Adaptively scaling the Metropolis algorithm using expected squared jumped distance*

Cristian Pasarica † Andrew Gelman‡

May 19, 2004

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 under its d-dimensional stationary distribution. The adaptive algorithm uses the information accumulated by a single path and adapts the choice of the parametric kernel in the direction of the local maximum of the objective function using multiple importance sampling techniques. We demonstrate the effectiveness of our method in several examples.

Key Words. Acceptance rates; Bayesian computation; iterative simulation; Markov chain Monte Carlo; Metropolis algorithm; multiple importance sampling

1 Introduction

1.1 Adaptive MCMC algorithms: motivation and difficulties

The algorithm of Metropolis et al. (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-target 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,

*We thank several reviewers for helpful comments and the National Science Foundation for financial support.
†Department of Statistics, Columbia University, New York, NY 10027, pasarica@stat.columbia.edu
‡Department of Statistics, Columbia University, New York, NY 10027, gelman@stat.columbia.edu

matching some criteria under the invariant distribution (e.g. Kim, Shephard, and Chib (1998), Haario, Saksman, and Tamminen (1999), Laskey and Myers (2001), 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 normal target distribution the optimal scaling of the jumping kernel is $c_d = 2.4/\sqrt{d}$ (Gelman, Roberts, and Gilks (1996)).

Another approach is to coerce the acceptance probability to a preset value (e.g. 44% for one-dimensional target) or to match moments; these can be difficult to apply due to the complicated form of target distribution which makes the optimal acceptance probability value or analytic moments 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 target distributions where initial optimization algorithms cannot find easily the global maximum of the target distribution, yielding a proposal covariance matrix different from the covariance matrix under the invariant distribution.

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 very restrictive conditions (see Haario, Saksman, and Tamminen (2001), Holden (1998) and Atchadé and Rosenthal (2003)). Adaptive methods that preserve the Markovian properties by using regeneration times have the challenge of estimation of regeneration times, which is difficult for algorithms other than independence chain Metropolis (see Gilks, Roberts, and Sahu (1998)). We shall follow a two-stage approach: a series of adaptive optimization steps followed by an MCMC run with fixed kernel.

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 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 (1995) for discussion). A natural measure of efficiency is the asymptotic variance of the sample mean $\bar{\theta} = \frac{1}{T}\sum_{t=1}^{T}\theta_t$. The asymptotic efficiency of Markov chain sampling for $\bar{\theta}$ is defined as

$$\mathrm{eff}_{\bar{\theta}}(\gamma) = \frac{\mathrm{var}_{(\pi)}(\bar{\theta})}{\mathrm{var}_{(J_\gamma)}(\bar{\theta})} = \left[1 + 2(\rho_1 + \rho_2 + \cdots)\right]^{-1}, \ \gamma \in \Gamma,$$

where $\mathrm{var}_{(\pi)}$ denotes the variance under independent sampling, var_{J_γ} is the limiting scale sample variance from the MCMC output, and ρ_t is the autocorrelation of the Markov chain at lag t. 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 is attained by having a low second eigenvalue. Maximizing 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 shall maximize the expected squared jumped distance (ESJD):

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

for a one-dimensional target distribution π, and a similar quantity in multiple dimensions (see Section 2.4). Clearly, $\mathrm{var}_\pi(\theta_t)$ 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). However, our method is more general and can apply to any 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 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 construct 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 Thomson (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).

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 in general and in Section 3 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 jumping (proposal) kernels. For fixed $\gamma\in\Gamma$ define

$$\alpha_\gamma(x,y)=\min\left\{\frac{J_\gamma(y,x)\pi(y)}{\pi(x)J_\gamma(x,y)},1\right\}.$$

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

$$p_\gamma(x,y)=\begin{cases} J_\gamma(x,y)\alpha_\gamma(x,y), & x\neq y\\ 0, & x=y\end{cases} \tag{1}$$

and set

$$r_\gamma(x)=1-\int p_\gamma(x,y)dy,$$

then the Metropolis transition kernel can be written as

$$\begin{aligned} K_\gamma(x,dy) &= \left(1 \wedge \frac{\pi(y)J_\gamma(y,x)}{\pi(x)J_\gamma(x,y)}\right) J_\gamma(x,dy)\mathbf{1}_{\{x\neq y\}} + \delta_x(y)\left(1-\int\left(1\wedge\frac{\pi(y)J_\gamma(y,x)}{\pi(x)J_\gamma(x,y)}\right)J_\gamma(x,y)dy\right) \\ &= p_\gamma(x,y)dy + r_\gamma(x)\delta_x(dy). \end{aligned}$$

Throughout this paper we use the notation θ_t^* for the proposal generated by the Metropolis-Hastings chain under jumping kernel $J_\gamma(\cdot,\theta_t)$ and denote by

$$\Delta_t \triangleq \theta_t^* - \theta_t,$$

the proposed jumping distance. Clearly $\theta_{t+1} = \theta_t^*$ with probability $\alpha(\theta_t,\theta_t^*)$, and $\theta_{t+1} = \theta_t$, with probability $1-\alpha(\theta_t,\theta_t^*)$.

2.2 Optimization of the jumping kernel after one set of simulations

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

$$h(\gamma) \triangleq \mathbf{E}\left[H(\gamma,\theta_t,\theta_t^*)\right] = \iint_{\mathbf{R}^d\times\mathbf{R}^d} H(\gamma,x,y)\pi(x)J_\gamma(x,y)dx\,dy,\ \forall\gamma\in\Gamma. \tag{2}$$

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

$$\hat{h}_T(\gamma|\gamma_0) \triangleq \frac{\sum_{t=1}^T H(\gamma,\theta_t,\theta_t^*)\cdot w_{\gamma|\gamma_0}(\theta_t,\theta_t^*)}{\sum_{t=1}^T w_{\gamma|\gamma_0}(\theta_t,\theta_t^*)}, \forall\gamma\in\Gamma, \tag{3}$$

or the mean estimator

$$h_T(\gamma|\gamma_0) \triangleq \frac{1}{T}\sum_{t=1}^T H(\gamma,\theta_t,\theta_t^*)\frac{J_\gamma(\theta_t,\theta_t^*)}{J_{\gamma_0}(\theta_t,\theta_t^*)}, \forall\gamma\in\Gamma. \tag{4}$$

where

$$w_{\gamma|\gamma_0}(x,y) \triangleq \frac{J_\gamma(x,y)}{J_{\gamma_0}(x,y)}, \tag{5}$$

are the importance sampling weights. On the left side of (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 $H(\gamma, (\theta, \theta^*)) = \|\theta - \theta^*\|^2_{\Sigma^{-1}} \alpha_\gamma(\theta, \theta^*) = (\theta - \theta^*)^t \Sigma^{-1} (\theta - \theta^*) \alpha_\gamma(\theta, \theta^*)$, where Σ is the covariance matrix of the target distribution π, because maximizing this distance is equivalent with 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 (3) using a numerical optimization algorithm such as Brent's (see, e.g., Press et al. (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 jumping kernel

If the starting point 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 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 jumping 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, $T_1 + T_2$, $T_1 + T_2 + T_3 + \cdots$. For $k = 1, 2, 3, \ldots$,

1. Run the Metropolis algorithm for T_k steps according to jumping 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 $h_{T_1+\cdots+T_k}(\gamma|\gamma_k, \ldots, \gamma_1)$, defined as

$$\hat{h}_{T_1+\cdots+T_k}(\gamma|\gamma_k, \ldots, \gamma_1) = \frac{\sum_{i=1}^k \sum_{t=1}^{T_i} H(\gamma, \theta_{it}, \theta^*_{it}) \cdot w_{\gamma|\gamma_k, \ldots, \gamma_1}(\theta_{it}, \theta^*_{it})}{\sum_{i=1}^k \sum_{t=1}^{T_i} w_{\gamma|\gamma_k, \ldots, \gamma_1}(\theta_{it}, \theta^*_{it})}, \tag{6}$$

where the *multiple importance sampling weights* are

$$w_{\gamma|\gamma_j,\ldots,\gamma_1}(\theta,\theta^*) \triangleq \frac{J_\gamma(\theta,\theta^*)}{\sum_{i=1}^{j} T_i J_{\gamma_i}(\theta,\theta^*)},\ j=1,\ldots,k. \tag{7}$$

We are treating the samples as having come from a mixture of k distributions. The weights satisfy the condition $\sum_{i=1}^{k}\sum_{t=1}^{T_i} w_{\gamma|\gamma_k,\ldots,\gamma_1}(\theta_{it},\theta^*_{it}) = 1$ and are derived from the individual importance sampling weights by substituting $J_\gamma = \omega_{\gamma|\gamma_j} J_{\gamma_j}$ in the numerator of (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. Since step 2 is nested within a larger optimization procedure, if suffices to run only a few steps of an optimization algorithm, there is no need to find the local optimum since it will be altered at next step anyway. Also, it is not always necessary to keep track of the whole chain and proposals, quantities that can become be 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 the acceptance probability to construct the adaptive empirical estimator. We further discuss these issues 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,

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

and corresponds to the objective function $H(\gamma,\theta,\theta^*) = (\theta-\theta^*)^2\alpha_\gamma(\theta,\theta^*)$. 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

$$ESJD(\gamma) \triangleq \mathbf{E}_{J_\gamma}[\|\theta_{t+1} - \theta_t\|^2_{\Sigma^{-1}}] = \mathbf{E}\left[\|\theta^*_t - \theta_t\|^2_{\Sigma^{-1}}\alpha_\gamma(\theta_t, \theta^*_t)\right].$$

This corresponds to the objective function, $H(\gamma, \theta, \theta^*) = \|\theta - \theta^*\|^2_{\Sigma^{-1}}\alpha_\gamma(\theta, \theta^*) = (\theta - \theta^*)^t\Sigma^{-1}(\theta - \theta^*)\alpha_\gamma(\theta, \theta^*)$, where Σ is the covariance matrix of the target distribution π. The adaptive estimator (6) then becomes,

$$\hat{h}_{T_1+\cdots+T_k}(\gamma \mid \gamma_k, \gamma_{k-1}, \ldots, \gamma_1) \triangleq \frac{\sum_{i=1}^k \sum_{t=1}^{T_i} \|\Delta_{it}\|^2_{\Sigma^{-1}}\alpha_{\gamma_i}(\theta_{it}, \theta^*_{it}) \cdot w_{\gamma|\gamma_k,\ldots,\gamma_1}(\theta_{it}, \theta^*_{it})}{\sum_{i=1}^k \sum_{t=1}^{T_i} w_{\gamma|\gamma_k,\ldots,\gamma_1}(\theta_{it}, \theta^*_{it})}. \quad (8)$$

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

$$ESJD(\gamma) = \mathbf{E}_{J_\gamma}\left[\|\theta_t\|^2_{\Sigma^{-1}}\right] - \mathbf{E}_{J_\gamma}\left[<\theta_{t+1}, \theta_t>_{\Sigma^{-1}}\right]. \quad (9)$$

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

For other choices of objective function in the MCMC literature, see Andrieu and Robert (2001). In this paper we shall 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 jumping kernel, under conditions on π and J_γ such that the Markov chain (θ_t, θ^*_t) is irreducible and aperiodic (see Meyn and Tweedie (1993)), the ratio estimator $\hat{h}_T$ converges to h with probability 1. 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-Hastings algorithm under transition kernel $J_{\gamma_0}(\cdot, \cdot)$. If the chain $\{(\theta_t, \theta^*_t)\}$ is irreducible, and $\hat{h}_T(\cdot \mid \gamma_0)$ and h are concave and twice differentiable everywhere, then $\hat{h}_T(\cdot \mid \gamma_0)$ converges to h uniformly on compacts 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, and 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-Hastings 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

$$\mathbf{E}_{J_{\gamma_0}} \left[\sup_{\phi \in B} H(\phi, \theta_t, \theta_t^*) \frac{J_\phi(\theta_t, \theta_t^*)}{J_{\gamma_0}(\theta_t, \theta_t^*)} \right] < \infty, \tag{10}$$

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

Proof. See 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 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 jumping kernel $J_{\gamma,\Sigma}(\theta_*, \theta) \approx \exp\left(-\frac{1}{2\gamma^2}\|\theta - \theta^*\|^2_{\Sigma^{-1}}\right)$.

Proof. See Appendix.

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

2.6 Practical optimization issues

Remark 1. The motivation for the ratio estimator (3) is that it preserves the range of the objective function, for example constraining the acceptance probability to the range $[0,1]$, 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 bias (see, for example, the regression and difference estimators of Hesterberg (1995)).

Multiple importance sampling is intended to give high weights to individual jumping kernels that are near 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, 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 which incorporates the knowledge of the first and second derivative (see, e.g., Press et al. (2002) for C code or the function `optim()` in `R`). If analytic derivatives do not exist or are expensive to compute, then pne can perform a grid maximization centered on the current estimated optimum.

Remark 3. Guidelines that ensure fast convergence of the importance sampling estimator $I_n(h) = \sum_{i=1}^{n} h(X_i)\frac{g_\gamma(X_i)}{g_{\gamma_0}(X_i)}$ 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 then 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 algorithm with Gaussian proposal density $J_{\gamma,\Sigma}(\theta_*,\theta) \approx \exp\left(-\frac{1}{2\gamma^2}\|\theta-\theta^*\|^2_{\Sigma^{-1}}\right)$, the adaptive empirical estimator (8) of the ESJD is

$$\hat{h}_{T_1+\cdots+T_k}(\gamma\,|\gamma_k,\gamma_{k-1},\ldots,\gamma_1) \triangleq \frac{\sum_{i=1}^{k}\sum_{t=1}^{T_i}\|\Delta_{it}\|^2_{\Sigma_i^{-1}}\alpha(\theta_{it},\theta^*_{it})\cdot w_{\gamma|\gamma_k,\ldots,\gamma_1}(\|\Delta_{it}\|^2_{\Sigma_i^{-1}})}{\sum_{i=1}^{k}\sum_{t=1}^{T_i} w_{\gamma|\gamma_k,\ldots,\gamma_1}(\|\Delta_{it}\|^2_{\Sigma_i^{-1}})}, \tag{11}$$

where

$$w_{\gamma|\gamma_k,\ldots,\gamma_1}(x) = \frac{\frac{1}{\gamma^d}\exp\left(-\frac{x}{2\gamma^2}\right)}{\sum_{i=1}^{k} T_i \frac{1}{\gamma_i^d}\exp\left(-\frac{x}{2\gamma_i^2}\right)}. \tag{12}$$

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 high 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 jumping kernel, for example $c_d = 2.4/\sqrt{d}$. 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 jumping kernel $J_{\gamma_k \Sigma_k}$. Update the covariance matrix using the iterative procedure

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

   where $T_{total} = T_1 + \cdots + T_k$, and update the scaling using the adaptive algorithm. We also must 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.

The updated covariance Σ_{k+1} might not be positive definite. In this situation we recommend using a eigenvalue decomposition of the updated covariance, setting the minimum eigenvalue to a small positive value, and rounding up the smaller eigenvalues to this minimum value.

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. If we update the scaling jumping kernel at each step of the iteration using Newton's method,

$$\gamma_{k+1} = \gamma_k - \frac{\hat{h}'_k(\gamma_k \mid \gamma_k, \ldots, \gamma_1)}{\hat{h}''_k(\gamma_k \mid \gamma_k, \ldots, \gamma_1)},$$

the scaling parameter γ will converge fast in a neighborhood of the true maximum; otherwise bounds on parameters are required in order to implement it successfully. In our examples, we have had success updating the jumping kernel every 50 iterations of the Metropolis algorithm, until approximate convergence. At this point the MCMC algorithm is ready for its "production" run.

4 Examples

In our first three examples we use target distributions and proposals for which optimal jumping 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 Gibbs-Metropolis updating.

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) apply regarding the choice of optimal scaling. This example provides some guidelines regarding the speed of convergence, the optimal sample size, and the effectiveness of our procedure for different dimensions. In our experiments, our approach outperforms the stochastic Robbins-Monro algorithm, as implemented by Atchadé and Rosenthal (2003).

Figure 1 shows the convergence of the adaptive optimization procedure for dimensions d=1, 10, 25, 50, and 100 as well as the corresponding values of the multiple importance sampling estimator of ESJD and average acceptance probability.

Insert Figure 1 here “Optimizing ESJD”

When starting from very small values, the estimated optimal scale shows some initial high upward jumps, because the importance sampling ratio can be unbounded Convergence to the optimal scaling is achieved in 20 steps with sample size $T = 50 \times 20 = 1000$ for dimension d less then 50. For dimension $d = 100$, reliable convergence requires 30 or more steps of 50 iterations each.

In order to compare our algorithm with the stochastic Robbins-Monro algorithm, we 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 $h(\gamma) = (\iint \alpha_\gamma(x, y) J_\gamma(x, y) dx\, dy - \alpha_*)^2$, where α_* is defined as the acceptance rate corresponding to the Gaussian kernel that minimizes the first-order autocorrelation.

Insert Figure 2 here “Coerced probability method”

The convergence of the algorithm coercing the acceptance probability is faster then when maximizing ESJD, which we attribute to the fact that the acceptance probability is less variable then 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. Even when jumps are not accepted, our imnportance sampling estimate uses the information in the attempted jumps via the acceptance probabilities.

To show that our method converges also in extreme cases, we apply our method with two starting values of (0.01,50)$\times 2.4/\sqrt{d}$ for $d = 25$. We use an optimization procedure that is a combination of golden search and successive parabolic interpolation (see Brent (1973)) on the interval $[0.01, 100]$.

Insert Figure 3 here "Extreme starting points"

4.2 Correlated normal target distribution

We next illustrate adaptive scaling for a target distribution with unknown covariance matrix. We consider a two-dimensional target distribution with covariance $\Sigma = \begin{pmatrix} 100 & 9 \\ 9 & 1 \end{pmatrix}$. A choice for the covariance matrix of the initial Gaussian proposal is the inverse of the Hessian, $-\bigtriangledown^2 \log(\pi)$, computed at the maximum of π. Unfortunately, numerical optimization methods can perform very badly for high dimensions when the starting point of the algorithm is not close to the maximum. Even for such a simple distribution, starting the BFGS optimization algorithm far from the true mode might not find the global maximum, resulting in a bad initial proposal covariance matrix Σ_0. To represent this possibility, we start here with an independent proposal $\Sigma_0 = \begin{pmatrix} 25 & 0 \\ 0 & 1 \end{pmatrix}$. Figure 4 shows the performance of our algorithm; approximate convergence is achieved in 20 steps.

Insert Figure 4 here "Convergence vs. step number"

4.3 Mixture target distribution

We consider now a target distribution that is a mixture of Gaussians 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 twofold: first to illustrate that for bimodal distribution the optimal scaling is different from the Gaussian results $c_d = 2.4/\sqrt{d}$, and second to compare our method with the stochastic Robbins-Monro algorithm of Andrieu and Robert (2001, Section 7.1) where the acceptance probability was 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.

Insert Figure 5 here "ESJD vs coerced acceptance probability method"

The convergence to the "optimal" acceptance probability for the coerced probability method is attained in 1000 iterations for all starting values, an improvement over the approximately 10000 iterations required under the stochastic optimization algorithm (see Andrieu and Robert (2001), Figure 6). Maximizing ESJD yields an optimal scaling of $\gamma = 9.0$, and a comparison of the correlation structure ρ_t (the bottom two graphs of Figure 5) at the optimal scales determined by the two objective functions shows that the autocorrelation decreases much faster for the optimal scale which maximizes ESJD, thus making the ESJD a more appropriate efficiency measure.

4.4 16-dimensional nonlinear model

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

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

where $g(x,\beta) = \beta_1 + \beta_2/\left(1+(x/\beta_3)^{-\beta_4}\right)$. For each sample j, we model

$$
\begin{aligned}
&\log x_j^{init} \sim N(\log(d_j^{init}\cdot\theta_j), (\sigma^{init})^2), \text{ for the standard sample } j=0\\
&x_j^{init} = \theta_j, \text{ for the unknown samples } j = 1,\ldots,10.
\end{aligned}
$$

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 using data 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 dilutions 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 BFGS to find the maximum likelihood 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, 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 30 steps with 50 iterations per step.

Insert Figure 6 here "16-dimensional nonlinear model"

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

4.5 Metropolis sampling within Gibbs

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

$$y_j \sim \mathrm{N}(\theta_j, \sigma_j^2),\ \sigma_j \text{ known, for j=1,...,8}$$

$$\theta_j|\nu, \mu, \tau \sim t_\nu(\mu, \tau^2),\ \text{for } j = 1, \dots, 8.$$

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 updating $1/\nu$. 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 the coercing the acceptance probability to 44%.

Insert Figure 7 here "Gibbs within Metropolis"

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 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 jumping kernels.

Coercing the acceptance probability has slightly faster convergence then maximizing the ESJD but not necessarily to an optimal value as we have seen in Figure 5. The proof of ergodicity of the adaptive chain which adapts both scaling and covariance remains an open theoretical question as does the relationship between ESJD, the eigenvalue structure of the Metropolis kernel, and convergence speed. 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.

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 jumping 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 that involve a translation parameter as well as a scale for the jumping kernel and can achieve higher efficiencies then symmetric Metropolis algorithms (see Roberts and Rosenthal (2001)).

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.

Besag, J., and Green, P. J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society B* **55**, 25-37.

Brent, R. (1973). *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, N.J.: Prentice-Hall.

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

Gelfand, A. E., and Sahu, S. K. (1994). On Markov chain Monte Carlo acceleration. *Journal of Computa-*

tional and Graphical Statistics **3**, 261-276.

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*, to appear.

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

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo inference. *Econometrics* **38**, 73-90.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society B* **56**, 261-274.

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

Geyer, C. J., and Thompson, E. A. (1992). Constrained maximum Monte Carlo maximum likelihood for dependent data (with discussion). *Journal of the Royal Statistical Society B* **54**, 657-699.

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

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

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. *Review of Economic Studies* **65**, 361-393.

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

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*. New York: Springer-Verlag.

Mira, A. (2001). Ordering and improving the performance of Monte Carlo Markov chains. *Statistical Science* **16**, 340-350.

Neal, R. M. (2003). Slice sampling (with discussion). *Annals of Statistics* **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*. New-York: Springer.

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

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

R Project (2000). The R project for statistical computing. `www.r-project.org`.

Tierney, L., and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in 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.

Appendix

Proof of Proposition 2

The chain $\{(\theta_t, \theta_t^*)\}$ is a positive Markov chain with invariant probability $\pi(dx)J_\gamma(x, dy)$. Given that θ_t is irreducible, it satisfies the conditions of Robert and Casella (1998, Theorem 6.2.5 i), and consequently,

$$h_T(\gamma|\gamma_0) = \frac{1}{T}\sum_{t=1}^{T} H(\gamma, \theta_t, \theta_t^*) w_{\gamma|\gamma_0}(\theta_t, \theta_t^*) \rightarrow \int\int H(\gamma, x, y)\pi(x)J_\gamma(x, y)dx\, dy, \; a.s., \forall\gamma \in \Gamma. \tag{13}$$

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 (13) holds $a.s.$ for all γ in a countable dense set in Γ. By the weak convergence of measures,

$$\inf_{\phi\in B} \frac{1}{T}\sum_{t=1}^{T} H(\phi, \theta_t, \theta_t^*) \cdot w_{\phi|\gamma_0}(\theta_t, \theta_t^*) \rightarrow \int\int \inf_{\phi\in B} H(\phi, x, y)\pi(x)J_\phi(x, y)dx\, dy, \; a.s. \tag{14}$$

holds, for all γ in a countable dense set in Γ. Convergence on compacts is a consequence of epiconvergence and hypoconvergence (see, for example, Geyer (1994)). In order to prove epiconvergence we need to show that

$$h(\gamma) \leq \sup_{B\in N(\gamma)} \liminf_{t\rightarrow\infty} \inf_{\phi\in B} \{h_t(\phi|\gamma_0)\} \tag{15}$$

$$h(\gamma) \geq \sup_{B\in N(\gamma)} \limsup_{t\rightarrow\infty} \inf_{\phi\in B} \{h_t(\phi|\gamma_0)\}, \tag{16}$$

where $N(\gamma)$ are the neighborhoods of γ. By topological properties of $\mathbf{R}$, there exist a countable base of open neighborhoods V_n. By the continuity of $\gamma \rightarrow H(\gamma,)J_\gamma$ we can replace the infima by infima over countable sets (e.g., rational numbers). Now construct a sequence $\Gamma_c = (x_n)_n \in \Gamma$ dense in $\mathbf{R}$ such that, x_n satisfies

$$h(x_n) \leq \inf_{x\in V_n} h(x) - \frac{1}{n}.$$

¿From (13) we have $\lim_{t\rightarrow\infty} h_t(\gamma|\gamma_0) \rightarrow h(\gamma)$, for all $\gamma \in V_n \cap \Gamma_c$. Consequently, for all $\gamma \in V_n \cap \Gamma_c$, and $B \in N(\gamma)$

$$h(\gamma) = \lim_{t\rightarrow\infty} h_t(\gamma|\gamma_0) \geq \limsup_{t\rightarrow\infty} \inf_{\phi\in B} h_t(\phi|\gamma_0)$$

which implies,

$$\inf_{\phi\in B\cap\Gamma_c} \{h(\phi)\} \geq \limsup_{t\rightarrow\infty} \inf_{\phi\in B} h_t(\phi|\gamma),$$

for any B neighborhood of γ. Take a decreasing collection B_n of neighborhoods of γ such that $\cap B_n = \{\gamma\}$, and we have

$$\limsup_{n\to\infty} \inf_{\phi\in B_n\cap\Gamma_c} \{h(\phi)\} \geq \limsup_{n\to\infty} \limsup_{t\to\infty} \inf_{\phi\in B_n} h_t(\phi|\gamma).$$

Now (16) reduces to proving that the left-hand side is less then $h(\gamma)$, and follows if h is continuous. The continuity of $H(\gamma,)\cdot J_\gamma$, assumption (10) and the dominated convergence theorem

$$h(\gamma) = \iint \left[\lim_{k\to\infty} H(\gamma_k,x,y)J_{\gamma_k}(x,dy)\right]\pi(x)dx = \lim_{k\to\infty}\iint \left[H(\gamma_k,x,y)J_{\gamma_k}(x,dy)\right]\pi(x)dx = \lim_{k\to\infty} h(\gamma_k)$$

yield that h is continuous, concluding the proof of (16).

In order to prove (15) we apply the dominated convergence theorem to get,

$$\sup_{B_k} \mathbf{E}\left[\inf_{\phi\in B_k} \frac{H(\phi,\theta_t,\theta_t^*)}{H(\gamma_0,\theta_t,\theta_t^*)}\right] \to \mathbf{E}\left[\sup_{B_k} \inf_{\phi\in B_k\cap\Gamma_c} \frac{H(\phi,\theta_t,\theta_t^*)}{H(\gamma_0,\theta_t,\theta_t^*)}\right] = h(\gamma).$$

The hypo-continuity follows from similar arguments.

Proof of Proposition 3

We need to prove that the assumptions of Proposition 2 are verified. Clearly the continuity assumption is satisfied, and we check now (10). For simplicity, we omit the subscript and use the notation $\|\| = \|\|_{\Sigma^{-1}}$. Fix $\gamma > 0$ and $\epsilon > 0$ small enough,

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

Taking into account that

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

with $K > 0$, the first integral becomes

$$\begin{aligned}
&\int \left(\int_{d(\gamma-\epsilon)^2 < \|y-x\|^2 < d(\gamma+\epsilon)^2} \sup_{\phi \in (\gamma-\epsilon, \gamma+\epsilon)} \left(J_\phi(\|y-x\|) \right) \|y-x\|^2 dy \right) \pi(x) dx \\
&\quad \leq K \int \left(\int_{0 < \|y-x\|^2 < d(\gamma+\epsilon)^2} \frac{1}{d(\gamma-\epsilon)^2} dy \right) \pi(x) dx \\
&\quad = K \int_{0 < \|z\|^2 < d(\gamma+\epsilon)^2} \frac{1}{d(\gamma-\epsilon)^2} dz < \infty,
\end{aligned} \tag{17}$$

and the second integral can be bounded as follows:

$$\begin{aligned}
&\int \left(\int_{\|y-x\|^2 \notin (d(\gamma-\epsilon)^2,\ d(\gamma+\epsilon)^2)} \sup_{\phi \in (\gamma-\epsilon, \gamma+\epsilon)} J_\phi(\|y-x\|) \|y-x\|^2 dy \right) \pi(x) dx \\
&\qquad = \int \left(\int_{\|y-x\|^2 \leq d(\gamma-\epsilon)^2} \sup_{\phi \in (\gamma-\epsilon, \gamma+\epsilon)} J_\phi(\|y-x\|) \|y-x\|^2 dy \right) \pi(x) dx \\
&\qquad \quad + \int \left(\int_{\|y-x\|^2 \geq d(\gamma+\epsilon)^2} \sup_{\phi \in (\gamma-\epsilon, \gamma+\epsilon)} J_\phi(\|y-x\|) \|y-x\|^2 dy \right) \pi(x) dx \\
&\qquad = \left(\int_{\|y-x\|^2 \leq d(\gamma-\epsilon)^2} J_{\gamma-\epsilon}(\|y-x\|) \|y-x\|^2 dy \right) \pi(x) dx \\
&\qquad \quad + \int \left(\int_{\|y-x\|^2 \geq d(\gamma+\epsilon)^2} J_{\gamma+\epsilon}(\|y-x\|) \|y-x\|^2 dy \right) \pi(x) dx < \infty.
\end{aligned} \tag{18}$$

Combining (17) and (18) proves (10).

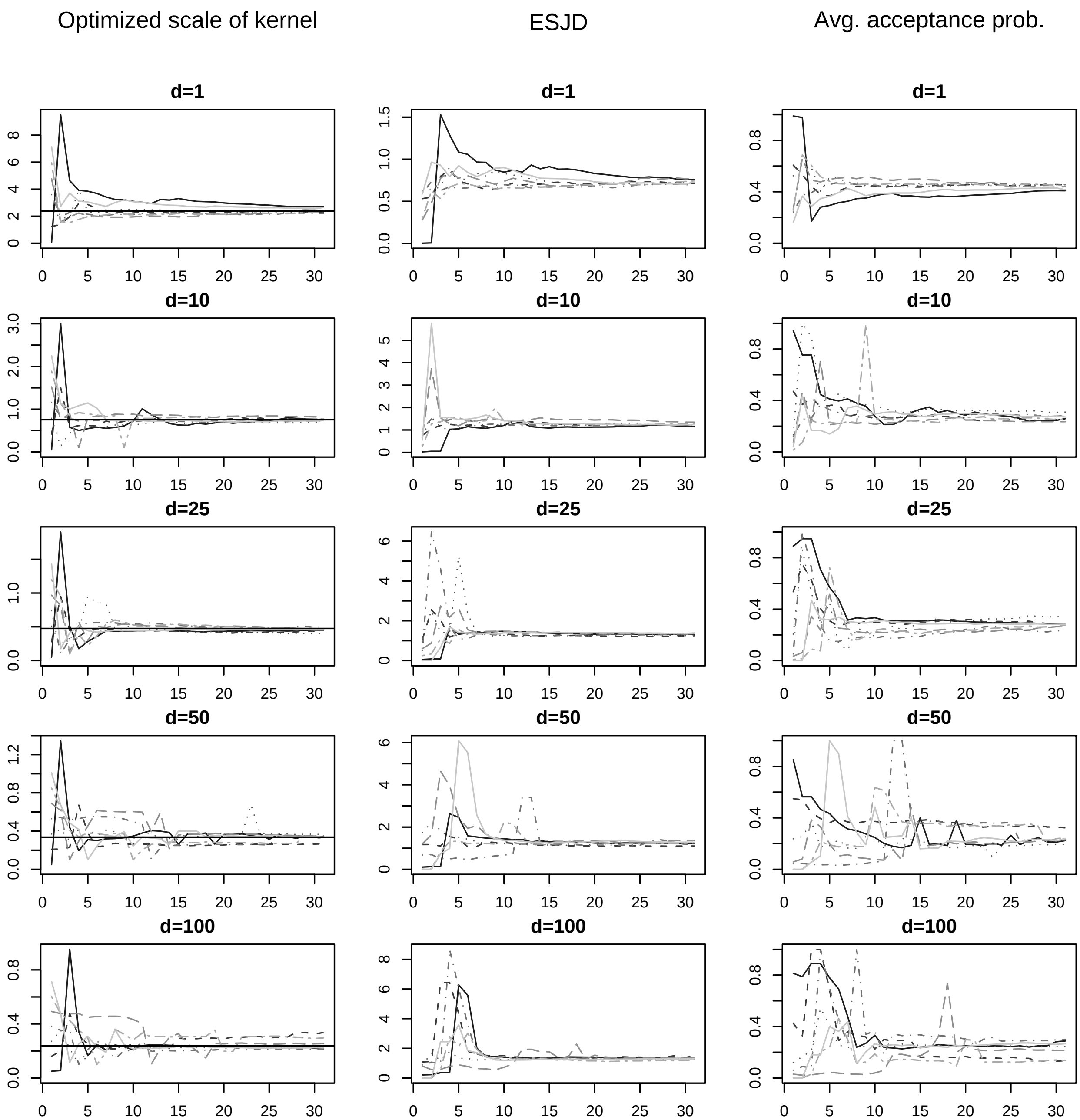

Figure 1: Convergence to the optimal value (solid horizontal line) of the adaptive optimization procedure, given seven equally spaced starting points points in the interval $[0, 3 * 2.4/\sqrt{d}]$, 50 iterations per step, for dimensions $d = 1$, 10, 25, 50, and 100 for the random walk Metropolis algorithm with multivariate normal target distribution. The second and third column of figures show the multiple importance sampling estimator of ESJD and average acceptance probability, respectively.

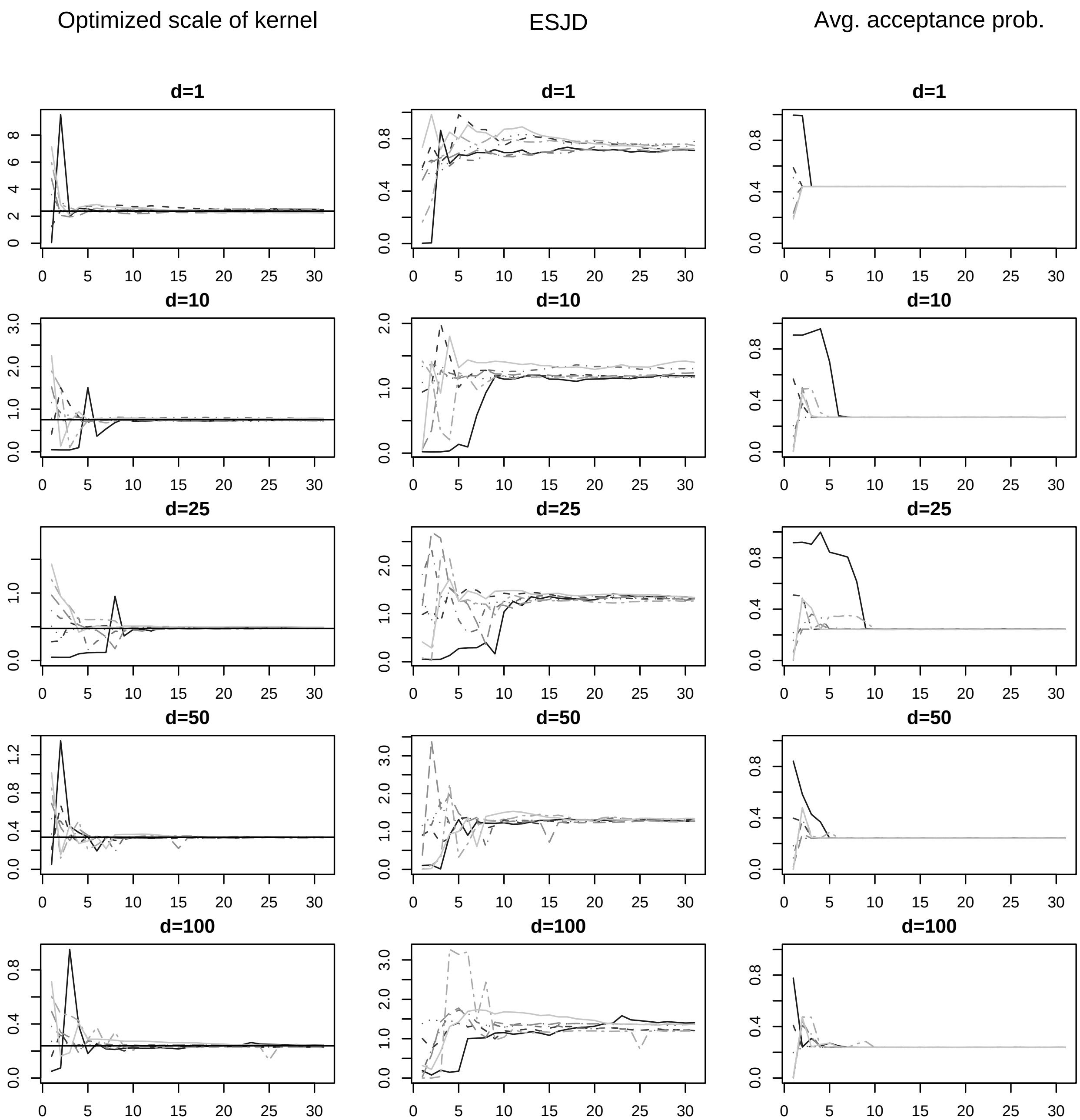

Figure 2: Convergence of the adaptive optimization procedure using as objective the coerced average acceptance probability (to the optimal acceptance value from Figure 1). The second and third column show the multiple importance sampling estimator of the ESJD and average acceptance probability, respectively. Convergence of the optimal scale is faster then optimizing ESJD, although not necessarily to the most efficient jumping kernel (see Figure 5).

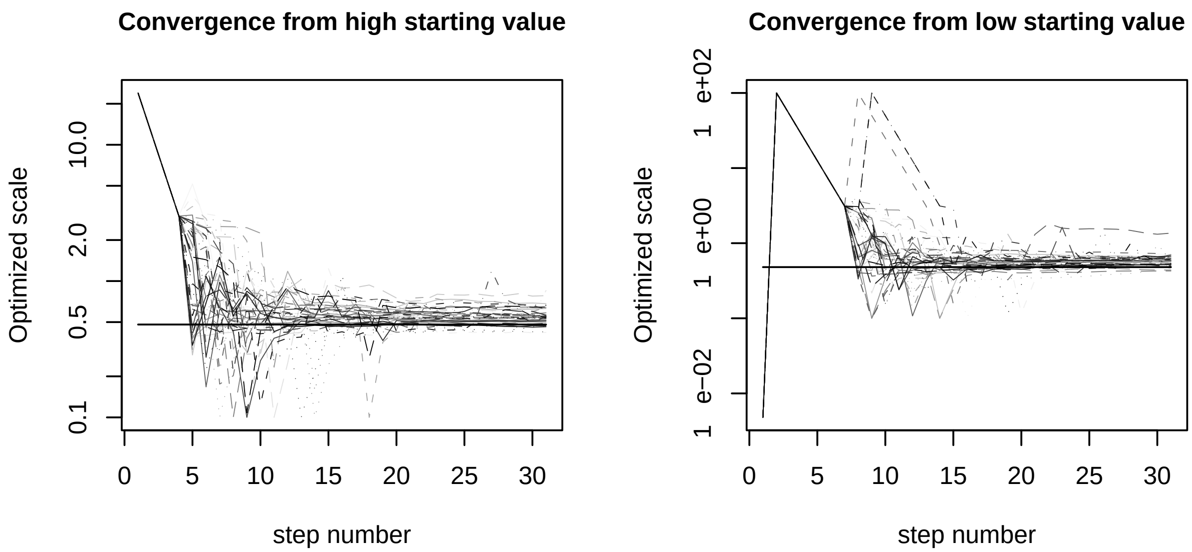

Figure 3: Convergence of the adaptive optimization procedure with extreme starting points of 0.01 and 50 times the optimum, for dimension $d = 25$ with multivariate normal target distribution, for 50 independent paths with 50 iterations per steps. The estimated optimal scales are plotted on the logarithmic scale.

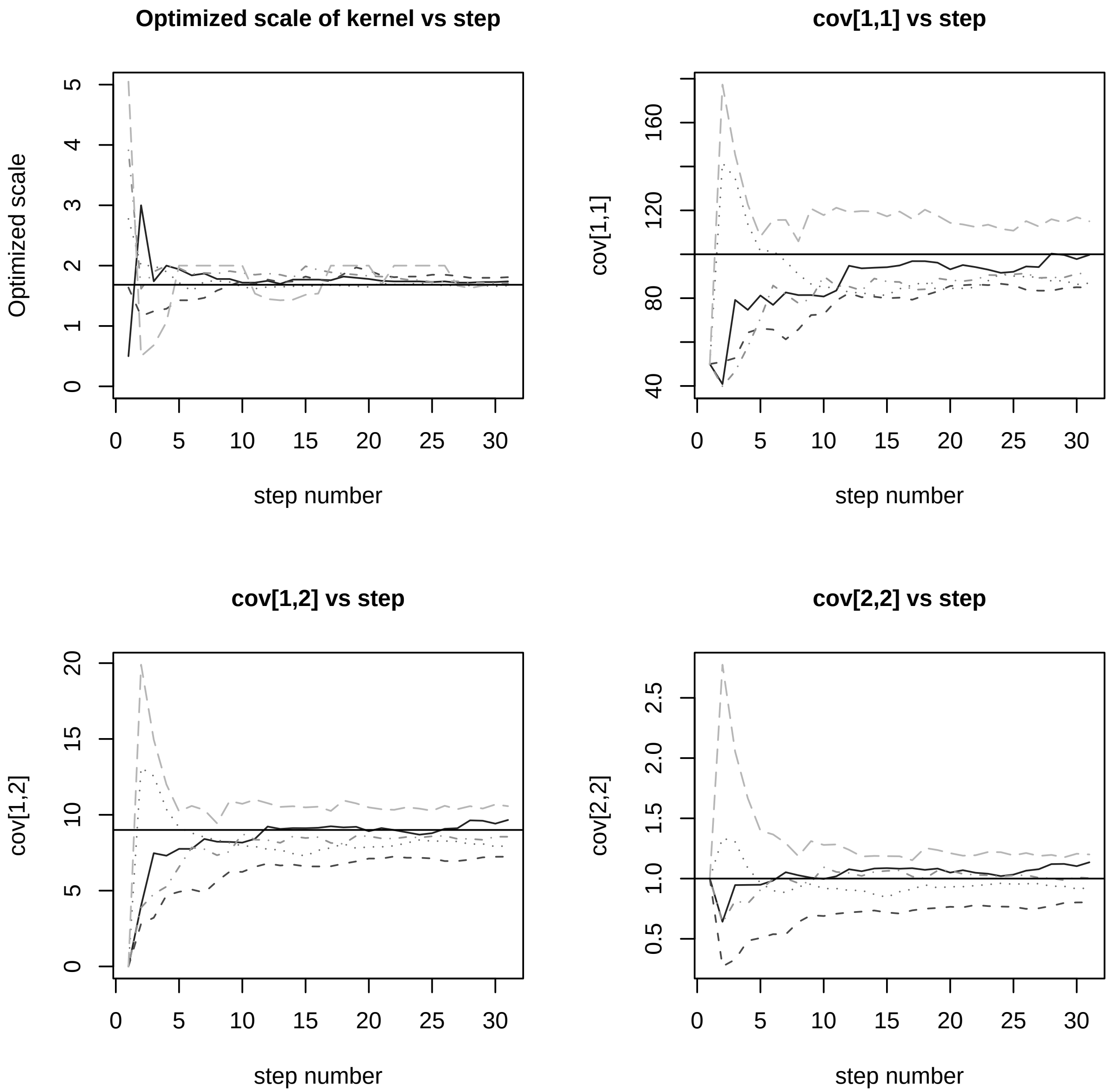

Figure 4: Convergence of the adaptive optimization procedure that maximizes the ESJD by scaling and updating the covariance matrix, starting with independent proposal density with 50 iterations per step. Convergence of the sample covariance matrix is attained in 20 steps and convergence to optimal scaling in 30 steps.

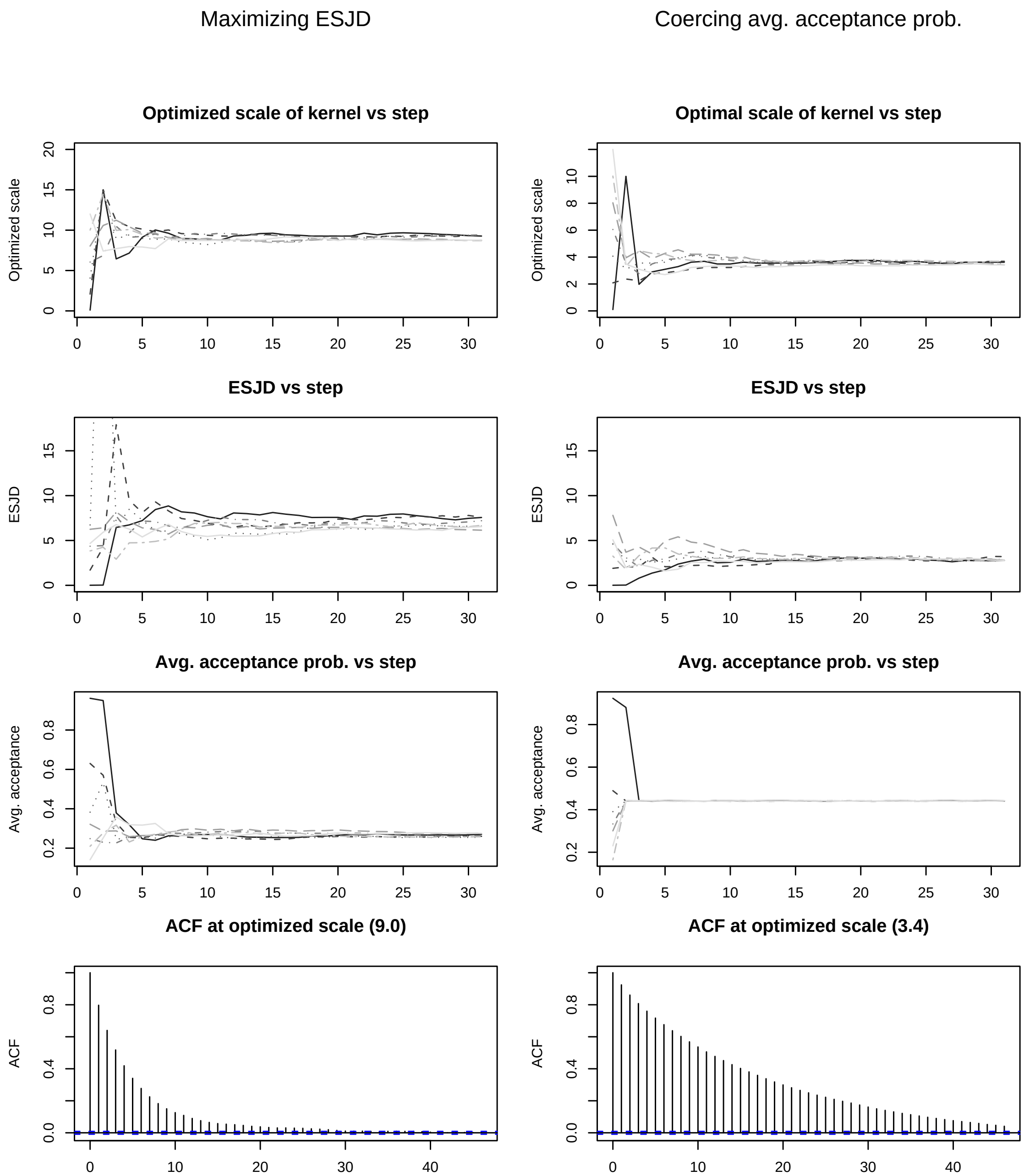

Figure 5: Comparison of two objective functions for the 2-component mixture target of Andrieu and Robert (2001) using our adaptive optimization algorithm: maximizing ESJD (left column of plots), and coercing the acceptance probability to 44% (right column of plots), with 50 iterations per step. The coerced acceptance probability method converges slightly faster but to a less efficient kernel (see ACF plot).

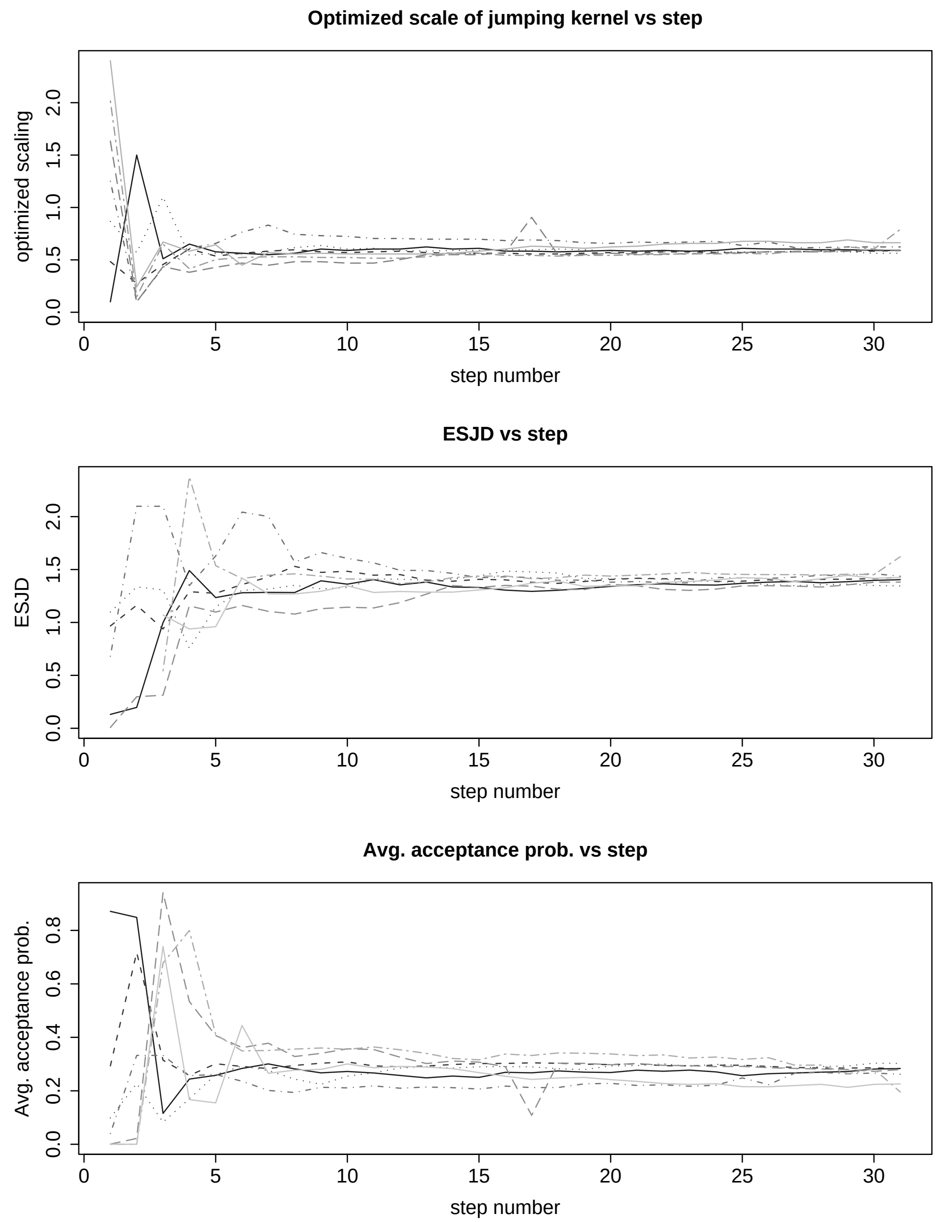

Figure 6: 16-dimensional nonlinear model for a serial dilution experiment from Gelman, Chew, and Shnaidman (2004); convergence to optimal scaling, for seven equally spaced starting values in $[0, 2.4]$ with 50 iterations per step and covariance matrix determined by initial optimization.

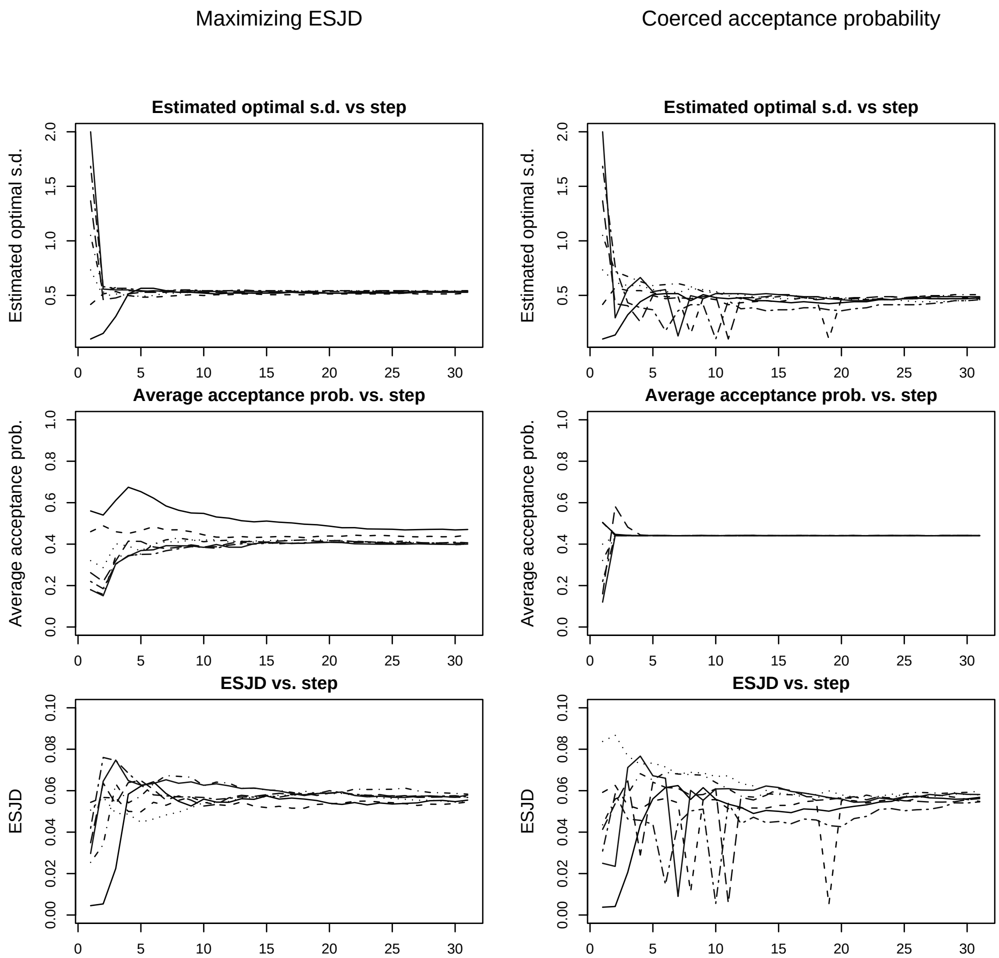

Figure 7: Gibbs sampling with a Metropolis step for the inverse-degrees-of-freedom parameter in the hierarchical t model for the eight schools example of Gelman et al. (2003); convergence of optimal scaling given starting values in $[0, 2]$ for two objective functions: maximizing ESJD (left column of plots) and coercing average acceptance probability to 44% (right column of plots).