Author(s): Vihola, Matti

Title: Ergonomic and Reliable Bayesian Inference with Adaptive Markov Chain Monte Carlo

Year: 2020

Version: Published version

Please cite the original version:

Vihola, M. (2020). Ergonomic and Reliable Bayesian Inference with Adaptive Markov Chain Monte Carlo. In N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri, & J. L. Teugels (Eds.), Wiley StatsRef : Statistics Reference Online (pp. 1-12). John Wiley & Sons. https://doi.org/10.1002/9781118445112.stat08286

This is the author version of a forthcoming book section:
M. Vihola. Ergonomic and reliable Bayesian inference with adaptive Markov chain Monte Carlo. In W. W. Piegrorsch, R. Levine, H. H. Zhang, and T. C. M. Lee, editors, *Handbook of Computational Statistics and Data Science*, Wiley, to appear.

Ergonomic and reliable Bayesian inference with adaptive Markov chain Monte Carlo

Matti Vihola
University of Jyväskylä, Finland

Abstract
Adaptive Markov chain Monte Carlo (MCMC) methods provide an ergonomic way to perform Bayesian inference, imposing mild modeling constraints and requiring little user specification. The aim of this section is to provide a practical introduction to selected set of adaptive MCMC methods, and to suggest guidelines for choosing appropriate methods for certain classes of models. We consider simple unimodal targets with random-walk based methods, multimodal target distributions with parallel tempering, and Bayesian hidden Markov models using particle MCMC. The section is complemented by an easy-to-use open-source implementation of the presented methods in Julia, with examples.

1. Introduction

The Markov chain Monte Carlo (MCMC) revolution in the 1990s and the following widespread popularity of the Bayesian methods was largely fuelled by the introduction of the BUGS software[27]. With BUGS, the user could focus on the statistically important part, and let the software take care of the MCMC inference automatically. Unfortunately, the Gibbs sampling approach used by (variants of) BUGS has certain limitations, such as imposing some modeling constraints due to conjugacy and suffering poor mixing with high correlations.

This section provides a self-contained review of selected simple, robust and general-purpose adaptive Markov chain Monte Carlo methods, which can deliver (nearly) automatic inference like BUGS, but can overcome some of its limitations. We focus on methods based on random-walk Metropolis (RWM)[28] and parallel tempering (PT; also known as replica exchange)[41]. We also discuss guidelines how the methods can be used with particle MCMC[1], in order to do inference for a wide class of Bayesian hidden Markov models.

Instead of rigorous theory, the aim is to give an intuitive understanding why the methods work, what methods are suitable for certain problem classes, and how they can be combined with some other methods. The methods are explained algorithmically, and guidelines are given for parameter values. For more in-depth insight to the theory and methods of adaptive MCMC, the reader is advised to consult the review[4] and references therein, and the articles about rigorous theoretical foundations[19,6,2,34]. The section is complemented by open-source Julia[9] packages[1,2] which implement the methods and illustrate them on examples.

1. https://github.com/mvihola/AdaptiveMCMC.jl
2. https://github.com/mvihola/AdaptiveParticleMCMC.jl

2. Random-walk Metropolis algorithm

Suppose for now that π is a probability density of interest on $\mathbb{R}^d$. Let ℓ stand for the unnormalized log-target, that is, $\ell(x) = \log \pi(x) + c$, where $c \in \mathbb{R}$ is a constant whose value need not be known. In the case of Bayesian inference, ℓ will typically be the sum of the log-likelihood an the log-prior density. Algorithm 1 presents pseudo-code for a random-walk Metropolis algorithm[28] targetting π, with initial state $x_0 \in \mathbb{R}^d$, number of iterations n, a symmetric proposal distribution q on $\mathbb{R}^d$, which we shall take as the standard normal, and a (non-singular) proposal shape $S \in \mathbb{R}^{d\times d}$.

Algorithm 1 $X_{1:n} \leftarrow \mathrm{RWM}(\ell,\, x_0,\, n,\, S)$

Set $X_0 \leftarrow x_0$ and $P_0 \leftarrow \ell(x_0)$.
for $k = 1, \ldots, n$ **do**:
  $(X_k, P_k; \alpha_k, Z_k) \leftarrow \mathrm{RWMStep}(X_{k-1}, P_{k-1}, \ell, S)$

function RWMStep(X, P, ℓ, S):
  Draw $Z \sim q$ and set $X' \leftarrow X + SZ$.
  Calculate $P' \leftarrow \ell(X')$, let $\alpha \leftarrow \min\{1, \exp(P' - P)\}$ and draw $U \sim U(0, 1)$.
  if $U \leq A$ **then return** $(X', P'; \alpha, Z)$; **else return** $(X, P; \alpha, Z)$.

The samples $X_{b:n} = (X_b, \ldots, X_n)$ produced by Algorithm 1, for some sufficiently large 'burn-in' length $1 \leq b \leq n$, say $b = 0.1n$, are approximately distributed as π. The samples are not independent, but if the chain is well-behaved and n sufficiently large, they provide a reliable empirical approximation of π.

It is sufficient to choose any initial state x_0 such that $\ell(x_0) > -\infty$, but it is generally advisable to choose x_0 near the maximum of ℓ. In order to make the method efficient, the proposal increment shape S needs to be tuned based on the properties of the target π. There are two general 'rules of thumb' for choosing S, originating from several theoretical results, starting from the seminal work[33]:

(R1) The proposal covariance $SS^T \approx 2.38^2 d^{-1} \Sigma_\pi$, where $\Sigma_\pi = \mathrm{cov}(\pi)$.

(R2) Choose S such that $\mathrm{avg}(\alpha_1, \ldots, \alpha_n) \approx 0.234$ (or perhaps 0.44 if $d = 1$).

The random-walk adaptations discussed below implement automatic adjustment of S based on these rules.

3. Adaptation of random walk Metropolis

All of the adaptive RWMs that we discuss may be written in a common form as summarized in Algorithm 2, where we use the RWMStep of Algorithm 1. Table 1 summarizes the ingredients of the four commonly used instances of Algorithm 2, which are discussed below.

Algorithm 2 $X_{1:n} \leftarrow \mathrm{ARWM}(\ell,\, x_0,\, n)$

Initialize ξ_0, set $X_0 \leftarrow x_0$ and $P_0 \leftarrow \ell(x_0)$.
for $k = 1, \ldots, n$ **do**:
  $(X_k, P_k; \alpha_k, Z_k) \leftarrow \mathrm{RWMStep}\big(X_{k-1}, P_{k-1}, \ell, \mathrm{Shape}(\xi_k)\big)$
  $\xi_k \leftarrow \mathrm{Adapt}(k, \xi_{k-1}, X_k, Z_k, \alpha_k)$.

3.1. Adaptive Metropolis (AM)

The seminal Adaptive Metropolis algorithm[19] is a direct implementation of the rule (R1). The adaptation defines $\mathrm{Shape}(\xi_k) = \mathrm{Chol}(2.38^2 d^{-1}\Sigma_k)$, where $\mathrm{Chol}(S)$ stands for the lower-triangular Cholesky factor L such that $LL^T = S$, and where where Σ_k is an estimator of $\mathrm{cov}(\pi)$. In the original work[19], the regularized empirical covariance $\Sigma_k = \mathrm{Cov}(X_1, \ldots, X_k) + \epsilon I_d$ was used, where $\epsilon > 0$ was a user-defined parameter.

The follow-up work[2] suggested a slightly modified AM adaptation rule, where Σ_k is a recursively defined covariance estimator defined as follows:

$$\begin{aligned} \mu_k &= \mu_{k-1} + \gamma_k(X_k - \mu_{k-1}) \\ \Sigma_k &= \Sigma_{k-1} + \gamma_k\big[(X_k - \mu_{k-1})(X_k - \mu_{k-1})^T - \Sigma_{k-1}\big], \end{aligned} \tag{1}$$

where γ_k is a step size sequence decaying to zero, typically $\gamma_k = (k+1)^{-\beta}$, where $\beta \in (1/2, 1]$, and initial values may be set as $\mu_0 = x_0$ and $\Sigma_0 = I_d$, the identity matrix on $\mathbb{R}^d$.

We suggest to use (1) with the common choice $\gamma_k = (k+1)^{-1}$, which behaves asymptotically similar to the original rule[19], with $\epsilon = 0$. The update (1) is appealing because it avoids the need to choose the regularization factor ϵ, and allows for calculation of $C_k = \mathrm{Chol}(\Sigma_k)$ using rank-1 Cholesky updates $C_{k-1} \to C_k$[11], which cost $O(d^2)$ in contrast with $O(d^3)$ cost of direct calculation of the Cholesky factor. We define the state of adaptation $\xi_k = (\mu_k, C_k)$.

In higher dimensions, the AM adaptation may sometimes suffer from poor initial behavior[46], which may be resolved by adding a fixed (non-adaptive) component in the proposal distribution[46,7], or using a regularization factor $\epsilon > 0$ as in the original work. Stability may also be improved by adding a delayed rejection stage to the algorithm[18], or using a modified update with X_{k-1} and Y_k weighted by rejection and acceptance probabilities, respectively, which corresponds to one-step Rao-Blackwellization[4].

3.2. Adaptive Scaling Metropolis (ASM)

Automatic selection of the parameter S of the RWM based on rule (R2) has been suggested at regeneration times[15] and attempting to directly optimize a loss function[3]. We consider the following simpler adaptation rule[5,2], which is called here adaptive scaling Metropolis: set $\mathrm{Shape}(\xi_k) = e^{\eta_k}$, where $\xi_k = \eta_k$ is adapted with

$$\eta_k = \eta_{k-1} + \gamma_k(\alpha_k - \alpha_*), \tag{2}$$

where $\alpha_* = 0.234$ (or 0.44 if $d = 1$), and with (recommended) step size $\gamma_k = k^{-2/3}$. This adaptation is simpler than the AM adaptation, and even more robust, in the sense that no specific initialization strategies or stabilizing mechanisms are necessary[47]. But because

Table 1: Summary of ingredients of Algorithm 2 for the four adaptive MCMC methods. I_d stands for the identity matrix in $\mathbb{R}^d$, and $\mathbb{L}_d \subset \mathbb{R}^{d\times d}$ is the set of lower-triangular matrices.

Method	Initialization ξ_0	State ξ_k	Domain of ξ_k	Adapt$(\cdot)$	Shape(ξ_k)
AM	(x_0, I_d)	(μ_k, C_k)	$\mathbb{R}^d \times \mathbb{L}_d$	(1)	$2.38 d^{-1/2} C_k$
ASM	1	η_k	$\mathbb{R}$	(2)	e^{η_k}
ASM+AM	$(x_0, I_d, \log(2.38 d^{-1/2}))$	(μ_k, C_k, η_k)	$\mathbb{R}^d \times \mathbb{L}_d \times \mathbb{R}$	(1) & (2)	$e^{\eta_k} C_k$
RAM	I_d	S_k	$\mathbb{L}_d$	(3)	S_k

ASM is essentially univariate, it cannot (automatically) capture correlation structures, which may lead to inefficient sampling.

It is quite natural to also use covariance information in the ASM. If no prior information about cov_π is available, we may directly use the AM adaptation together with ASM[6,5,4], by setting $\mathrm{Shape}(\xi_k) = e^{\eta_k} C_k$, where $\xi_k = (\mu_k, C_k, \eta_k)$ and (μ_k, C_k) is adapted with AM (1). In this approach, hereafter ASM+AM, it is recommended that a common step size, for instance $\gamma_k = (k+1)^{-2/3}$, is used for both the AM and ASM adaptations.

3.3. Robust Adaptive Metropolis (RAM)

There is an alternative to the combination of AM and ASM, which implements the rule (R2) using directional information. The robust adaptive Metropolis (RAM)[48] uses the following direct update on $\mathrm{Shape}(\xi_k) = S_k$:

$$S_k S_k^T = S_{k-1} S_{k-1}^T + \gamma_k(\alpha_k - \alpha_*) V_k V_k^T, \qquad \text{where} \qquad V_k = S_{k-1} Z_k / \|Z_k\|, \tag{3}$$

which may also be implemented as $O(d^2)$ cost rank-1 Cholesky update/downdate[11].

In the univariate case, the RAM update shares similar behavior with the ASM (2), in the sense that then $S_k^2 \approx e^{\eta_k}$. This is because

$$2 \log S_k = 2 \log S_{k-1} + \log\big(1 + \gamma_k(\alpha_k - \alpha_*)\big) \approx 2 \log S_{k-1} + \gamma_k(\alpha_k - \alpha_*),$$

for small γ_k. This suggests that RAM can be seen as a multivariate extension of the ASM adaptation. The recommended step size of RAM is $\min\{1, d \cdot k^{-2/3}\}$, where the dimension d inflates the step size because of the directional adaptation[48].

Similar to the ASM, the RAM adaptation has been found stable empirically, typically not requiring specific initialization strategies. However, the ASM+AM adaptation has been suggested to be used initially, before starting the RAM adaptation[39].

3.4. Rationale behind the adaptations

When looking at the adaptation formulae (1)–(3), it is evident that they all are similar: the previous value of the state is updated by an increment weighted by a decreasing positive step size γ_k. The fact that the changes in the adaptation get smaller and smaller is key point for the validity of the methods, and is called 'diminishing' or 'vanishing' adaptation[34,2]. Roughly speaking this combined with suitable uniform-in-S mixing assumption of the RWM ensure the validity of the algorithms.

The specific forms of adaptation considered here can all be viewed as stochastic gradient type methods[32,8] as pointed out in[3,2]. Their limiting behavior is intuitively characterized by replacing the increments with their stationary expectations, regarding ξ_{k-1} as constant. For instance, such an 'averaged' version of the AM update (1) would be

$$\begin{aligned} \mu_k &= \mu_{k-1} + \gamma_k(\mu_\pi - \mu_{k-1}) \\ \Sigma_k &= \Sigma_{k-1} + \gamma_k\big[\Sigma_\pi - \Sigma_{k-1} - \big(\mu_\pi - \mu_{k-1}\big)\big(\mu_\pi - \mu_{k-1}\big)^T\big], \end{aligned} \tag{4}$$

where μ_π is the mean of π. If the averaged update has a limit, then the adaptation tends to the same limit, under technical assumptions[2]; see also[4] for further intuitive discussion about the behavior of this type of adaptation.

It is not hard to see that (4) has a unique fixed point (μ_π, Σ_π), so AM adaptation $C_k \to \mathrm{Chol}(\Sigma_\pi)$ under general conditions. Empirically, the convergence appears to happen

always (as long as Σ_π is finite). Similarly, in case of the ASM, it is relatively easy to see[45] that the mean acceptance rate $\mathbb{E}[\alpha_k] \to 0$ as the proposal increments get smaller $\eta_{k-1} \to -\infty$, and vice versa, $\mathbb{E}[\alpha_k] \to 1$ as $\eta_{k-1} \to \infty$, suggesting that a limit always exists, but the limit might not be unique[21]. In case π is elliptically symmetric, the limit point of RAM coincides with the shape of π, up to a constant[48], as does the ASM+AM.

3.5. Summary and discussion on the methods

The adaptive RWM algorithms are simple and generally well-behaved when the corresponding RWM algorithms, with fixed (non-singular) proposal shape S, are. This requires essentially the following:

- Moderate dimension d.
- Essentially unimodal target π, that is, π does not have well-separated nodes.
- Target π has bounded support, or sufficiently regular tails that are fast decaying (super-exponentially, such as Gaussian[22]).

The tail decay rate may be enforced by a suitably chosen prior, for instance a Gaussian. There are some theoretical results about the stability of the algorithms under further technical conditions[37,47,46]. If the algorithms are modified to include auxiliary stabilizing mechanisms, typically enforcing the values of ξ_k to a compact set, they may be guaranteed to be valid even more generally[34,2,5].

The recommended step sizes γ_k differ between the algorithms, due to their different characteristics. The step sizes must ensure that the adaptations remain 'effective', in the sense that $\sum_k \gamma_k = \infty$. If this condition was not met, the algorithms could converge prematurely to a spurious limit. The limiting behavior of the methods may be guaranteed to satisfy a central limit theorem if $\sum_k \gamma_k^2 < \infty$[2]. If we focus on sequences with polynomially decaying tails $O(n^{-\beta})$, then the above are satisfied with $\beta \in (1/2, 1]$. As commented earlier, the given step size for the AM makes the algorithm behave similarly in the limit to the original algorithm, where Σ_k were sample covariances. However, with bounded increments, such as with the ASM, the choice $\gamma_k = O(k^{-1})$ would lead to η_k that can deviate from η_0 at most of order $\log k$, rendering the adaptation ineffective. With ASM+AM, there is potential interaction between the covariance and scale adaptations, and using different step sizes might amplify this. Because RAM is similar to ASM, the suggested step size decay rate is similar, but because of directional adaptation, the step size is inflated with dimension.

In a univariate case, ASM is the recommended method because of its simplicity. In a general multivariate case, using AM, ASM+AM or RAM is recommended, because these methods can adapt to different scaling of variables and correlations. In simple scenarios, they work equally well, but in some cases, differences may arise[48]. All of the adaptive RWM methods have good theoretical backing, but the results are not complete. If the user is in doubt, adaptation may also be stopped (typically after burn-in), to ensure theoretical validity with minimal conditions (irreducibility).

4. Multimodal targets with parallel tempering

RWM is based on small increments of X_k, which are accepted or rejected individually. This makes RWM behave poorly with multimodal distributions, where reaching one mode from another would require several steps that are each accepted with small probability. The higher the dimension, the more easily this problem arises, because the steps made by the RWM need to be smaller in higher dimension; of order $O(d^{-1/2})$[33].

If further information about the π, such as location of modes, is available, tailored transitions may be designed. We focus on the case where little is known about π a priori. Then, a general 'tempering' procedure may be applied, where the target density $\pi(x)$ is modified to one proportional to $\pi^{\beta}(x)$, where $\beta \in (0,1)$ is an 'inverse temperature' parameter; equivalently, the unnormalized log-density of the modified target is $\beta\ell(x)$. The lower the value of β, the more π is 'flattened' by making the modes less pronounced and the unlikely states more likely.

The parallel tempering (PT), or replica exchange algorithm[41] uses a number $L \geq 2$ of levels, with inverse temperatures $1 = \beta^{(1)} > \beta^{(2)} > \cdots > \beta^{(L)} > 0$, and corresponding unnormalized log-targets $\tilde{\ell}_{\beta^{(i)}}(x) := \beta^{(i)}\ell(x)$. The algorithm updates a joint state $X_{k-1}^{(1:L)} \to X_k^{(1:L)}$ in two stages. The first step consists of independent updates $X_{k-1}^{(1)} \to X_k^{(1)}, \ldots, X_{k-1}^{(L)} \to X_k^{(L)}$ with MCMCs targetting $\tilde{\ell}_{\beta^{(1)}}, \ldots, \tilde{\ell}_{\beta^{(L)}}$, respectively. The second step involves an attempt to swap the states of two random adjacent levels, $X_k^{(I)} \leftrightarrow X_k^{(I-1)}$, where $I \sim U\{2, \ldots, L\}$, which is accepted with probability

$$\min\left\{1, \frac{\pi^{\beta^{(I)}}(X^{(I-1)})\pi^{\beta^{(I-1)}}(X^{(I)})}{\pi^{\beta^{(I-1)}}(X^{(I-1)})\pi^{\beta^{(I)}}(X^{(I)})}\right\},$$

which ensures that $X_b^{(1)}, \ldots, X_n^{(1)}$ approximates the target distribution of interest π.

An adaptive version of this algorithm, the adaptive parallel tempering (APT)[29] which uses adaptive RWM together with inverse temperature adaptation, is summarized in Algorithm 3. The temperature adaptation in Algorithm 3 implements the ASM adapta-

Algorithm 3 $X_{1:n}^{(1)} \leftarrow \mathrm{APT}(\ell,\, x_0,\, n,\, L)$

Initialize $\xi_0^{(i)}$, set $\rho_0^{(1:L-1)} \leftarrow 0$, $\beta_0^{(i)} = i^{-1}$ for $i \in \{1{:}L\}$, $X_0^{(i)} \leftarrow x_0$, and $P_0^{(i)} \leftarrow \ell_{\beta_0^{(i)}}(x_0)$.
for $k = 1, \ldots, n$ **do**:
  for $i = 1, \ldots, L$ **do**:
    $(\tilde{X}_k^{(i)}, \tilde{P}_k^{(i)}; \tilde{A}_k^{(i)}, \tilde{Z}_k^{(i)}) \leftarrow \mathrm{RWMStep}\big(X_{k-1}^{(i)}, \beta_{k-1}^{(i)} P_{k-1}^{(i)}, \tilde{\ell}_{\beta_{k-1}^{(i)}}, \mathrm{Shape}(\xi_k^{(i)})\big)$
    $\xi_k^{(i)} \leftarrow \mathrm{Adapt}(k, \xi_{k-1}^{(i)}, \tilde{X}_k^{(i)}, \tilde{Z}_k^{(i)}, \tilde{A}_k^{(i)})$
  $\tilde{L}_k^{(i)} \leftarrow \tilde{P}_k^{(i)}/\beta_{k-1}^{(i)}$ for $i = 1, \ldots, L$.
  $(X_k^{(1:L)}, L_k^{(1:L)}, A_k, I_k) \leftarrow \mathrm{SwapStep}(\tilde{X}_k^{(1:L)}, \tilde{L}_k^{(1:L)}, \beta_{k-1}^{(1:L)})$
  $(\rho_k^{(1:L-1)}, \beta_k^{(1:L)}) \leftarrow \mathrm{AdaptTemp}(k, \rho_{k-1}^{(1:L-1)}, A_k, I_k)$
  $P_k^{(i)} \leftarrow \beta_k^{(i)} L_k^{(i)}$ for $i = 1, \ldots, L$.

function $\mathrm{SwapStep}(X^{(1:L)},\, L^{(1:L)},\, \beta^{(1:L)})$:
  $I \sim U\{1, \ldots, L-1\}$, $A \leftarrow \min\left\{1, \exp\left((\beta^{(I)} - \beta^{(I+1)})(L^{(I+1)} - L^{(I)})\right)\right\}$ and $U \sim U(0,1)$
  if $U \leq A$ **then** swap $(X^{(I+1)}, X^{(I)}) \leftarrow (X^{(I)}, X^{(I+1)})$ and $(L^{(I+1)}, L^{(I)}) \leftarrow (L^{(I)}, L^{(I+1)})$
  return $(X^{(1:L)}, L^{(1:L)}, A, I)$

function $\mathrm{AdaptTemp}(k, \rho^{(1:L)},\, A,\, I)$:
  $\tilde{\rho}^{(I)} \leftarrow \rho^{(I)} + \gamma_k(A - \alpha^*)$, and $\tilde{\rho}^{(i)} \leftarrow \rho^{(i)}$ for $i \neq I$.
  $T^{(1)} \leftarrow 1$ and $T^{(i+1)} = T^{(i)} + \exp(\tilde{\rho}^{(i)})$ for $i = 2, \ldots, L$.
  return $(\tilde{\rho}^{(1:L-1)}, \tilde{\beta}^{(1:L)})$ where $\tilde{\beta}^{(i)} = 1/T^{(i)}$.

tion (2) to $\rho^{(i)}$, which parameterize the log-differences of the consecutive temperatures, via

$1/\beta^{(i+1)} = 1/\beta^{(i)} + e^{\rho^{(i)}}$. The mean acceptance rate of the swaps between levels $\{i-1, i\}$ was shown in[29] to be monotonically decreasing with respect to $\rho^{(i)}$, and therefore the algorithm converges to $\beta_*^{(1:L)}$ which ensures constant $\alpha_* = 0.234$ acceptance rate of the swaps. This rule of thumb, which is equivalent with RWM rule (R2), is loosely justified in the APT context[36], and appears to work well.

In a multimodal case, the lower level RWM moves act 'locally', exploring one mode at a time. The AM often works well under unimodality, but in the multimodal case, the AM proposal may become too wide leading to poor acceptance rate. Therefore, we suggest to use either ASM+AM or RAM within APT. We use the step size $\gamma_k = (L-1)(k+1)^{-2/3}$ for the temperature adaptation, which is similar to the one suggested with ASM, with an additional factor accouting for random update to one of $L-1$ temperature difference adptations.

In Bayesian statistics, the target distribution $\pi(x) \propto \mathrm{pr}(x)\mathrm{lik}(x)$, product of the prior density and the likelihood, respectively. Equivalently, the log-target factorizes to $\ell(x) = \ell_{\mathrm{pr}}(x) + \ell_{\mathrm{lik}}(x)$. Often, the prior distribution is regular and unimodal, and the multimodality is caused by the likelihood term only. In this case, it is advisable to 'temper' only the log-likelihood part, so that $\tilde{\ell}_{\beta^{(i)}}(x) := \ell_{\mathrm{pr}}(x) + \beta^{(i)}\ell_{\mathrm{lik}}(x)$[17]. This leads to slight modification of Algorithm 3, so that $\tilde{L}_k^{(i)} \leftarrow \big(\tilde{P}_k^{(i)} - \ell_{\mathrm{pr}}(\tilde{X}_k^{(i)})\big)/\beta_{k-1}^{(i)}$ and $P_k^{(i)} \leftarrow \ell_{\mathrm{pr}}(X_k^{(i)}) + \beta_k^{(i)} L_k^{(i)}$.

It is possible to further refine the APT algorithm by using different swap strategies, for instance by alternating between odd and even swaps with large L[42], or to reduce the number of levels L adaptively[24]. Multimodal distributions are considered also in the framework presented in[31], which consists of an 'exploratory' phase aiming to find the modes, and a consequent sampling phase. The APT could be used in the former phase. It is possible to extend the PT by adding a transformation to the swap step, based on information of the modes[43].

5. Dynamic models with particle filters

Hidden Markov models (HMMs, also known as state-space models) are a flexible class of models often used in modern time-series analysis[13,10]. The data $y^{(1:T)} = (y^{(1)}, \ldots, y^{(T)})$ are modeled conditionally independent given the latent Markov process $x^{(1:T)}$, with initial distribution $f_\theta^{(1)}(x^{(1)})$ and transitions $f_\theta^{(k)}(x^{(k)} \mid x^{(k-1)})$, and with observation densities $g_\theta^{(k)}(y^{(k)} \mid x^{(k)})$, all parameterized by (hyper)parameters θ with prior $\mathrm{pr}(\theta)$. The full joint posterior of the parameters and the latent state satisfies $\pi(\theta, x^{(1:T)}) \propto \mathrm{pr}(\theta) p_\theta(x^{(1:T)}, y^{(1:T)})$ where

$$p_\theta(x^{(1:T)}, y^{(1:T)}) = f_\theta^{(1)}(x^{(1)}) g_\theta^{(1)}(y^{(1)} \mid x^{(1)}) \prod_{k=2}^{T} f_\theta^{(k)}(x^{(k)} \mid x^{(k-1)}) g_\theta^{(k)}(y^{(k)} \mid x^{(k)}).$$

In the context of HMMs, the parameters $\theta \in \mathbb{R}^d$ are often of moderate dimension, but the dimension of the latent process $x^{(1:T)}$ is proportional to the data record length T, making direct MCMC for $(\theta, x^{(1:T)})$ inefficient. The pioneering work[1] introduced several 'particle MCMC' methods, which uses particle filters, a generic class of Monte Carlo algorithms tailored for HMMs, with MCMC methods such that the resulting algorithm will be valid MCMC for the full posterior π. Adaptive MCMC has been suggested to automatically design proposals for the hyperparameters θ within particle MCMC[1,40,30], and we discuss some guidelines how this may be done in practice.

Algorithms 4 and 5 summarize the two distinct particle MCMC methods, the particle marginal Metropolis-Hastings (PMMH) and the particle Gibbs (PG)[1], with adaptation. The algorithms are written with generic particle filter parameters: the 'proposals' $M_\theta^{(k)}$ and the 'potentials' $G_\theta^{(k)}$. The simplest valid choice is $M_\theta^{(k)} \equiv f_\theta^{(k)}$ and $G_\theta^{(k)}(x^{(k)}) = g_\theta^{(k)}(y^{(k)} \mid x^{(k)})$, which is known as the bootstrap filter[16], but any other choice is valid as long as

$$M_\theta^{(1)}(x^{(1)})G_\theta^{(1)}(x^{(1)}) \prod_{k=2}^{T} M_\theta^{(k)}(x^{(k)} \mid x^{(k-1)})G_\theta^{(k)}(x^{(k)}) \equiv p_\theta(x^{(1:T)}, y^{(1:T)}),$$

as a function of $(\theta, x^{(1:T)})$. (Note that both $M_\theta^{(k)}$ and $G_\theta^{(k)}$ may depend on $y^{(1:T)}$, but this dependence is suppressed from the notation.)

The functions $\mathrm{PF}(\,\cdot\,)$ and $\mathrm{CPF}(\,\cdot\,)$ are abstractions of the 'particle filter' and the 'conditional particle filter,' respectively[1]. More specifically, $\mathrm{PF}(\,\cdot\,, N)$ refers to the particle filter run with N particles and the given parameters, and the output consists of the logarithm of the marginal likelihood estimate, and one trajectory picked from the generated particle system. PF only requires that $M_\theta^{(k)}(\,\cdot\mid x)$ can be sampled from, and that (logarithm of) $G_\theta^{(k)}$ can be calculated. The call of $\mathrm{CPF}(\,\cdot\,)$ is similar, with the third argument being the previous (reference) trajectory. We refer the reader to consult the original paper[1] for details, but remark that the backward sampling variant of the CPF[49,26] may be used if the (logarithmic) density values of $M_\theta^{(k)}(x' \mid x)$ can be calculated. It is recommended if applicable, because it can improve the performance dramatically, and is provably stable with large T[25].

Algorithm 4 $(\Theta_{1:n}, X_{1:n}^{(1:T)}) \leftarrow \mathrm{AdaptivePMMH}(\ell_{\mathrm{pr}}, \theta_0, n, N, M_\theta^{(1:T)}, G_\theta^{(1:T)})$

Initialize ξ_0, $\Theta_0 \leftarrow \theta_0$, $P_0 \leftarrow \ell_{\mathrm{pr}}(\Theta_0)$ and $(V_0, X_0^{(1:T)}) \leftarrow \mathrm{PF}(M_{\Theta_0}^{(1:T)}, G_{\Theta_0}^{(1:T)}, N)$
for $k = 1, \ldots, n$ **do**:
  $\tilde{\Theta}_k \leftarrow \Theta_{k-1} + \mathrm{Shape}(\xi_{k-1})Z_k$ where $Z_k \sim q$
  $\tilde{P}_k \leftarrow \ell_{\mathrm{pr}}(\tilde{\Theta}_k)$, $(\tilde{V}_k, \tilde{X}_k^{(1:T)}) \leftarrow \mathrm{PF}(M_{\tilde{\Theta}_k}^{(1:T)}, G_{\tilde{\Theta}_0}^{(1:T)}, N)$ and $U_k \sim U(0,1)$
  if $U_k \le \alpha_k := \min\{1, \exp(\tilde{P}_k + \tilde{V}_k - P_{k-1} - V_{k-1})\}$ **then**:
    $(\Theta_k, P_k, V_k, X_k^{(1:T)}) \leftarrow (\tilde{\Theta}_k, \tilde{P}_k, \tilde{V}_k, \tilde{X}_k^{(1:T)})$
  else:
    $(\Theta_k, P_k, V_k, X_k^{(1:T)}) \leftarrow (\Theta_{k-1}, P_{k-1}, V_{k-1}, X_{k-1}^{(1:T)})$
  $\xi_k \leftarrow \mathrm{Adapt}(k, \xi_{k-1}, \Theta_k, Z_k, \alpha_k)$.

Algorithm 5 $(\Theta_{1:n}, X_{1:n}^{(1:T)}) \leftarrow \mathrm{AdaptivePG}(\ell_{\mathrm{pr}}, \theta_0, n, N, M_\theta^{(1:T)}, G_\theta^{(1:T)})$

Initialize ξ_0, $\Theta_0 \leftarrow \theta_0$, $P_0 \leftarrow \ell_{\mathrm{pr}}(\Theta_0)$ and $(-, X_0^{(1:T)}) \leftarrow \mathrm{PF}(M_{\Theta_0}^{(1:T)}, G_{\Theta_0}^{(1:T)}, N)$
for $k = 1, \ldots, n$ **do**:
  $\tilde{\Theta}_k \leftarrow \Theta_{k-1} + \mathrm{Shape}(\xi_{k-1})Z_k$ where $Z_k \sim q$, and $\tilde{P}_k \leftarrow \ell_{\mathrm{pr}}(\tilde{\Theta}_k)$
  $V_{k-1} \leftarrow \log p_{\Theta_{k-1}}(X_{k-1}^{(1:T)}, y^{(1:T)})$, $\tilde{V}_k \leftarrow \log p_{\tilde{\Theta}_k}(X_{k-1}^{(1:T)}, y^{(1:T)})$ and $U_k \sim U(0,1)$
  if $U_k \le \alpha_k := \min\{1, \exp(\tilde{P}_k + \tilde{V}_k - P_{k-1} - V_{k-1})\}$ **then**:
    $(\Theta_k, P_k) \leftarrow (\tilde{\Theta}_k, \tilde{P}_k)$
  else:
    $(\Theta_k, P_k) \leftarrow (\Theta_{k-1}, P_{k-1})$
  $\xi_k \leftarrow \mathrm{Adapt}(k, \xi_{k-1}, \Theta_k, Z_k, \alpha_k)$.
  $X_k^{(1:T)} \leftarrow \mathrm{CPF}(M_{\Theta_k}^{(1:T)}, G_{\Theta_k}^{(1:T)}, X_{k-1}^{(1:T)}, N)$

Table 2: Summary of recommended algorithms for specific problems and their step sizes.

Method	PMMH	PG	MwG-1	MwG-d	PT	γ_k
AM	✓	×	×	×	×	$(k+1)^{-1}$
ASM	×	×	✓	×	×	$k^{-2/3}$
ASM+AM	×	✓	×	✓	✓	$(k+1)^{-2/3}$
RAM	×	✓	×	✓	✓	$\min\{1, d \cdot (k+1)^{-2/3}\}$

In principle, it is possible to apply any simple RWM adaptation of Section 3 within both Algorithms 4 and 5. However, in case of PMMH (Algorithm 4), the mean acceptance rate depends both on Shape(ξ_k), and on the number of particles N, making it difficult to know what desired acceptance rate value α_* should be used. Therefore, it is simpler to employ the AM adaptation, which does not rely on acceptance rate, but only on the posterior covariance, which is independent of N. The number of particles N needs to be chosen per application; some guidelines are given with related theoretical developments[12,38]. When using adaptation within PMMH, the number of particles may be best chosen slightly higher than the guidelines suggest (yielding at least 10% acceptance rate, say), in order to avoid potential instability of the adaptation.

In the case of particle Gibbs, the update of θ is a Metropolis-within-Gibbs update targetting the posterior conditional $\theta \mid x^{(1:T)}$. This step is independent of N, and the acceptance rate remains an effective proxy for adaptation. Therefore, we suggest to use either AM+ASM or the RAM adaptation with PG. The 'global' nature of AM adaptation, as discussed in Section 4, makes it inappropriate for sampling the conditional distributions, which are typically more concentrated than the posterior marginal.

It may be possible to design more efficient independent proposals for the PMMH, by fitting a mixture distribution to the posterior marginal of θ[40,23]. This may be achieved by first running Algorithm 4 or 5, and then using the simulated samples for mixture fitting.

6. Discussion

We reviewed a set of adaptive MCMC methods applicable for some general model classes. Our focus was on relatively simple methods, which require minimal user specification. More refined methods may improve the efficiency of the methods, but often come with a cost of further user specification, in the form of more careful choice of algorithm or their parameters.

Adaptation may be applied in a straightforward manner with hierarchical models, by using multiple independent adaptations for individual Metropolis-within-Gibbs updates of either single parameters or blocks of parameters[20,35,44]. This avoids conjugacy constraints, and using block updates for tightly correlated variables may lead to improved mixing. Some variables could also be updated by pure Gibbs moves (if perfect sampling of the conditional is possible). However, to the knowledge of the author, there is no general-purpose software that would allow for this, even though such an extension of a BUGS-type implementation would be technically straightforward.

Table 2 summarizes the recommendations which RWM adaptations are appropriate in different contexts: dynamic models (PMMH and PG methods), hierarchical models

(Metropolis-within-Gibbs, univariate and multivariate update), and with multimodal targets (PT). The recommended step size sequence is also shown.

Unfortunately, all MCMC methods come with their strengths and weaknesses, and therefore the 'end user' may need to make certain choices. Hamiltonian Monte Carlo (HMC) type methods, such as those implemented in STAN software[14], have recently become very popular. They have shown great promise for challenging inference problems, but also come with limitations. For instance, HMC cannot be used to sample discrete variables, and the model may need to be re-scaled and/or reparameterized before inference. The more domain-specific methods, such as particle MCMC in the time-series context, tend also to outperform general-purpose methods, such as HMC. Inference software that would allow for flexibly using all successful samplers to date, including the HMC type methods, Gibbs sampling, particle MCMC and adaptive methods, could provide a way forward and push the boundaries of ergonomic practical Bayesian inference.

7. Acknowledgments

The author was supported by Academy of Finland grants 274740, 312605 and 315619.

8. Bibliography

[1] Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, *72*(3), 269–342.

[2] Andrieu, C. & Moulines, É. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *Ann. Appl. Probab.*, *16*(3), 1462–1505.

[3] Andrieu, C. & Robert, C. P. (2001). Controlled MCMC for optimal sampling. Technical Report Ceremade 0125, Université Paris Dauphine.

[4] Andrieu, C. & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statist. Comput.*, *18*(4), 343–373.

[5] Atchadé, Y. & Fort, G. (2010). Limit theorems for some adaptive MCMC algorithms with subgeometric kernels. *Bernoulli*, *16*(1), 116–154.

[6] Atchadé, Y. F. & Rosenthal, J. S. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, *11*(5), 815–828.

[7] Bai, Y., Roberts, G. O., & Rosenthal, J. S. (2011). On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Advances and Applications in Statistics*, *21*(1), 1–54.

[8] Benveniste, A., Métivier, M., & Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Number 22 in Applications of Mathematics. Springer-Verlag.

[9] Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, *59*(1), 65–98.

[10] Cappé, O., Moulines, E., & Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer.

[11] Dongarra, J. J., Bunch, J. R., Moler, C. B., & Stewart, G. W. (1979). *LINPACK Users' Guide*. Society for Industrial and Applied Mathematics.

[12] Doucet, A., Pitt, M. K., Deligiannidis, G., & Kohn, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, *102*(2), 295–313.

[13] Durbin, J. & Koopman, S. J. (2012). *Time Series Analysis by State Space Methods* (2nd ed.). New York: Oxford University Press.

[14] Gelman, A., Lee, D., & Guo, J. (2015). Stan: A probabilistic programming language for Bayesian inference and optimization. *Journal of Educational and Behavioral Statistics*, *40*(5), 530–543.

[15] Gilks, W. R., Roberts, G. O., & Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration. *J. Amer. Statist. Assoc.*, *93*(443), 1045–1054.

[16] Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, *140*(2), 107–113.

[17] Gwiazda, P., Miasojedow, B., & Rosińska, M. (2016). Bayesian inference for age-structured population model of infectious disease with application to varicella in Poland. *J. Theoret. Biol.*, *407*, 38–50.

[18] Haario, H., Laine, M., Mira, A., & Saksman, E. (2006). DRAM: Efficient adaptive MCMC. *Statist. Comput.*, *16*(4), 339–354.

[19] Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, *7*(2), 223–242.

[20] Haario, H., Saksman, E., & Tamminen, J. (2005). Componentwise adaptation for high dimensional MCMC. *Comput. Statist.*, *20*(2), 265–274.

[21] Hastie, D. (2005). *Toward Automatic Reversible Jump Markov Chain Monte Carlo*. PhD thesis, University of Bristol.

[22] Jarner, S. F. & Hansen, E. (2000). Geometric ergodicity of Metropolis algorithms. *Stochastic Process. Appl.*, *85*(2), 341–361.

[23] Knape, J. & De Valpine, P. (2012). Fitting complex population models by combining particle filters with Markov chain Monte Carlo. *Ecology*, *93*(2), 256–263.

[24] Łącki, M. K. & Miasojedow, B. (2016). State-dependent swap strategies and automatic reduction of number of temperatures in adaptive parallel tempering algorithm. *Statist. Comput.*, *26*(5), 951–964.

[25] Lee, A., Singh, S. S., & Vihola, M. (to appear). Coupled conditional backward sampling particle filter. *Ann. Statist.*, *to appear*.

[26] Lindsten, F., Jordan, M. I., & Schön, T. B. (2014). Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.*, *15*(1), 2145–2184.

[27] Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS—a Bayesian modelling framework: concepts, structure, and extensibility. *Statist. Comput.*, *10*(4), 325–337.

[28] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.*, *21*(6), 1087–1092.

[29] Miasojedow, B., Moulines, E., & Vihola, M. (2013). An adaptive parallel tempering algorithm. *J. Comput. Graph. Statist.*, *22*(3), 643–664.

[30] Peters, G. W., Hosack, G. R., & Hayes, K. R. (2010). Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC). Preprint arXiv:1005.2238.

[31] Pompe, E., Holmes, C., & Łatuszyński, K. (2018). A framework for adaptive MCMC targeting multimodal distributions. Preprint arXiv:1812.02609.

[32] Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*, 400–407.

[33] Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, *7*(1), 110–120.
[34] Roberts, G. O. & Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Probab.*, *44*(2), 458–475.
[35] Roberts, G. O. & Rosenthal, J. S. (2009). Examples of adaptive MCMC. *J. Comput. Graph. Statist.*, *18*(2), 349–367.
[36] Roberts, G. O. & Rosenthal, J. S. (2014). Minimising MCMC variance via diffusion limits, with an application to simulated tempering. *Ann. Appl. Probab.*, *24*(1), 131–149.
[37] Saksman, E. & Vihola, M. (2010). On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. *Ann. Appl. Probab.*, *20*(6), 2178–2203.
[38] Sherlock, C., Thiery, A. H., Roberts, G. O., & Rosenthal, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Ann. Statist.*, *43*(1), 238–275.
[39] Siltala, L. & Granvik, M. (2020). Asteroid mass estimation with the robust adaptive Metropolis algorithm. *Astronomy & Astrophysics*, *633*(A46).
[40] Silva, R., Giordani, P., Kohn, R., & Pitt, M. (2009). Particle filtering within adaptive Metropolis Hastings sampling. Preprint arXiv:0911.0230.
[41] Swendsen, R. H. & Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Phys. Rev. Lett.*, *57*(21), 2607–2609.
[42] Syed, S., Bouchard-Côté, A., Deligiannidis, G., & Doucet, A. (2019). Non-reversible parallel tempering: an embarassingly parallel MCMC scheme. Preprint arXiv:1905.02939.
[43] Tawn, N. G. & Roberts, G. O. (2019). Accelerating parallel tempering: Quantile tempering algorithm (QuanTA). *Adv. in Appl. Probab.*, *51*(3), 802–834.
[44] Vihola, M. (2010a). Grapham: Graphical models with adaptive random walk Metropolis algorithms. *Comput. Statist. Data Anal.*, *54*(1), 49–54.
[45] Vihola, M. (2010b). *On the convergence of unconstrained adaptive Markov chain Monte Carlo algorithms*. PhD thesis, University of Jyväskylä.
[46] Vihola, M. (2011a). Can the adaptive Metropolis algorithm collapse without the covariance lower bound? *Electron. J. Probab.*, *16*, 45–75.
[47] Vihola, M. (2011b). On the stability and ergodicity of adaptive scaling Metropolis algorithms. *Stochastic Process. Appl.*, *121*(12), 2839–2860.
[48] Vihola, M. (2012). Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statist. Comput.*, *22*(5), 997–1008.
[49] Whiteley, N. (2010). Discussion on Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, *72*(3), 306–307.