from a single plate with 16 calibration measurements and 8 measurements per unknown sample. We know the initial concentration of standard sample θ_0 and the dilution d_i, and we need to estimate the 10 unknown concentrations θ_j and the parameters $\beta_1, \beta_2, \beta_3, \beta_4, \sigma_\theta, \sigma_y, \alpha$. For faster convergence the θ_i's are reparameterized as $\log \eta_i = \log \theta_j - \log \beta_3$. We use the Broyden-Fletcher-Goldfarb-Shanno algorithm (e.g., Press et al. (2002)), to find the maximum likelihood, estimate and start the Metropolis with a Gaussian proposal with the covariance set to the inverse of the Hessian of the log likelihood computed in the maximum. We keep the covariance matrix fixed and optimize only the choice of scaling. After the algorithm converges to the maximum ESJD, we verify that the sample covariance matches the choice of our initial covariance. Despite the complex structure of the target distribution, the adaptive method converges to the theoretical optimal value $c_d \approx 2.4/\sqrt{16} = 0.6$ in k =30 steps with T_i =50 iterations per step.

The computation time is 0.01 seconds per iteration in the Metropolis step, and the optimization step takes an average 0.04 seconds per step. We update after every 50 iterations and so the optimization adds $0.04/(50 * 0.01)$, or 8%, to the computing time.

4.4. Hierarchical student-t model

Finally, we apply our method to Metropolis within Gibbs sampling with a hierarchical Student-t model applied to the educational testing example from Gelman, Carlin, Stern and Rubin (2003, Appendix C). The model has the form,

$$\begin{aligned} &y_j \sim \mathrm{N}(\theta_j, \sigma_j^2), \quad \sigma_j \text{ known, for } \quad j = 1, \ldots, 8, \\ &\theta_j | \nu, \mu, \tau \sim t_\nu(\mu, \tau^2) \text{ for } \quad j = 1, \ldots, 8, \end{aligned}$$

where for each of eight schools j, y_i is an unbiased estimate of the effect of a scholastic aptitude test coaching program in the school, θ_j is the true effect in the school, and the effects are modeled hierarchically. We use an improper joint uniform prior density for $(\mu, \tau, 1/\nu)$. To treat ν as an unknown parameter, the Gibbs sampling simulation includes a Metropolis step for sampling from the conditional distribution of $1/\nu$. Maximizing ESJD, the adaptive procedure converges to the optimal scale $\gamma = 0.5$ in 10 steps of 50 iterations each, the same optimal value for coercing the acceptance probability to 44%.

5. Discussion

The proposed adaptive method is computationally easy to implement, and maximizing ESJD greatly improves the performance of the Metropolis algorithm. Our algorithm follows similar steps as recent work in adaptive updating of the

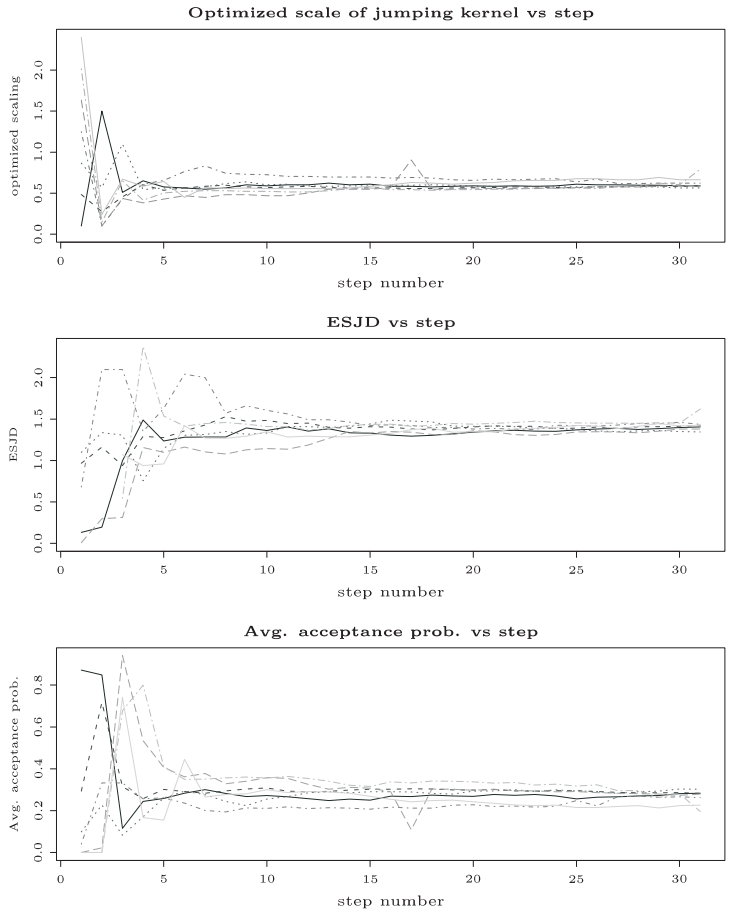

Figure 6.4. 16-dimensional nonlinear model for a serial dilution experiment of Gelman, Chew and Shnaidman (2004); convergence to optimal scaling, for seven equally spaced starting values in $[0, 2.4]$ with 50 iterations per step and covariance matrix determined by initial optimization.

Metropolis kernel (Haario et al. (1999), Andrieu and Robert (2001), and Atchadé and Rosenthal (2003)), but appears to converge faster, presumably because of the numerical stability of the multiple importance sampling estimate in the context of a Gaussian parametric family of proposal kernels. Coercing the acceptance probability has slightly faster convergence than maximizing the ESJD, but not necessarily to an optimal value as we have seen in Figure 6.3. For Gaussian and independent distributions in high dimensions, samples of the Metropolis algorithm approach an Ornstein-Uhlenbeck process and all reasonable optimization criteria are equivalent (Roberts, Gelman and Gilks (1997)), but this is not necessarily the case for finite-dimensional problems or adaptive algorithms.

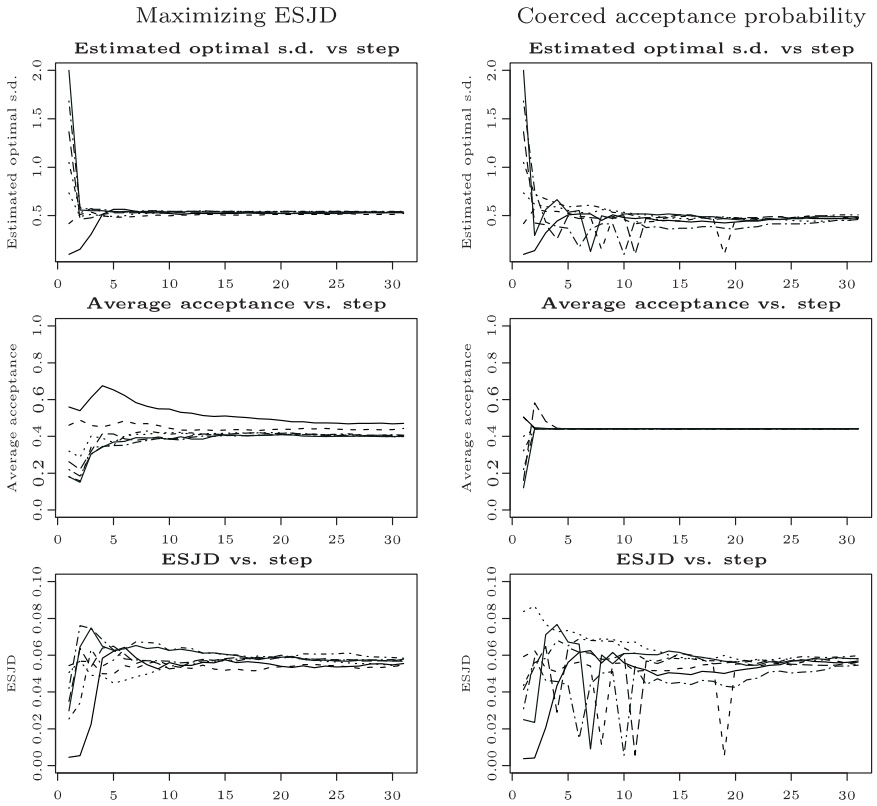

Figure 6.5. Gibbs sampling with a Metropolis step for the inverse of the degrees of freedom for the hierarchical t model for the eight schools example of Gelman et al. (2003); convergence of optimal scaling given starting values in $[0, 2]$ for two objective functions: maximizing ESJD (left column of plots) and coercing average acceptance probability to 44% (right column of plots).

Other issues that arise in setting up the algorithm are the choice of multiple sampling weights, the choice of number of iterations per step, and when to stop the adaptation. In high-dimensional problems, we have optimized the scale of the proposal kernel while updating the covariance matrix using empirical weighting of posterior simulations (as in Haario et al. (1999)). We also anticipate that these methods can be generalized to optimize over more general MCMC algorithms, for example slice sampling (Neal (2003)) and Langevin algorithms which could achieve higher efficiencies than symmetric Metropolis algorithms (see Roberts and Rosenthal (2001)).

Acknowledgement

We thank several reviewers for helpful comments and the National Science

Foundation for financial support.

Appendix

Proof of Proposition 2. The process $\{(\theta_t, \theta_t^*)\}$ is a positive recurrent Markov chain with invariant probability $\pi(dx)J_\gamma(x, dy)$. By assumption, θ_t is irreducible and thus satisfies the conditions of Robert and Casella (1998, Theorem 6.2.5 i); consequently,

$$h_T(\gamma|\gamma_0) = \frac{1}{T}\sum_{t=1}^{T} H(\gamma, \theta_t, \theta_t^*) w_{\gamma|\gamma_0}(\theta_t, \theta_t^*) \tag{A.1}$$
$$\to \int\int H(\gamma, x, y)\pi(x)J_\gamma(x, y)dxdy, \ a.s., \ \forall \gamma \in \Gamma.$$

The next part of the proof is a particular version of Geyer (1994, Theorems 1 and 2), and we reproduce it here for completeness. Taking into account that the union of null sets is a null set, we have that (A.1) holds *a.s.* for all γ in a countable dense set in Γ. By the weak convergence of measures,

$$\inf_{\phi\in B} \frac{1}{T}\sum_{t=1}^{T} H(\phi, \theta_t, \theta_t^*) \cdot w_{\phi|\gamma_0}(\theta_t, \theta_t^*) \to \int\int \inf_{\phi\in B} H(\phi, x, y)\pi(x)J_\phi(x, y)dxdy, \ a.s.$$

holds, for all γ in a countable dense set in Γ. Convergence on compact sets is a consequence of epiconvergence and hypoconvergence (see, for example, Geyer (1994)).

Proof of Proposition 3. We need to prove that the assumptions of Proposition 2 are verified. Clearly the continuity assumption is satisfied, and we now check (2.10). For simplicity, we omit the subscript and use the notation $\|\| = \|\|_{\Sigma^{-1}}$. Fix $\gamma > 0$ and $\epsilon > 0$ small enough,

$$\begin{aligned}
&\int\int \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} \left(\|y-x\|^2 \frac{J_\phi(\|y-x\|^2)}{J_{\gamma_0}(\|y-x\|^2)}\alpha(x, y) \right) J_{\gamma_0}(\|y-x\|^2)\pi(x)dydx \\
&\quad = \int\int \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} \Big(J_\phi(\|y-x\|^2) \Big) \|y-x\|^2 \alpha(x, y)\pi(x)dydx \\
&\quad \le \int \left(\int_{d(\gamma-\epsilon)^2<\|y-x\|^2<d(\gamma+\epsilon)^2} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} \Big(J_\phi(\|y-x\|) \Big) \|y-x\|^2 dy \right) \pi(x)dx \\
&\qquad + \int \left(\int_{\|y-x\|^2\notin(d(\gamma-\epsilon)^2,\ d(\gamma+\epsilon)^2)} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} J_\phi(\|y-x\|)\|y-x\|^2 dy \right) \pi(x)dx.
\end{aligned}$$

Taking into account that

$$\sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} \frac{1}{\phi^d}\exp\left\{-\frac{\|y-x\|^2}{2\phi^2}\right\} = \begin{cases} K\frac{1}{\|y-x\|^2}, & \|y-x\|^2 \in (d(\gamma-\epsilon)^2,\ d(\gamma+\epsilon)^2) \\ J_{\gamma-\epsilon}(\|y-x\|^2), & \|y-x\|^2 \le d(\gamma+\epsilon)^2 \\ J_{\gamma+\epsilon}(\|y-x\|^2), & \|y-x\|^2 \ge d(\gamma-\epsilon)^2 \end{cases}$$

with $K>0$, the first integral becomes

$$\begin{aligned} &\int\left(\int_{d(\gamma-\epsilon)^2<\|y-x\|^2<d(\gamma+\epsilon)^2} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} \Big(J_\phi(\|y-x\|)\Big)\|y-x\|^2dy\right)\pi(x)dx \\ &\le K\int\left(\int_{0<\|y-x\|^2<d(\gamma+\epsilon)^2} \frac{1}{d(\gamma-\epsilon)^2}dy\right)\pi(x)dx \\ &= K\int_{0<\|z\|^2<d(\gamma+\epsilon)^2} \frac{1}{d(\gamma-\epsilon)^2}dz<\infty, \end{aligned} \tag{A.2}$$

and the second integral can be bounded as follows:

$$\begin{aligned} &\int\left(\int_{\|y-x\|^2\notin(d(\gamma-\epsilon)^2,d(\gamma+\epsilon)^2)} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} J_\phi(\|y-x\|)\|y-x\|^2dy\right)\pi(x)dx \\ &= \int\left(\int_{\|y-x\|^2\le d(\gamma-\epsilon)^2} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} J_\phi(\|y-x\|)\|y-x\|^2dy\right)\pi(x)dx \\ &\quad+\int\left(\int_{\|y-x\|^2\ge d(\gamma+\epsilon)^2} \sup_{\phi\in(\gamma-\epsilon,\gamma+\epsilon)} J_\phi(\|y-x\|)\|y-x\|^2dy\right)\pi(x)dx \\ &= \left(\int_{\|y-x\|^2\le d(\gamma-\epsilon)^2} J_{\gamma-\epsilon}(\|y-x\|)\|y-x\|^2dy\right)\pi(x)dx \\ &\quad+\int\left(\int_{\|y-x\|^2\ge d(\gamma+\epsilon)^2} J_{\gamma+\epsilon}(\|y-x\|)\|y-x\|^2dy\right)\pi(x)dx<\infty. \end{aligned} \tag{A.3}$$

Combining (A.2) and (A.3) proves (2.10).

References

Andrieu, C. and Robert, C. P. (2001). Controlled MCMC for optimal sampling. Technical report, Université Paris-Dauphine.

Atchadé, Y. F. and Rosenthal, J. S. (2003). On adaptive Markov chain Monte Carlo algorithms. Technical report, University of Montreal.

Bédard, M. (2006). On the robustness of optimal scaling for Random Walk Metropolis algorithm, Ph.D. thesis, University of Toronto.

Brockwell, A. E. and Kadane, J. B. (2005). Identification of regeneration times in MCMC Simulations, with application to adaptive schemes. *J. Comput. Graph. Statist.* **14**, 436-458.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *J. Roy. Statist. Soc. Ser. B* **55**, 25-37.

Cochran, W. G. (1977). *Sampling Techniques*, third edition. Wiley, New York.

Gelman, A., Roberts, G. O. and Gilks, W. R. (1996). Efficient Metropolis jumping rules. *Bayesian Statistics* **5**, 599-608.

Gelman, A., Chew, G. L. and Shnaidman, M. (2004). Bayesian analysis of serial dilution assays. *Biometrics* **60**, 407–417.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2003). *Bayesian Data Analysis*, second edition. Chapman and Hall, London.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *J. Roy. Statist. Soc. Ser. B* **56**, 261-274.

Geyer, C. J. (1996). Estimation and optimization of functions. In *Markov Chain Monte Carlo in Practice* (Edited by W. R. Gilks, S. Richardson and D. J. Spiegelhalter), 241-258. Chapman and Hall, London.

Geyer, C. J. and Thompson, E. A. (1992). Constrained maximum Monte Carlo maximum likelihood for dependent data (with discussion). *J. Roy. Statist. Soc. Ser. B* **54**, 657-699.

Gilks, W. R., Roberts, G. O. and Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration. *J. Amer. Statist. Assoc.* **93**, 1045-1054.

Haario, H., Saksman, E. and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Comput. Statist.* **14**, 375-395.

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

Haario, H., Laine, M., Mira, A. and Saksman, E. (2006). DRAM: Efficient adaptive MCMC. *Statist. Comput.* **16**, 339-354.

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

Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distribution. *Technometrics* **37**, 185-194.

Holden, L. (2000). Adaptive chains. Technical report, Norwegian Computing Centre.

Kim, S., Shephard, N. and Chib, J. B. (1998). Stochastic volatility; likelihood inference and comparison with ARCH models. *Rev. Econom. Stud.* **65**, 361-393.

Kushner, H. J. and Yin, G. G. (2003). *Stochastic Approximation Algorithms and Applications*, second edition. Springer-Verlag, New York.

Laskey, K. B. and Myers, J. (2003). Population Markov chain Monte Carlo. *Machine Learning*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. and Teller, E. (1953). Equations of state calculations for fast computing machines. *Journal of Chemical Physics* **21**, 1087-1092.

Meyn, S. P. and Tweedie, R. (1993). *Markov Chains and Stochastic Stability*. Springer-Verlag, New York.

Neal, R. M. (2003). Slice sampling (with discussion). *Ann. Statist.* **31**, 705-767.

Press, A., Teukolski, S., Vetterling, W. and Flannery, B. (2002). *Numerical Recipes*. Cambridge University Press.

Robert, C. P. and Casella, G. (1998). *Monte Carlo Statistical Methods*. Springer-Verlag, New York.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of the random walk Metropolis algorithms. *Ann. Appl. Probab.* **7**, 110-220.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statist. Sci.* **16**, 351-367.

Roberts, G. and Rosenthal, J. S. (2006). Coupling and Ergodicity of Adaptive MCMC.

Tierney, L. and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. *Statist. Medicine* **18**, 2507-2515.

Veach, E. and Guibas, L. (1995). Optimally combining importance sampling techniques for Monte Carlo rendering. *SIGRAPH'95 Conference Proceedings*, 419-428.

J. P. Morgan 270 Park Ave, New York, NY 10017, U.S.A.

E-mail: cristian.pasarica@jpmorgan.com

Department of Statistics, Columbia University, New York, NY10027-5904, U.S.A.

E-mail: gelman@stat.columbia.edu

(Received April 2006; accepted December 2008)