UNIVERSITY OF
OXFORD

DEPARTMENT OF
STATISTICS

Multicore adaptive MCMC for multimodal distributions

A thesis submitted in partial fulfillment of the requirements for
transfer of status from PRS to DPhil

July 2018

Emilia Pompe
St Peter's College

Department of Statistics
University of Oxford

Contents

Acknowledgements

I wish to thank my supervisors, Chris Holmes and Krys Łatuszyński, for their ideas, support and patience. Additionally, my thanks are extended to Louis Aslett for his advice on creating the `R` package `MultiMCMC`.

Chapter 1

Introduction and literature review

Poor mixing of standard Markov Chain Monte Carlo (MCMC) methods, such as the Metropolis-Hastings algorithm or Hamiltonian Monte Carlo, on multimodal target distributions with isolated modes is a well-described problem in statistics. Sequential Monte Carlo (SMC) has proven to outperform MCMC on this task (see [Jasra et al., 2015]), its robust behaviour is however strongly enhanced if the Markov kernel used within the SMC algorithm mixes well between the modes. Therefore, constructing an MCMC algorithm which enables fast exploration of the state space for complicated target functions is still of great interest.

In this report π will typically denote the target distribution and $\mathcal{X}$ the state space on which it is defined. With a slight abuse of notation we will denote the density corresponding to the target distribution with the same letter π.

Numerous MCMC methods have been proposed to address the issue of multimodality. We present an overview of those in the remaining part of this chapter, dividing them into categories based on the main underlying concept. In Chapter 2 we present our MCMC method designed to overcome the problem of multimodality Multicore Adaptive MCMC for Multimodal Distributions, and discuss its main properties. Furthermore, we define a new class of algorithms, Auxiliary Variable Adaptive MCMC. We prove some general ergodic results for the whole class before specialising to the case of our proposed algorithm.

In Chapter 3 we outline issues related to the efficient implementation of the algorithm and illustrate the performance of our method with toy examples. We conclude with a summary of our results in Chapter 4, indicating also possible directions of future work.

1.1 Tempering-based methods

The key idea behind tempering-based methods, such as parallel or simulated tempering, relies on observing that raising a multimodal distribution π to the power $0 < \beta < 1$ makes it more "spread out" and the low probability regions are not such any more. Hence, it is easier to collect MCMC samples covering the whole state space. Since eventually π remains the distribution of interest, some tricks are needed to obtain the samples from π using samples from π^β as an intermediate step. It is worth pointing out that parallel and simulated tempering are by far the most commonly used MCMC methods for multimodal distributions. In fact, only these two algorithms are available as part of a CRAN `R` ([R Core Team, 2014]) package, with `R` being arguably the most popular programming language for data analysis. We present details regarding both these methods below. We discuss also briefly a slightly less popular method called "tempered transitions".

The parallel tempering method, proposed by [Geyer, 1991], is based on simulating a Markov chain defined on $\underbrace{\mathcal{X} \times \ldots \times \mathcal{X}}_{N+1}$ targeting

$$\bar{\pi}(x_0, x_1, \ldots, x_N) \propto \prod_{j=0}^{N} \pi(x_j)^{\beta_j} \tag{1.1}$$

for a sequence of parameters $\beta_0, \ldots, \beta_N$ (often called "inverse temperatures") satisfying $0 \leq \beta_0 < \ldots < \beta_N = 1$. Hence, we may view this as having $N+1$ chains on $\mathcal{X}$, each of them targeting the tempered distribution π^{β_j} for $j = 0, \ldots, N$. In particular the last chain targets the distribution of interest. Let $x_j^{(n)}$ denote the state of chain j at iteration n. The algorithm alternates between the following two types of moves.

1. **Update moves**: for $j = 0, \ldots, N$ propose a new state $y \sim q_j\left(\cdot | x_j^{(n)}\right)$ and accept this move with probability
$$\min\left[1, \frac{\pi(y)^{\beta_j}}{\pi\left(x_j^{(n)}\right)^{\beta_j}} \frac{q_j\left(x_j^{(n)}|y\right)}{q_j\left(y|x_j^{(n)}\right)}\right].$$
Set $x_j^{(n+1)} = x_j^{(n)}$ otherwise.

2. **Temperature swaps**: sample $j \sim \text{Unif}(0, \ldots, N-1)$. Propose $(x_j^{(n+1)}, x_{j+1}^{(n+1)}) := (x_{j+1}^{(n)}, x_j^{(n)})$ and accept this move with probability
$$\min\left[1, \left(\frac{\pi\left(x_{j+1}^{(n)}\right)}{\pi\left(x_j^{(n)}\right)}\right)^{\beta_j - \beta_{j+1}}\right].$$
Set $\left(x_j^{(n+1)}, x_{j+1}^{(n+1)}\right) := \left(x_j^{(n)}, x_{j+1}^{(n)}\right)$ otherwise.

The acceptance probabilities are designed in a way such that detailed balance holds for (1.1). The rationale behind this method is that the chains targeting π^{β_j} for small β_j move freely around the state space via update moves and this fast mixing is then transferred to the chains corresponding to larger values of β_j via temperature swaps. The design of the sequence of temperatures is of critical importance for ensuring that the acceptance ratio of swaps between temperatures is reasonably high. If the swaps are rare, the process of transferring well-mixed samples from higher to lower temperatures might be too slow. The initial rule of thumb was to construct a temperature ladder following the geometric progression (see e.g. [Woodard et al., 2009b]). More recent results, however, show that this choice should be linked with the acceptance rate of temperature swaps, which we discuss later in this section.

Simulated tempering was proposed by [Marinari and Parisi, 1992]. Here we consider a target distribution $\tilde{\pi}$ on an augmented state space $\mathcal{X} \times \{0, \ldots, N\}$ defined as

$$\tilde{\pi}(x, j) \propto \pi(x)^{\beta_j} \quad \text{for } j = 0, \ldots, N \text{ and } 0 \leq \beta_0 < \ldots < \beta_N = 1. \tag{1.2}$$

Thus the conditional distribution $\tilde{\pi}(x, j)$ given $j = N$ is the distribution of interest. Let $(x^{(n)}, j^{(n)})$ denote the state of the chain at time n. The algorithm consists of moves of two types.

1. **Update moves**: (with probability p) sample $y \sim q_{j^{(n)}}\left(\cdot|x^{(n)}\right)$ and set $(x^{(n+1)}, j^{(n+1)}) := \left(y, j^{(n)}\right)$ with probability
$$\min\left[1, \frac{\pi(y)^{\beta_{j^{(n)}}}}{\pi\left(x^{(n)}\right)^{\beta_{j^{(n)}}}} \frac{q_{j^{(n)}}\left(x^{(n)}|y\right)}{q_{j^{(n)}}\left(y|x^{(n)}\right)}\right].$$
Set $\left(x^{(n+1)}, j^{(n+1)}\right) := \left(x^{(n)}, j^{(n)}\right)$ otherwise.

2. **Changes of temperature**: (with probability $1-p$) sample $j \sim \text{Unif}(j^{(n)} - 1, j^{(n)} + 1)$ if $j^{(n)} \in \{1, \ldots, N-1\}$. If $j^{(n)} = 0$, set $j := 1$, if $j^{(n)} = N$, set $j := N - 1$. Set $\left(x^{(n+1)}, j^{(n+1)}\right) := \left(x^{(n)}, j\right)$ with probability
$$\min\left[1, \pi\left(x^{(n)}\right)^{\beta_j - \beta_{j^{(n)}}}\right].$$
Set $\left(x^{(n+1)}, j^{(n+1)}\right) := \left(x^{(n)}, j^{(n)}\right)$ otherwise.

Analogously to parallel tempering, the choice of temperatures is crucial for the efficiency of the algorithm. Another potential problem with this method is that the marginal distribution of $\tilde{\pi}$ (defined by (1.2)) with respect to j may have a very small probability of $j = N$. In such case the number of samples that one can use to approximate π will also be small. To overcome this problem, $\tilde{\pi}$ may be redefined as follows:

$$\tilde{\pi}(x, j) \propto c_j \pi(x)^{\beta_j} \quad \text{for } j = 0, \ldots, N \text{ and } 0 \leq \beta_0 < \ldots < \beta_N = 1. \tag{1.3}$$

for c_i such that the marginal probability of $\tilde{\pi}$ with respect to j is close to uniform over $\{0, \ldots, N\}$. Note, however, that achieving the exact uniform distribution would be equivalent to setting

$$c_j^{-1} := \int \pi(x)^{\beta_j} dx \quad \text{for } j = 0, \ldots, N,$$

but if these constants were known, we would not use MCMC in the first place. In practice estimates of these normalising constants are used even though obtaining them usually requires preliminary MCMC runs.

[Neal, 1996] addressed the issue of the choice of the constants c_j by introducing the aforementioned tempered transitions method. This technique is again based on a single chain, but the constants are not needed. Firstly N pairs of transition kernels T_j and T'_j (corresponding to the inverse temperature β_j) are constructed such that they satisfy

$$\pi(x)^{\beta_j} T_j(x'|x) = \pi(x')^{\beta_j} T'_j(x|x') \quad \text{for all } x, x' \in \mathcal{X} \text{ and } j = 0, \ldots, N-1,$$

where $0 \leq \beta_0 < \ldots < \beta_N = 1$. Assume that the current state of the chain is $x^{(n)}$. To propose a new state y, the algorithm passes twice through the temperature ladder in the following way:

1. Set $x_N := x^{(n)}$. For $j = N-1, \ldots, 0$ generate $x_j \sim T_j(\cdot|x_{j+1})$.
2. Set $x'_0 := x_0$. For $j = 0, \ldots, N-1$ generate $x'_{j+1} \sim T'_j(\cdot|x'_j)$. Set $y := x'_N$.

The new state is then accepted with probability

$$\min\left[1, \left(\prod_{j=1}^{N} \pi(x_j)^{\beta_{j-1}-\beta_j}\right)\left(\prod_{j=1}^{N} \pi(x'_j)^{\beta_j-\beta_{j-1}}\right)\right].$$

The algorithm described this way fulfils the detailed balance condition for the invariant distribution π. [Neal, 1996] pointed out that there is no overall winner among the three methods discussed above and each one may outperform the others, depending on the target density.

Theoretical properties of tempering-based methods are well-understood and we present below some of the results known in the literature. We first recall briefly the relation between mixing time and the spectral gap. Let P be a π-irreducible, aperiodic, nonnegative definite and π-reversible Markov chain. Then P converges to π in total variation and the rate of this convergence is characterised by

$$\|P^n(x,\cdot) - \pi(\cdot)\|_{TV} \leq M(x)\left(1 - \mathbf{Gap}(P)\right)^n$$

for a function M such that $M(x) < \infty$ for π-a.e. $x \in \mathcal{X}$ (see e.g. [Roberts and Rosenthal, 1997]). We say that an MCMC algorithm mixes torpidly when the spectral gap decreases exponentially as a function of the dimension of the state space and we say it mixes rapidly when the gap decreases polynomially.

[Woodard et al., 2009b] studied examples of multimodal distribution on which the Metropolis-Hastings algorithm mixes torpidly, whereas parallel and simulated tempering mix rapidly. Their idea was to split the domain into chunks on which the target distribution is unimodal, bound the spectral gap in terms of mixing on each of those subsets and finally – in terms of mixing between the subsets.

In particular, parallel and simulated tempering mix rapidly on the mean field Ising model and on the weighted and unweighted mixture of Gaussian distributions with identity covariance matrices for a certain choice of temperature levels constructed by [Woodard et al., 2009b]. It is important to point out that the number of temperatures needs to grow proportionally with the dimension of the state space. In fact, [Woodard et al., 2009a] proved that if the sequence of temperatures is fixed, parallel and simulated tempering mix torpidly on the mean field Ising model.

[Woodard et al., 2009a] considered an example of a mixture of Gaussians with unequal covariance matrices, in particular they studied the following target distribution:

$$\pi(x) = \frac{1}{2}N\left(-\underbrace{(1,\ldots,1)}_{d}, \sigma_1^2 I_d\right) + \frac{1}{2}N\left(\underbrace{(1,\ldots,1)}_{d}, \sigma_2^2 I_d\right),$$

where $\sigma_1 \neq \sigma_2$. Parallel and simulated tempering turn out to mix torpidly on this example for any choice of the temperature ladder. A practical consequence of this fact is that if the target distribution contains both narrow and wide modes, a simulated tempering chain started in a wide mode may take a long time before it discovers the narrow ones. A similar result holds for a parallel tempering algorithm if starting points for all temperatures are located in the wide mode.

The problem of the choice of the optimal temperature ladder was considered by several researchers from the theoretical viewpoint. [Kone and Kofke, 2005] noticed that good results are obtained when the acceptance rate of temperature swaps is around 0.2. [Atchadé et al., 2011] proved formally that the optimal choice of inverse temperatures for parallel and simulated tempering satisfies that the average acceptance rate of the temperature swaps is close to 0.234. [Miasojedow et al., 2013] applied this result in practice, proposing an adaptive version of parallel tempering. The way the parameters are adapted is the following:

- The parameters β_j are adapted so that the acceptance rate of swaps between any two adjacent temperature levels converges to 0.234.
- Update moves are Random Walk Metropolis steps with the Gaussian proposal. The covariance matrices at each temperature level are adapted so that the acceptance rate converges to 0.234 (in the spirit of [Haario et al., 1999a] or [Vihola, 2011]).

Essentially the only parameter that remains fixed is the number of rungs in the temperature ladder. This method is therefore much more user-friendly than standard tempering techniques as it does not require virtually any tuning. Appealing as it may seem, the adaptive approach does not resolve the immanent issues of parallel tempering discussed in [Woodard et al., 2009a].

1.2 Equi-energy sampler

The equi-energy sampler was introduced by [Kou et al., 2006]. Let $\pi(x) \propto \exp(-h(x))$ be the target distribution; $h(x)$ is the so-called energy function. To present how the method works, we start with introducing a sequence of energy levels $H_0 < H_1 < \ldots < H_{N+1} = \infty$ such that $H_0 \leq \inf_{x \in \mathcal{X}} h(x)$, and an associated sequence of temperatures $1 = T_0 < T_1 < \ldots < T_N$. We now define $N+1$ distributions π_j as follows:

$$\pi_j(x) \propto \exp(-h_j(x)), \text{ where } h_j(x) = \frac{1}{T_j}(h(x) \vee H_j) \text{ for } j = 0, \ldots, N.$$

Additionally, we introduce a partition of the state space $\mathcal{X}$ into energy rings

$$D_k := \{x : h(x) \in [H_k, H_{k+1})\} \text{ for } k = 0, \ldots, N. \tag{1.4}$$

The idea of the equi-energy sampler bears resemblance to parallel tempering described in the previous section. Analogously, we employ $N+1$ chains targeting distributions π_j for $j = 0, \ldots, N$. Intuitively, for large j distribution π_j is flatter than π (in fact, its energy is bounded by H_j/T_j) and does not have problematic low probability regions separating the modes. Therefore, distributions π_j can play a similar role to the tempered distributions in parallel tempering. Finally, the fast mixing of the high-temperature chains is transferred down the temperature ladder via "equi-energy jumps" between adjacent chains.

Let $x_j^{(n)}$ denote the state of chain j at iteration n. The equi-energy algorithm starts from simulating chain N using Metropolis-Hastings steps with π_N as the invariant distribution. The generated samples (excluding the burn-in period) are grouped into empirical energy rings according to their energy level, following the concept of (1.4). Namely, an empirical energy ring $\hat{D}_k^N$ consists of samples $x_N^{(i)}$ such that $x_N^{(i)} \in D_k$. Analogous level k empirical energy rings $\hat{D}_k^j$, where $j = 0, \dots, N-1$, will be constructed for the remaining N chains after their initial burn-in periods.

The dynamics of chains j for $j = 0, \dots, N-1$ are shaped by steps of two types, as described below.

1. **Equi-energy jumps**: (with probability p) let k_n be such that the current state $x_j^{(n)}$ belongs to $\hat{D}_{k_n}^j$. Propose a new state y sampled uniformly from the empirical energy ring $\hat{D}_{k_n}^{j+1}$. Set $x_j^{(n+1)} := y$ with probability
$$\min\left[1, \frac{\pi_j(y)\pi_{j+1}\left(x_j^{(n)}\right)}{\pi_j\left(x_j^{(n)}\right)\pi_{j+1}(y)}\right],$$
otherwise set $x_j^{(n+1)} := x_j^{(n)}$. Update the empirical energy rings $\hat{D}_k^j$ for $k = 0, \dots, N$ by adding the new state to its corresponding ring.

2. **Standard M-H steps**: (with probability $1-p$) perform a standard Metropolis-Hastings step with π_j as the invariant distribution. Update the empirical energy rings $\hat{D}_k^j$ for $k = 0, \dots, N$ by adding the new state $x_j^{(n+1)}$ to its corresponding ring.

The slight difference between parallel tempering and equi-energy sampler is that the latter does not activate all the chains at the same time: chain j needs to use the knowledge of the energy rings of chain $j+1$, so it can only be started after the burn-in period of chain $j+1$. [Kou et al., 2006] demonstrated that (under mild conditions) each chain j is ergodic with π_j as the invariant distribution, therefore the chain corresponding to the lowest temperature converges to the distribution of interest. The authors discuss very briefly how proposal distributions used by each chain can be adapted, but there still remains a significant number of parameters to be tuned (energy levels and temperatures). [Schreck et al., 2013] proposed an adaptive versions of the equi-energy sampler, in which the energy rings are adapted on the fly.

1.3 Metropolis-Hastings algorithms with a special design of the proposal distribution

[Tjelmeland and Hegstad, 2001] introduced a method based on proposing moves to distant regions of the state space, at the same time ensuring that the proposed points lie in areas of high-probability. We describe below the way their transition kernel is constructed. To this end we define two proposal kernels Q_0 and Q_1 with their corresponding densities $q_0(y|x)$ and $q_1(y|x)$ and combine them in a certain way.

Let $x^{(n)}$ be the current state of the Markov chain. Let ψ be a vector of the same dimension as $x^{(n)}$, drawn from a distribution $f(\psi)$ with large variance. This is how we propose a new state $y \sim Q_0\left(\cdot|x^{(n)}\right)$. First a deterministic local maximisation procedure on the target distribution π is started from $x^{(n)} + \psi$.

Let $\mu(x^{(n)}+\psi)$ denote the point obtained using this procedure and let $\Sigma(x^{(n)}+\psi)$ denote the local covariance matrix around $\mu(x^{(n)}+\psi)$. The new state y is then proposed from the normal distribution $N\left(\mu(x^{(n)}+\psi), \Sigma(x^{(n)}+\psi)\right)$, i.e. $Q_0\left(\cdot|x^{(n)}\right)$ is the normal distribution $N\left(\mu(x^{(n)}+\psi), \Sigma(x^{(n)}+\psi)\right)$. In turn, $Q_1\left(\cdot|x^{(n)}\right)$ is given by $N\left(\mu(x^{(n)}-\psi), \Sigma(x^{(n)}-\psi)\right)$. The final transition kernel P is defined as

$$P(A|x^{(n)}) = \frac{1}{2}\int_A q_0(y|x^{(n)})\alpha_{0,1}(y|x^{(n)})dy + \frac{1}{2}\int_A q_1(y|x^{(n)})\alpha_{1,0}(y|x^{(n)})dy + \mathbb{I}_{x^{(n)}\in A} r\left(x^{(n)}\right), \tag{1.5}$$

where

$$r\left(x^{(n)}\right) = \frac{1}{2}\int q_0(y|x^{(n)})\left(1-\alpha_{0,1}(y|x^{(n)})\right)dy + \frac{1}{2}\int q_1(y|x^{(n)})\left(1-\alpha_{1,0}(y|x^{(n)})\right)dy$$

and

$$\alpha_{i,1-i}(y|x^{(n)}) = \min\left[1, \frac{\pi(y)q_{1-i}(x^{(n)}|y)}{\pi(x^{(n)})q_i(y|x^{(n)})}\right] \quad \text{for } i = 0, 1.$$

In other words, we pick $i = 0$ or $i = 1$ with probability 0.5 each, then we draw a sample from the corresponding proposal distribution q_i and we use the corresponding acceptance probability $\alpha_{0,1}$ or $\alpha_{1,0}$ for the acceptance/rejection step. This turns out to be the Metropolis-Hastings algorithm but with a different than usual choice of the acceptance probability. This, while not being optimal in the sense of minimizing the asymptotic variance (see [Peskun, 1973]), gives a smaller computational cost of one iteration (which is important in this case, since evaluating $q_i(x|y)$ is costly due to the optimisation procedure involved). Intuitively, the proposal obtained this way will exhibit the required properties and additionally, combining the two kernels as described above allows for higher acceptance rate of those moves.

[Tjelmeland and Hegstad, 2001] proposed to use this method together with standard Metropolis-Hastings steps, e.g. alternate between one hundred Random Walk Metropolis steps and one move with kernel P defined by (1.5).

Below we present a more recent approach, based on proposal via forced downhill and forced uphill Metropolis steps ([Tak et al., 2017]). This method relies on using the so-called down-up jumping density q^{DU} used for proposing the next step. Given the current state $x^{(n)}$, first an intermediate downhill proposal x' is generated from density $q^D(x'|x^{(n)})$, which is followed by an uphill proposal x^*generated from $q^U(x^*|x')$. Hence,

$$q^{DU}(x^*|x^{(n)}) = \int q^D(x'|x^{(n)})q^U(x^*|x')dx'.$$

The densities q^D and q^U are chosen such that they favour lower and higher values of the target distribution, respectively. The authors propose to use

$$q^D(x'|x^{(n)}) \propto q(x'|x^{(n)})\alpha_\epsilon^D(x'|x^{(n)}) \quad \text{and} \quad q^U(x^*|x') \propto q(x^*|x')\alpha_\epsilon^U(x^*|x'),$$

where q is an arbitrary proposal density,

$$\alpha_\epsilon^D(x'|x^{(n)}) = \min\left[1, \frac{\pi(x^{(n)}) + \epsilon}{\pi(x') + \epsilon}\right] \quad \text{and} \quad \alpha_\epsilon^U(x^*|x') = \min\left[1, \frac{\pi(x^*) + \epsilon}{\pi(x') + \epsilon}\right]$$

for some small ϵ added to avoid numerical problems. In other words, the intermediate step x' is generated repeatedly until it is accepted (the probability of acceptance at each step is given by α_D) and analogous procedure is used to generate x^* given $x^{(n)}$. Intuitively, x' will be located in the valley between modes and x^* will then more likely jump to a different mode than the one $x^{(n)}$ belonged to.

Once the sample x^* is proposed, it remains to accept/ reject it with probability such that detailed balance holds. This probability, however, turns out to be intractable. A way around this problem is to introduce an auxiliary variable z and consider a target distribution $\bar{\pi}(x, z) = \pi(x)q(x|z)$. Given x^*, z^* is generated using $q^D(z^*|x^*)$, i.e. it is again an enforced downhill move obtained in a way described above for x'. Finally, the new point (x^*, z^*) is accepted with probability

$$\alpha_J\left(x^*, z^*|x^{(n)}, z^{(n)}\right) = \min\left[1, \frac{\pi(x^*)\min\left[1, \frac{\pi(x^{(n)})+\epsilon}{\pi(z^{(n)})+\epsilon}\right]}{\pi(x^{(n)})\min\left[1, \frac{\pi(x^*)+\epsilon}{\pi(z^*)+\epsilon}\right]}\right]. \tag{1.6}$$

The procedure described above may be viewed as a Metropolis-Hastings algorithm targeting $\bar{\pi}$ with a special form of proposal distribution $q^J(x^*, z^*|x^{(n)}, z^{(n)}) = q^{DU}(x^*|x^{(n)})q^D(z^*|x^*)$. [Tak et al., 2017] prove that if the acceptance probability is given by (1.6), the detailed balance holds for $\bar{\pi}$ and consequently, RAM is ergodic for π as the marginal distribution of $\bar{\pi}$ with respect to x.

1.4 Optimisation-based methods

Several methods known in the literature are based on running preliminary searches of local maxima, which facilitates "smarter" movement of an MCMC sampler. Once the regions of high probability are identified, the MCMC sampler may be designed so that it proposes moves to distant regions of high probability, which enables faster mixing. As we shall see later, the method that we propose in Chapter 2 also falls into this category.

Recall that the adaptive methods described in the previous sections (e.g. adaptive parallel tempering) are based on estimating a single covariance matrix for each chain and updating the proposal distribution based on this matrix. However, if the modes of the target distribution have very different local covariance structures, the optimal proposal distributions are different for regions associated with different modes. Observe that in the methods presented so far the sampling algorithms do not have the knowledge of the locations of the modes. In particular, there is no mechanism of controlling close to which mode a given state of the Markov chain is. The optimisation-based approach gives a chance to estimate the covariance matrix separately for each mode and use a different proposal distribution adjusted to the local shape rather than the global empirical covariance matrix.

We first review several research papers devoted to the idea of Smart Darting Monte Carlo, introduced in [Andricioaei et al., 2001]. Suppose we know the locations of M distinct modes of the target distribution

μ_i for $i = 1, \ldots, M$. Around each of them we create an ϵ-sphere with ϵ small enough so that the spheres do not overlap. We define the following displacement vectors:

$$D_{ij} = \mu_i - \mu_j \text{ for } i \neq j.$$

Suppose the current state of the Monte Carlo algorithm is $x^{(n)}$. We can now perform steps of two types:

1. **Local moves**: (with probability p) perform a Random Walk Metropolis step.
2. **Jump moves**: (with probability $1-p$) if $x^{(n)}$ is inside an ϵ-sphere centred at μ_j, propose a jump to one of the remaining spheres, i.e. pick randomly a vector D_{ij} and propose $y = x + D_{ij}$. If $x^{(n)}$ is outside any ϵ-sphere, stay in the same place.

The acceptance probabilities in both cases are constructed in such a way that the detailed balance condition holds. The rationale behind the idea of the jumps of deterministic lengths D_{ij} is that the algorithm moves to the "corresponding" point in a different sphere. The value of the target density is therefore expected to have the same order of magnitude for the proposed state and the current state, which ensures that the probability of acceptance is relatively high.

[Sminchisescu and Welling, 2011] made an attempt to generalise the Smart Darting Monte Carlo method. The generalisation is two-fold: jumping regions are allowed to have arbitrary volume and shape (they do not need to be ϵ-spheres) and they are allowed to overlap. Since the shapes of the jumping regions may not be the same, proposing the next point using displacement vectors is not possible. Instead, when $x^{(n)}$ enters a jumping region, [Sminchisescu and Welling, 2011] propose a new region with probability proportional to its volume. Then a new point is sampled uniformly inside the proposed region, which is followed by an acceptance/rejection step to ensure that detailed balance is fulfilled. The acceptance probability must take into account the number of jumping regions to which $x^{(n)}$ and the new proposed point belong. As observed in [Sminchisescu and Welling, 2011], in practice it is difficult to define the optimal shape and volume of the jumping regions but at the same time defining them in a way that does not correspond to the shape of π may lead to prohibitively low acceptance rates of the jumps. What is more, the methods presented above are not adaptive. [Ahn et al., 2013] extended Darting Monte Carlo substantially by introducing updates of the jumping regions and parameters of the proposal distribution at regeneration times, which partly addressed the issues listed above.

The method presented by [Ahn et al., 2013], Regeneration Darting Monte Carlo (RDMC), relies similarly to Darting Monte Carlo on a local sampler (the authors use Hamiltonian Monte Carlo for this purpose) and a jump kernel. The latter is an independence sampler with the proposal distribution q close as much as possible to the target. In this case q is a mixture of Gaussian distributions truncated to the jumping regions. Gaussian distributions are more flexible (through the choice of the covariance matrix), so they may potentially provide a better fit to the target distribution than the uniform distributions used in [Sminchisescu and Welling, 2011]. The RDMC algorithm additionally identifies the times when the Markov chain regenerates (see [Mykland et al., 1995]) and updates the parameters of the proposal q, such as shapes of the jumping regions and covariance matrices of the Gaussian components. In particular, the new approach allows for discovering new modes while the algorithms runs and including them on the fly by adding a new jumping region at a regeneration time. Thus, the method becomes parallelisable to a significant extent, as mode searching optimisation procedures may run on different cores and communicate with the main MCMC sampler at regeneration times. Moreover, since the parameters are adapted only at regeneration times, detailed balance is preserved. Unfortunately, [Ahn et al., 2013] do not reveal the details of their procedure and the settings of their experiments are not fully reported.

Another optimisation-based method, Wormhole Hamiltonian Monte Carlo, was introduced by [Lan et al., 2014] as an extension of Riemanian Manifold HMC (RMHMC, see [Girolami and Calderhead, 2011]). Let us recall briefly how the HMC (the standard version) and RMHMC methods work.

Let $\pi(q)$ be the target distribution of interest (q is often referred to as the position variable). We introduce an auxiliary momentum variable p of the same dimension as q following the multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix M. In the standard version of HMC (see [Duane et al., 1987]) the matrix M (the mass matrix) is set to identity. Let us now define the potential energy $U(q)$ and the kinetic energy $K(p)$ as

$$U(q) = -\log(\pi(q)) \quad \text{and} \quad K(p) = \frac{p^T M^{-1} p}{2}.$$

This corresponds to the minus loglikelihood of $\pi(q)$ and $N(0, M)$ respectively (plus a constant). The Hamiltonian function can be written as follows:

$$H(q, p) = U(q) + K(p). \tag{1.7}$$

Let $\dot{q}$ and $\dot{p}$ denote the derivatives of q and p with respect to time. The Hamilton's equations below describe how q and p change in time:

$$\begin{aligned} \dot{q} =& \nabla_p H(q, p) = M^{-1} p \\ \dot{p} =& -\nabla_q H(q, p) = -\nabla_q U(q), \end{aligned}$$

where ∇_p and ∇_q denote the gradient functions with respect to p and q, respectively. Since the equations are difficult to solve exactly, they are approximated by discretising time with a small step size ϵ. To propose a new point in the algorithm we make L such steps (called leapfrog steps) and then accept/reject the move based on the standard Metropolis ratio.

In Riemanian Manifold HMC the popular choice for the mass matrix M is the Fisher information $G_0(q)$, which enables adapting to the local structure of the target distribution, thus improving the efficiency of the algorithm.

Wormhole HMC in its vanilla version assumes that the modes $\mu_1, \dots \mu_M$ of the target distribution are known. We construct a network of wormholes connecting these modes – a wormhole is a small neighbourhood around a segment between μ_i and μ_j for $i \neq j$. The idea is to replace the metric G_0 in

RMHMC with G being a mixture of the original metric and the wormhole metric G_W:

$$G(q) = (1 - m(q))\, G_0(q) + m(q) G_W(q).$$

G_W is designed in such a way that it shortens the distance between modes, whereas m is a weighting function chosen so that G_W has more influence in the vicinity of the wormholes. This construction facilitates fast mixing between modes compared to the original RMHMC method.

[Lan et al., 2014] extend their method by allowing for including new modes (and consequently, new wormholes) on the fly at regeneration times. Since identifying regeneration times for the WHMC proposal described above is problematic, [Lan et al., 2014] mix it with an independence sampler and then follow a procedure specified in [Ahn et al., 2013].

Interestingly, [Lan et al., 2014] propose also a method of finding new modes, which is not common in the MCMC literature. The typical issue with optimisation-based methods is that if a mode-searching procedure is started from a number of starting points, usually the same mode is rediscovered multiple times, which is a waste of computational resources. To circumvent this problem, after a mode has been identified, its effect is removed from the target function by subtracting a Gaussian distribution providing the best fit to the mode. [Fok et al., 2017] propose some improvements to this concept.

The major drawback of methods of [Ahn et al., 2013] and [Lan et al., 2014] is that the adaptation is only allowed at regeneration times. Although this approach seems appealing from the point of view of theory, since no further proofs of convergence are needed, it does not work well in practice in high dimensions. The reason for this is that regenerations happen rarely in large dimensions, which makes the adaptive scheme prohibitively inefficient. Additionally, identifying regeneration times using the method of [Mykland et al., 1995] requires case-specific calculations which precludes any generic implementation of an algorithm based on regenerations. Moreover, the resulting identified regenerations are of orders of magnitude more infrequent than the true ones which are already rare.

1.5 Methods based on the Wang-Landau algorithm

The idea of the Wang-Landau algorithm ([Wang and Landau, 2001b], [Wang and Landau, 2001a]) is based on partitioning the state space $\mathcal{X}$ into $N + 1$ regions D_j for $j = 0, \ldots, N$ according to the value of the energy function, in the similar manner to the energy rings used in the equi-energy sampler (see (1.4)). The algorithm then draws samples from a biased target distribution $\bar{\pi}_n$ (at iteration n) in a way such that each of the regions is visited equally often. Ideally, the algorithm would target

$$\bar{\pi}(x) := \pi(x) \frac{1}{N+1} \sum_{j=0}^{N} \frac{\mathbb{I}_{D_j}(x)}{\psi(j)},$$

where $\psi(j) = \int_{D_j} \pi(x) dx$. Since these constants are unknown, the method involves learning them on the fly and hence modifying the target distribution while the algorithm runs. Namely, iteration n is a

Metropolis-Hasting step admitting $\bar{\pi}^{(n)}$ as the invariant distribution, where

$$\bar{\pi}^{(n)}(x) := \pi(x)\frac{1}{N+1}\sum_{j=0}^{N}\frac{\mathbb{I}_{D_j}(x)}{\theta_j^{(n)}}. \tag{1.8}$$

The value of $\theta_j^{(n)}$ is therefore increased each time the chain has visited D_j, because this makes it less likely for the chain to fall into D_j again. In order to obtain an unbiased estimator, at the end of the algorithm one must reweight the collected samples. Intuitively, the chain constructed this way spends a significant amount of time in regions of high energy (low probability), which facilitates moving between modes of the target distribution.

The algorithm is often described as aiming to obtain a "flat histogram" across the regions D_j (indeed, in the limit the proportion of samples in each region will be the same) and that is why D_j for $j = 0, \ldots, N$ are commonly referred to as "bins" (of the histogram).

Analogously to the equi-energy sampler, the performance of the Wang-Landau algorithm depends crucially on the choice of the energy levels $H_0, \ldots H_{N+1}$ and choosing them in a close to optimal way typically requires a lot of tuning. To address this issue and provide general improvement of the efficiency, [Bornn et al., 2013] proposed an adaptive version of the Wang-Landau method. Firstly, the new algorithm successively compares the values of $\theta_j^{(n)}$ between adjacent bins and splits bin D_j if its corresponding $\theta_j^{(n)}$ is relatively large. Secondly, instead of running one chain, [Bornn et al., 2013] propose using a number of interacting chains running in parallel. At iteration n they use the same proposal distribution for the Metropolis-Hastings steps and target the same invariant distribution. The values $\theta_j^{(n)}$ for $j = 0, \ldots, N$ are then updated based on the information about all chains. [Bornn et al., 2013] show that in some settings running K such chains for T iterations outperforms running one chain for $K \times T$ iterations. Finally, to speed up mixing of the chains, the covariance matrix of the proposal distribution is adapted aiming to achieve the optimal acceptance rate 0.234 of the Metropolis-Hastings steps (see [Roberts and Rosenthal, 2009]).

A drawback of the Wang-Landau algorithm in both its adaptive and non-adaptive version is that the chain spends a lot of time in the regions of low probability, hence, a relatively small number of samples comes from the regions of high probability. As a result, in the final reweighted sample the regions of high probability are not well represented.

[Zhou, 2011] proposed a multi-domain sampling technique, which combines the idea of the Wang-Landau algorithm with the optimisation-based approach discussed in Section 1.4. Assume that $\{\mu_1, \ldots, \mu_M\}$ is the set of modes of the target distributions π. The state space is now partitioned into M domains of attraction $A_1, \ldots A_M$ in the following way: x belongs to domain of attraction A_k associated with mode μ_k if a gradient ascent algorithm for π started from x converges to μ_k. Additionally, as in the original Wang-Landau algorithm, we introduce a sequence of energy levels $H_0, \ldots H_{N+1}$. Finally, we obtain a partition of X into $M \times (N+1)$ regions D_{kj} as follows;

$$D_{kj} := \{x : h(x) \in [H_j, H_{j+1}) \text{ and } x \in A_k\}.$$

The target distribution at iteration n is given by

$$\tilde{\pi}^{(n)} = \pi(x) \sum_{k=1}^{M} \sum_{j=0}^{N} \frac{\mathbb{I}_{D_j}(x)}{\exp\left(w_{kj}^{(n)}\right)}$$

for a matrix of weights $W^{(n)}$. Note that values $\exp\left(w_{kj}^{(n)}\right)$ play a similar role to $(N+1)\theta_j^{(n)}$ in (1.8). Analogously, the weights $w_{kj}^{(n)}$ are updated in a way such that the proportion of visits to each region D_{kj} converges to $\frac{1}{M \times (N+1)}$ as $n \to \infty$.

The algorithm makes steps of two types:

1. **Local moves**: (with probability p) perform a Random Walk Metropolis step.

2. **Global moves**: (with probability $1-p$) perform an independence sampler step proposing from a mixture of Gaussian distributions $\frac{1}{M}\sum_{k=1}^{M} N(\mu_k, \Sigma_k)$.

The matrices Σ_k are being adapted while the algorithm runs, namely, the matrix Σ_k is recalculated each time when a new sample has appeared in the domain of attraction A_k. This approach allows the proposal distribution for global moves to be well-adjusted to the local shape of each mode. Note that no adaptation scheme is proposed to optimise the step size for the local moves. It is worth emphasising that a major drawback of this method is the fact that at each iteration the new state must be assigned to its corresponding region D_{kj}, so in particular its domain of attraction must be identified. This is performed using the gradient ascent algorithm, which imposes a crippling computational burden on the whole algorithm. In fact, running an optimisation procedure is much more expensive than an MCMC step itself.

As we shall see later, there is some resemblance between the multi-domain sampling algorithm and the method that we propose in Chapter 2. Our algorithm also relies on local and global moves and tuning the proposal distribution is done separately for each mode. What is more, in both these algorithms adapting is allowed to happen beyond regeneration times (in contrast to [Ahn et al., 2013] and [Lan et al., 2014]). In our case, however, the issue of assigning each point to a mode is solved differently, thanks to which the computational burden discussed above is removed. What is more, the method we present is not based on the Wang-Landau algorithm so we do not need to introduce and recalculate weights of the regions.

1.6 Other methods

A relatively recent method of the pseudo-extended Markov chain Monte Carlo algorithm was introduced by [Nemeth et al., 2017] as an extension of Hamiltonian Monte Carlo. Let us now introduce auxiliary variables $x_1, \ldots, x_N$ and a new target distribution $\tilde{\pi}^N(x_1, \ldots, x_N)$ on $\mathcal{X}^N$ given by

$$\tilde{\pi}^N(x_1, \ldots, x_N) := \frac{1}{N} \sum_{i=1}^{N} \pi(x_i) \prod_{j \neq i} q(x_j)$$

for some proposal q. The algorithm first runs the HMC sampler targeting $\tilde{\pi}^N$. The marginal distribution of x^N with respect to x_i is a weighted mixture $\frac{1}{N}\pi(x_i) + \frac{N-1}{N}q(x_i)$, therefore a post-hoc correction procedure is required in order to obtain samples from the original target. [Nemeth et al., 2017] prove that using a certain weighting scheme on the samples from $\tilde{\pi}^N$ gives samples from $\pi(x)$.

The intuition underlying this method is that introducing q increases the target density in the regions of low probability for π, therefore inducing faster mixing. The examples presented in [Nemeth et al., 2017] demonstrate that the performance of this method is comparable to tempering-based ones, without requiring that much tuning.

The list of algorithms designed to deal with multimodal distributions is much longer. Other interesting approaches include adaptive independence samplers (with a fixed number of components, e.g. [Luengo and Martino, 2013] or with a possibility of adding new components on the fly e.g. [Maire et al., 2016]), evolutionary MCMC, which may be viewed as a generalisation of the parallel tempering method ([Liang and Wong, 2000]), or combining MCMC with free energy biasing ([Chopin et al., 2012]).

Chapter 2

Adaptive MCMC for Multimodal Distributions and theoretical results

2.1 Adaptive MCMC for Multimodal Distributions

Let π be the target distribution defined on a state space $\mathcal{X}$. We introduce a collection of target distributions $\{\tilde{\pi}_\gamma\}_{\gamma \in \mathcal{Y}}$ on the augmented state space $\mathcal{X} \times \mathcal{I}$, where $\mathcal{I}$ is the set of modes $\{\mu_1, \dots, \mu_N\}$ of π. We defer the discussion about finding the modes to Chapter 3. For a fixed $\gamma \in \mathcal{Y}$, $\tilde{\pi}_\gamma$ is defined as follows

$$\tilde{\pi}_\gamma(x, i) := \pi(x) \frac{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)}{\sum_{j \in \mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}. \tag{2.1}$$

The sampling algorithm that we propose is summarised in Algorithm 1. The method relies on steps of two types, performed with probabilities $1 - \epsilon$ and ϵ, respectively.

- **Local Move**: Assuming the current state of the chain is (x, i) and that γ is the current parameter, a Random Walk Metropolis step is performed using the proposal distribution $R_{\gamma,L,i}$. This move preserves the mode.
- **Jump Move**: A new mode j is proposed with probability $a_{\gamma,j}$. Then a new point y is proposed using the distribution $R_{\gamma,J,k}$.

Using the standard Metropolis-Hastings formula for the acceptance probability we obtain the following expression for the local moves:

$$\alpha_{\gamma,L}\left((x,i) \to (y,i)\right) = \min\left[1, \frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right] = \min\left[1, \frac{\pi(y) Q_i(\mu_i, \Sigma_{\gamma,i})(y)}{\pi(x) Q_i(\mu_i, \Sigma_{\gamma,i})(x)} \frac{\sum_{j \in \mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}{\sum_{j \in \mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(y)}\right]. \tag{2.2}$$

Similarly, for the jump moves we get:

$$\begin{aligned}\alpha_{\gamma,J}\left((x,i)\to(y,k)\right) &= \min\left[1, \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)}\frac{a_{\gamma,i}R_{\gamma,J,i}(x)}{a_{\gamma,k}R_{\gamma,J,k}(y)}\right] \\ &= \min\left[1, \frac{\pi(y)w_{\gamma,k}Q_k(\mu_k,\Sigma_{\gamma,k})(y)}{\pi(x)w_{\gamma,i}Q_i(\mu_i,\Sigma_{\gamma,i})(x)}\frac{\sum_{j\in\mathcal{I}}w_{\gamma,j}Q_j(\mu_j,\Sigma_{\gamma,j})(x)}{\sum_{j\in\mathcal{I}}w_{\gamma,j}Q_j(\mu_j,\Sigma_{\gamma,j})(y)}\frac{a_{\gamma,i}R_{\gamma,J,i}(x)}{a_{\gamma,k}R_{\gamma,J,k}(y)}\right].\end{aligned} \tag{2.3}$$

It is assumed further that the proposal distributions for jumps $R_{\gamma,J,i}$, the proposal distributions for local moves $R_{\gamma,L,i}$ and the "augmented target" distributions $Q_i(\mu_i,\Sigma_{\gamma,i})$ follow either the normal (light-tailed) or the t distribution (heavy-tailed) with ν degrees of freedom. Alternatively, the proposal distributions for jumps may follow the normal truncated to the ellipsoid

$$E_{\gamma,i} = \{x\in\mathcal{X} : (x-\mu_i)^T\Sigma_{\gamma,i}^{-1}(x-\mu_i)\leq\delta\} \tag{2.4}$$

with δ chosen so that $E_{\gamma,i}$ covers a pre-specified percentage of the probability mass of the original distribution. It is worth pointing out that we treat the three families of distributions listed above only as an illustration and in fact, the ergodic results presented below hold for a wider range of families of distributions.

The distributions $Q_i(\mu_i,\Sigma_{\gamma,i})$ and $R_{\gamma,J,i}$ are centred at μ_i, whereas the local proposal distributions $R_{\gamma,L,i}$ are centred at the current state x. To summarise, $Q_i(\mu_i,\Sigma_{\gamma,i})$ is either the normal distribution $N(\mu_i,\Sigma_{\gamma,i})$ or the t distribution $t(\mu_i,\Sigma_{\gamma,i},\nu)$ and analogously, $R_{\gamma,J,i}$ follows either $N(\mu_i,\Sigma_{\gamma,i})$ or $t(\mu_i,\Sigma_{\gamma,i},\nu)$ or the truncated normal distribution $TN(\mu_i,\Sigma_{\gamma,i},\delta)$. For the local proposal distributions the covariance matrices are additionally scaled by the factor $2.38^2/d$, which is commonly used as optimal for adaptive Metropolis algorithms (see, for example, [Roberts and Rosenthal, 2009]). Therefore, $R_{\gamma,L,i}$ follows either $N\left(x, 2.38^2/d\Sigma_{\gamma,i}\right)$ or $t\left(x, 2.38^2/d\Sigma_{\gamma,i},\nu\right)$.

Observe that the algorithm learns the covariance structure while it runs, so the covariance matrix of the proposal distribution is based on the (scaled) empirical covariance matrix of the samples obtained so far, which is a common trick used in the classical Adaptive MCMC. We apply this method separately to the covariance structure associated with each mode.

The weights $w_{\gamma,i}$ and the probabilities $a_{\gamma,i}$ of proposing mode i in a jump may also follow some adaptive rule. We discuss briefly a possible adaptive scheme for $a_{\gamma,i}$ and $w_{\gamma,i}$ in Chapter 3.

It is worth outlining that this special construction of the target distribution makes it unlikely for the algorithm to escape via local steps far away from the mode it is assigned to. Indeed, if a proposed point y is very distant from the current mode i, the acceptance probability becomes very small due to the expression $Q_i(\mu_i,\Sigma_{\gamma,i})(y)$ in the numerator in (2.2), which will typically be tiny in such case. This allows for retaining control over "from which mode a given state was drawn". The property of our algorithm mentioned above is crucial for its efficiency as it enables estimating matrices $\Sigma_{\gamma,i}$ based on samples that are indeed close to mode μ_i, which in turn improves both the within-mode and the between-mode mixing.

Observe that all the derivations above (and the theoretical results in the subsequent sections) hold if $\mathcal{I}$ is any finite set of values in $\mathcal{X}$, not necessarily the modes. This is important, because in practice we

Algorithm 1 Adaptive MCMC for Multimodal Distributions (iteration $n+1$)

```
Input: the list of modes $\{\mu_1, \dots \mu_N\}$, the parameter $\gamma_{n+1} = \{(\Sigma_1, \dots, \Sigma_N), (w_1, \dots, w_N), (a_1, \dots, a_N)\}$, empirical means and covariance matrices around each mode, a starting point $(x_n, i_n)$, constants $\alpha$, $\beta$ and $\epsilon$ and a constant for the optimal acceptance rate opt acc s.t. $0 < \alpha, \beta, \epsilon, \text{opt acc} < 1$, a positive integers $AC_1$ and $AC_2$, the dimension of the state space $d$.
Generate $u_1 \sim U[0,1]$.
if $u_1 > \epsilon$ then
    Local Move:
    Propose a new value $y \sim R_{L,i_n}(x_n, \cdot)$.
    Accept $y$ with probability $\min\left[1, \frac{\pi(y)Q_i(\mu_i,\Sigma_i)(y)}{\pi(x)Q_i(\mu_i,\Sigma_i)(x)} \frac{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(x)}{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(y)}\right]$.
    if $y$ accepted then
        $(x_{n+1}, i_{n+1}) = (y, i_n)$.
    else
        $(x_{n+1}, i_{n+1}) = (x_n, i_n)$.
    end if
else
    Jump Move:
    Propose a new mode $k \sim (a_1, \dots, a_N)$.
    Propose a new value $y \sim R_{J,k}(\cdot)$.
    Accept $(y,k)$ with probability $\min\left[1, \frac{\pi(y)w_k Q_k(\mu_k,\Sigma_k)(y)}{\pi(x)w_i Q_i(\mu_i,\Sigma_i)(x)} \frac{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(x)}{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(y)} \frac{a_i R_{J,i}(x)}{a_k R_{J,k}(y)}\right]$.
    if $(y,k)$ accepted then
        $(x_{n+1}, i_{n+1}) = (y, k)$.
    else
        $(x_{n+1}, i_{n+1}) = (x_n, i_n)$.
    end if
end if
if the number of samples in mode $i_n < AC_1$ then
    if Local Move then
        $\tilde{\Sigma}_{i_n} = \exp\Big((\text{number of samples in mode } i_n)^{-\alpha}(\text{the last acceptance probability - opt acc})\Big)\tilde{\Sigma}_{i_n}$.
        $\Sigma_{i_n} = \tilde{\Sigma}_{i_n} + \beta I_d$.
        Update $w_i$ and $a_i$ for $i = 1, \dots, N$.
    end if
else
    if the number of samples in mode $i_{n+1}$ is divisible by $AC_2$ then
        $\Sigma_{i_{n+1}} = \text{empirical covariance matrix in mode } i_{n+1} + \beta I_d$.
        Update $w_i$ and $a_i$ for $i = 1, \dots, N$.
    end if
end if
$\gamma_{n+2} = \{(\Sigma_1, \dots, \Sigma_N), (w_1, \dots, w_N), (a_1, \dots, a_N)\}$.
return The new sample $(x_{n+1}, i_{n+1})$, the parameter $\gamma_{n+2}$, the list of matrices $\Big(\tilde{\Sigma}_1, \dots, \tilde{\Sigma}_N\Big)$.
```

will not normally know the exact locations of the local maxima, but only their approximations.

A natural question to ask is whether this algorithm is ergodic. While the theory of ergodicity of Adaptive MCMC is quite well-understood, Algorithm 1 goes beyond this framework, since it keeps updating not only the transition kernels but also the target distribution. In fact, what would it even mean for such an algorithm to be ergodic if there is no fixed target distribution? In what follows we define formally ergodicity in this context and we discuss when, according to this definition, our algorithm is ergodic.

2.2 Auxiliary Variable Adaptive MCMC class

To facilitate formal treatment of Adaptive MCMC for Multimodal Distributions, we introduce a general class of Auxiliary Variable Adaptive MCMC algorithms, as follows.

Recall that $\pi(\cdot)$ is a fixed target probability density on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. For an auxiliary pair $(\Phi, \mathcal{B}(\Phi))$, define $\tilde{\mathcal{X}} := \mathcal{X} \times \Phi$, and for an index set $\mathcal{Y}$, consider a family of probability measures $\{\tilde{\pi}_\gamma(\cdot)\}_{\gamma \in \mathcal{Y}}$ on $(\tilde{\mathcal{X}}, \mathcal{B}(\tilde{\mathcal{X}}))$, such that

$$\tilde{\pi}_\gamma(B \times \Phi) = \pi(B) \qquad \text{for every} \quad B \in \mathcal{B}(\mathcal{X}) \quad \text{and} \quad \gamma \in \mathcal{Y}. \tag{2.5}$$

Let $\{\tilde{P}_\gamma\}_{\gamma \in \mathcal{Y}}$ be a collection of Markov chain transition kernels on $(\tilde{\mathcal{X}}, \mathcal{B}(\tilde{\mathcal{X}}))$, such that each $\tilde{P}_\gamma$ has $\tilde{\pi}_\gamma$ as its invariant distribution and is Harris ergodic, i.e. for all $\gamma \in \mathcal{Y}$,

$$(\tilde{\pi}_\gamma \tilde{P}_\gamma)(\cdot) = \tilde{\pi}_\gamma(\cdot), \quad \text{and} \quad \lim_{n \to \infty} \|\tilde{P}_\gamma^n(\tilde{x}, \cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} = 0, \quad \text{for all } \tilde{x} := (x, \phi) \in \tilde{\mathcal{X}}. \tag{2.6}$$

Here $\| \cdot - \cdot \|_{TV}$ is the usual total variation distance, defined for two probability measures μ and ν on a $\sigma-$algebra of sets $\mathcal{G}$ as $\|\mu(\cdot) - \nu(\cdot)\|_{TV} = \sup_{B \in \mathcal{G}} |\mu(B) - \nu(B)|$.

To define the dynamics of the Auxiliary Variable Adaptive MCMC sequence $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^\infty$, where Γ represents a random variable taking values in $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$, denote its filtration as

$$\mathcal{G}_n := \sigma\{\tilde{X}_0, \ldots, \tilde{X}_n, \Gamma_0, \ldots, \Gamma_n\}.$$

Now, the conditional distribution of Γ_{n+1} given $\mathcal{G}_n$ will be specified by the adaptive algorithm being used, such as Algorithm 1, while the dynamics of the $\tilde{X}$ coordinate follows

$$\mathbb{P}\big[\tilde{X}_{n+1} \in \tilde{B} | \tilde{X}_n = \tilde{x}, \Gamma_n = \gamma, \mathcal{G}_{n-1}\big] = \tilde{P}_\gamma(\tilde{x}, \tilde{B}), \qquad \tilde{x} \in \tilde{\mathcal{X}}, \gamma \in \mathcal{Y}, \tilde{B} \in \mathcal{B}(\tilde{\mathcal{X}}). \tag{2.7}$$

Note that depending on the adaptive update rule for Γ_n of the algorithm, the sequence $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^\infty$, defined above is not necessarily a Markov chain. By $\tilde{A}_n^{\mathcal{G}_t}(\cdot)$ denote the distribution of the $\tilde{\mathcal{X}}$-marginal of $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^\infty$ at time n, conditionally on the history up to time t, i.e.

$$\tilde{A}_n^{\mathcal{G}_t}(\tilde{B}) := \mathbb{P}\big[\tilde{X}_n \in \tilde{B} | \tilde{X}_0 = \tilde{x}_0, \ldots, \tilde{X}_t = \tilde{x}_t, \Gamma_0 = \gamma_0, \ldots, \Gamma_t = \gamma_t\big], \qquad \tilde{B} \in \mathcal{B}(\tilde{\mathcal{X}}),$$

and in particular for $t = 0$, we shall write

$$\tilde{A}_n^{(\tilde{x}, \gamma)}(\tilde{B}) := \tilde{A}_n^{\mathcal{G}_0}(\tilde{B}) = \mathbb{P}\big[\tilde{X}_n \in \tilde{B} | \tilde{X}_0 = \tilde{x}, \Gamma_0 = \gamma\big], \qquad \tilde{B} \in \mathcal{B}(\tilde{\mathcal{X}}).$$

By $A_n^{\mathcal{G}_t}(\cdot)$ and $A_n^{(\tilde{x}, \gamma)}(\cdot)$ denote the further marginalisation of $\tilde{A}_n^{\mathcal{G}_t}(\cdot)$ and $\tilde{A}_n^{(\tilde{x}, \gamma)}(\cdot)$, respectively, onto the space of interest $\mathcal{X}$, where the target measure $\pi(\cdot)$ lives, namely

$$A_n^{\mathcal{G}_t}(B) := \tilde{A}_n^{\mathcal{G}_t}(B \times \Phi), \quad \text{and} \quad A_n^{(\tilde{x}, \gamma)}(B) := \tilde{A}_n^{(\tilde{x}, \gamma)}(B \times \Phi), \qquad B \in \mathcal{B}(\mathcal{X}).$$

Finally, in order to define ergodicity of the Auxiliary Variable Adaptive MCMC, let

$$T_n(\tilde{x},\gamma) := \|A_n^{(\tilde{x},\gamma)}(\cdot) - \pi(\cdot)\|_{TV} = \sup_{B\in\mathcal{B}(\mathcal{X})} |A_n^{(\tilde{x},\gamma)}(B) - \pi(B)|.$$

Definition 1. We say that the Auxiliary Variable Adaptive MCMC algorithm generating $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^{\infty}$, is *ergodic*, if

$$\lim_{n\to\infty} T_n(\tilde{x},\gamma) = 0, \qquad \text{for all} \quad \tilde{x} \in \tilde{\mathcal{X}}, \gamma \in \mathcal{Y}.$$

As we shall see later in this chapter, Adaptive MCMC for Multimodal Distributions belongs to the class defined above. There exist other algorithms falling into this category, therefore the results presented in this chapter, in particular Theorems 2 and 3 may potentially be useful for analysing their ergodicity. An instance of another algorithm in this class is adaptive parallel tempering introduced by [Miasojedow et al., 2013]. Indeed, let us consider $\Phi := \mathcal{X}^N$ and $\tilde{X} := \mathcal{X} \times \mathcal{X}^N$ and

$$\tilde{\pi}_\gamma\left(x_N, (x_0, \ldots, x_{N-1})\right) := \prod_{i=0}^{N} \pi(x_i)^{\beta_{j,\gamma}}.$$

Then for any $B \in \mathcal{B}(\mathcal{X})$ we have

$$\tilde{\pi}_\gamma\left(B \times \Phi\right) = \int_B \pi(x_N)^{\beta_{N,\gamma}} dx_N \int_\Phi \prod_{i=0}^{N-1} \pi(x_i)^{\beta_{j,\gamma}} dx_0 \ldots dx_{N-1} = \int_B \pi(x_N)^{\beta_{N,\gamma}} dx_N = \pi(B),$$

where the last equality follows since $\beta_{N,\gamma} = 1$ for all γ. Additionally, the transition kernels used in adaptive parallel tempering $\{\tilde{P}_\gamma\}_{\gamma\in\mathcal{Y}}$ are defined in such a way that detailed balance holds.

Note that not all adaptive algorithms utilising auxiliary variables belong to this class. For example, the multi-domain sampling method by [Zhou, 2011] discussed in Chapter 1 introduces an auxiliary matrix of weights of the regions. However, condition 2.5 is not satisfied.

2.2.1 Uniform case

The following result is an analogue of Theorem 1 of [Roberts and Rosenthal, 2007], and establishes ergodicity of the class of Auxiliary Variable Adaptive MCMC in the uniform case, i.e. under the analogues of the usual conditions of simultaneous uniform ergodicity and diminishing adaptation.

Theorem 2. *Consider an Auxiliary Variable Adaptive MCMC algorithm on a state space $\tilde{\mathcal{X}} = \mathcal{X} \times \Phi$, following dynamics* (2.7) *with a family of transition kernels $\{\tilde{P}_\gamma\}_{\gamma\in\mathcal{Y}}$ satisfying* (2.6) *and* (2.5)*. If conditions (a) and (b) below are satisfied, then the algorithm is ergodic in the sense of Definition 1.*

(a) (Simultaneous uniform ergodicity.) For all $\varepsilon > 0$, there exists $N = N(\varepsilon) \in \mathbb{N}$ such that

$$\|\tilde{P}_\gamma^N(\tilde{x},\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \le \varepsilon, \qquad \textit{for all} \quad \tilde{x} \in \tilde{\mathcal{X}} \quad \textit{and} \quad \gamma \in \mathcal{Y};$$

(b) (Diminishing adaptation.) The random variable

$$D_n := \sup_{\tilde{x} \in \tilde{\mathcal{X}}} \|\tilde{P}_{\Gamma_{n+1}}(\tilde{x}, \cdot) - \tilde{P}_{\Gamma_n}(\tilde{x}, \cdot)\|_{TV}$$

converges to 0 *in probability.*

Proof. The proof will proceed by a coupling construction in the spirit of [Roberts and Rosenthal, 2007] (see also [Roberts and Rosenthal, 2013] for a more rigorous presentation), however the more complex setting of Auxiliary Variable MCMC necessitates a few preliminary steps.

On the same probability space define two additional sequences, namely $\{(\tilde{X}_n^{m(t^*)}, \Gamma_n^{m(t^*)})\}_{n=0}^{\infty}$ and $\{(\tilde{X}_n^{i(t^*,\kappa)}, \Gamma_n^{i(t^*,\kappa)})\}_{n=0}^{\infty}$ which are identical to $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^{\infty}$, before pre-specified time t^*, i.e.

$$(\tilde{X}_n^{m(t^*)}, \Gamma_n^{m(t^*)}) \;=\; (\tilde{X}_n^{i(t^*,\kappa)}, \Gamma_n^{i(t^*,\kappa)}) \;:=\; (\tilde{X}_n, \Gamma_n) \qquad \text{for } \; n \leq t^*.$$

After time t^*, the adaptive parameter $\Gamma^{m(t^*)}$ of $\{(\tilde{X}_n^{m(t^*)}, \Gamma_n^{m(t^*)})\}_{n=0}^{\infty}$ freezes and $\tilde{X}_n^{m(t^*)}$ becomes a Markov chain with the marginal dynamics defined for $n+1 > t^*$ as:

$$\Gamma_{n+1}^{m(t^*)} \;:=\; \Gamma_n^{m(t^*)} \left(= \Gamma_{t^*}^{m(t^*)}\right), \tag{2.8}$$

$$\mathbb{P}\left[\tilde{X}_{n+1}^{m(t^*)} \in \tilde{B} \mid \tilde{X}_n^{m(t^*)} = \tilde{x},\, \Gamma_{t^*}^{m(t^*)} = \gamma\right] \;=\; \tilde{P}_\gamma(\tilde{x}, \tilde{B}), \qquad \tilde{B} \in \mathcal{B}(\tilde{\mathcal{X}}). \tag{2.9}$$

The second sequence $\{(\tilde{X}_n^{i(t^*,\kappa)}, \Gamma_n^{i(t^*,\kappa)})\}_{n=0}^{\infty}$ interpolates between $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^{\infty}$ and $\{(\tilde{X}_n^{m(t^*)}, \Gamma_n^{m(t^*)})\}_{n=0}^{\infty}$. We first define the dynamics of $\Gamma^{i(t^*,\kappa)}$ for $n+1 > t^*$, as:

$$\Gamma_{n+1}^{i(t^*,\kappa)} \;:=\; \begin{cases} \Gamma_{n+1} & \text{if } \; \sup_{\tilde{x} \in \tilde{\mathcal{X}}} \|\tilde{P}_{\Gamma_{n+1}}(\tilde{x}, \cdot) - \tilde{P}_{\Gamma_n^{i(t^*,\kappa)}}(\tilde{x}, \cdot)\|_{TV} \leq \kappa, \\ \Gamma_n^{i(t^*,\kappa)} & \text{otherwise;} \end{cases} \tag{2.10}$$

and define an auxiliary stopping time that records decoupling of Γ_n and $\Gamma_n^{i(t^*,\kappa)}$ as

$$\tau_{i(t^*,\kappa)} \;:=\; \min\{n : \Gamma_n^{i(t^*,\kappa)} \neq \Gamma_n\}, \tag{2.11}$$

with the convention $\min \emptyset = \infty$. Now, define the dynamics of $\tilde{X}_n^{i(t^*,\kappa)}$ as:

$$\tilde{X}_n^{i(t^*,\kappa)} \;:=\; \tilde{X}_n \qquad \text{for } \; n \leq \tau_{i(t^*,\kappa)}, \quad \text{and} \tag{2.12}$$

$$\mathbb{P}\left[\tilde{X}_{n+1}^{i(t^*,\kappa)} \in \tilde{B} \mid \tilde{X}_n^{i(t^*,\kappa)} = \tilde{x}, \Gamma_n^{i(t^*,\kappa)} = \gamma\right] \;=\; \tilde{P}_\gamma(\tilde{x}, \tilde{B}) \quad \text{for } \; n+1 > \tau_{i(t^*,\kappa)} \; \text{ and } \; \tilde{B} \in \mathcal{B}(\tilde{\mathcal{X}}). \tag{2.13}$$

Define also the filtration $\{\mathcal{G}_n^*\}_{n=0}^{\infty}$ as an extension of $\{\mathcal{G}_n\}_{n=0}^{\infty}$ by:

$$\mathcal{G}_n^* := \sigma\left\{\{(\tilde{X}_k, \Gamma_k)\}_{k=0}^{n},\; \{(\tilde{X}_k^{m(t^*)}, \Gamma_k^{m(t^*)})\}_{k=0}^{n},\; \{(\tilde{X}_k^{i(t^*,\kappa)}, \Gamma_k^{i(t^*,\kappa)})\}_{k=0}^{n}\right\}. \tag{2.14}$$

Let $\tilde{A}_n^{m(t^*),(\tilde{x},\gamma)}(\cdot)$, $\tilde{A}_n^{m(t^*),\mathcal{G}_t^*}(\cdot)$, $A_n^{m(t^*),(\tilde{x},\gamma)}(\cdot)$, $A_n^{m(t^*),\mathcal{G}_t^*}(\cdot)$, and $\tilde{A}_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot)$, $\tilde{A}_n^{i(t^*,\kappa),\mathcal{G}_t^*}(\cdot)$, $A_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot)$, $A_n^{i(t^*,\kappa),\mathcal{G}_t^*}(\cdot)$, be analogues of $\tilde{A}_n^{(\tilde{x},\gamma)}(\cdot)$, $\tilde{A}_n^{\mathcal{G}_t}(\cdot)$, $A_n^{(\tilde{x},\gamma)}(\cdot)$, $A_n^{\mathcal{G}_t}(\cdot)$, where in the definitions of the above terms, instead of $\{(\tilde{X}_n, \Gamma_n)\}_{n=0}^{\infty}$, we use the sequences $\{(\tilde{X}_n^{m(t^*)}, \Gamma_n^{m(t^*)})\}_{n=0}^{\infty}$ and $\{(\tilde{X}_n^{i(t^*,\kappa)}, \Gamma_n^{i(t^*,\kappa)})\}_{n=0}^{\infty}$, respectively, and condition by the extended sigma algebras defined in (2.14).

By the triangle inequality, for any n, t^*, κ, and γ^* we have

$$\begin{aligned} T_n(\tilde{x},\gamma) = \|A_n^{(\tilde{x},\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \quad &\leq \quad \|A_n^{(\tilde{x},\gamma)}(\cdot) - A_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot)\|_{TV} \\ &\qquad + \|A_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot) - A_n^{m(t^*),(\tilde{x},\gamma)}(\cdot)\|_{TV} \\ &\qquad + \|A_n^{m(t^*),(\tilde{x},\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \\ &\leq \quad \|\tilde{A}_n^{(\tilde{x},\gamma)}(\cdot) - \tilde{A}_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot)\|_{TV} \\ &\qquad + \|\tilde{A}_n^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot) - \tilde{A}_n^{m(t^*),(\tilde{x},\gamma)}(\cdot)\|_{TV} \\ &\qquad + \|A_n^{m(t^*),(\tilde{x},\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \\ &=: \quad \diamondsuit_n^{(1)} + \diamondsuit_n^{(2)} + \diamondsuit_n^{(3)}, \end{aligned} \tag{2.15}$$

where in the second inequlity, for the first two terms, we have used that the total variation distances on $\tilde{\mathcal{X}}$ involve suprema over larger classes of sets than those on $\mathcal{X}$. In the sequel, when dealing with the third term, we shall also use the fact that for any $t < n \in \mathbb{N}$, $t^* \in \mathbb{N}$, $\tilde{x} \in \tilde{\mathcal{X}}$, $\gamma \in \mathcal{Y}$ and any $\gamma^* \in \mathcal{Y}$, we have

$$\|A_n^{m(t^*),\mathcal{G}_t^*}(\cdot) - \pi(\cdot)\|_{TV} \leq \|\tilde{A}_n^{m(t^*),\mathcal{G}_t^*}(\cdot) - \tilde{\pi}_{\gamma^*}(\cdot)\|_{TV}. \tag{2.16}$$

Now fix $\delta > 0$. To prove the claim, it is enough to construct a target time $K^* = K^*(\delta)$, s.t.

$$T_K(\tilde{x},\gamma) \leq \delta \qquad \text{for all} \quad K > K^*, \ \tilde{x} \in \tilde{\mathcal{X}} \ \text{ and } \ \gamma \in \mathcal{Y}. \tag{2.17}$$

We shall find such K^*. To this end let $\varepsilon = \delta/3$, and fix $N = N(\varepsilon)$ from assumption (a). Furthermore, let $H_n = \{D_n \geq \varepsilon/N^2\}$, note that by assumption (b) we have $\mathbb{P}(H_n) \leq \varepsilon/N$ for $n \geq n^* = n^*(\varepsilon)$, and for $K > n^* + N =: K^*$, define $E := \bigcap_{n=K-N}^{K-1} H_n^c$, satisfying $\mathbb{P}(E) \geq 1 - \varepsilon$.

First we deal with $\diamondsuit_K^{(1)}$ in (2.15).

Consider the process $\{(\tilde{X}_n^{i(t^*,\kappa)}, \Gamma_n^{i(t^*,\kappa)})\}_{n=0}^{\infty}$ with $t^* = K-N$ and $\kappa = \varepsilon/N^2$. Note that on E we have $\tilde{X}_n = \tilde{X}_n^{i(t^*,\kappa)}$, for $n = 0, 1, \ldots, K$, and therefore by the coupling inequality of [Roberts and Rosenthal, 2004],

$$\diamondsuit_K^{(1)} \quad \leq \quad \mathbb{P}(E^c) \ \leq \ \varepsilon. \tag{2.18}$$

To deal with $\diamondsuit_K^{(2)}$ in (2.15), first note that $\tilde{A}_n^{i(t^*,\kappa),\mathcal{G}_t^*}(B)$ and $\tilde{A}_n^{m(t^*),\mathcal{G}_t^*}(B)$ are $\mathcal{G}_t^*$ measurable random variables, and hence Jensen's inequality yields

$$\diamondsuit_K^{(2)} = \|\tilde{A}_K^{i(t^*,\kappa),(\tilde{x},\gamma)}(\cdot) - \tilde{A}_K^{m(t^*),(\tilde{x},\gamma)}(\cdot)\|_{TV} \ \leq \ \mathbb{E}\Big[\|\tilde{A}_K^{i(t^*,\kappa),\mathcal{G}_{t^*}^*}(\cdot) - \tilde{A}_K^{m(t^*),\mathcal{G}_{t^*}^*}(\cdot)\|_{TV}\Big]. \tag{2.19}$$

Recall equations (2.8) and (2.9) to note that taking $n = K$, $t^* = K - N$, gives

$$\tilde{A}_n^{m(t^*),\mathcal{G}_{t^*}^*}(\cdot) = \tilde{P}_{\Gamma_{t^*}}^{n-t^*}(\tilde{X}_{t^*}^{m(t^*)}, \cdot) = \tilde{P}_{\Gamma_{K-N}}^{N}(\tilde{X}_{K-N}, \cdot), \tag{2.20}$$

that is $\{\tilde{X}_n^{m(K-N)}\}_{n=K-N}^{K}$ is a Markov chain started from $\tilde{X}_{K-N}$ with dynamics $\tilde{P}_{\Gamma_{K-N}}$.

Now recall (2.10), (2.12), (2.13), i.e. the dynamics of $\{(\tilde X_n^{i(K-N,\kappa)}, \Gamma_n^{i(K-N,\kappa)})\}_{n=K-N}^{K}$, and in particular observe that (2.10) with $\kappa = \varepsilon/N^2$ yields

$$\sup_{\tilde x \in \tilde{\mathcal X}} \|\tilde P_{\Gamma_{K-N}}(\tilde x, \cdot) - \tilde P_{\Gamma_n^{i(K-N,\kappa)}}(\tilde x, \cdot)\|_{TV} \leq N\kappa = \frac{\varepsilon}{N}, \quad \text{for } n = K-N, \dots, K-1. \tag{2.21}$$

Hence, for every $n = K-N, \dots, K-1$, if $\tilde X_n^{m(K-N)} = \tilde X_n^{i(K-N,\kappa)}$, by (2.21) and Proposition 3(g) of [Roberts and Rosenthal, 2004], there exists a coupling of $\tilde X_{n+1}^{m(K-N)}$ and $\tilde X_{n+1}^{i(K-N,\kappa)}$, such that $\mathbb P\big[\tilde X_{n+1}^{m(K-N)} = \tilde X_{n+1}^{i(K-N,\kappa)}\big] \geq 1 - \varepsilon/N$. Reiterating this construction N times from $n = K-N$ to $n = K-1$ implies that there exists a coupling such that

$$\mathbb P\big[\tilde X_K^{m(K-N)} = \tilde X_K^{i(K-N,\kappa)} \mid \tilde X_{K-N}^{m(K-N)} = \tilde X_{K-N}^{i(K-N,\kappa)}\big] \geq 1 - \varepsilon. \tag{2.22}$$

Hence by the coupling inequality

$$\|\tilde A_K^{i(K-N,\kappa),\mathcal G^*_{K-N}}(\cdot) - \tilde A_K^{m(K-N),\mathcal G^*_{K-N}}(\cdot)\|_{TV} \leq \varepsilon,$$

which together with (2.19) yields

$$\Diamond_K^{(2)} \leq \varepsilon. \tag{2.23}$$

To deal with $\Diamond_K^{(3)}$ in (2.15), we first recycle the conditioning as in (2.19), then apply (2.16) inside the expectation, and then use the dynamics of $\{\tilde X_n^{m(K-N)}\}_{n=K-N}^{K}$ in (2.20), together with the choice of $N = N(\varepsilon)$ based on assumption (a) as follows.

$$\begin{aligned}
\Diamond_K^{(3)} &= \|A_K^{m(K-N),(\tilde x,\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \\
&\leq \mathbb E\Big[\|A_K^{m(K-N),\mathcal G^*_{K-N}}(\cdot) - \pi(\cdot)\|_{TV}\Big] \\
&\leq \mathbb E\Big[\|\tilde A_K^{m(K-N),\mathcal G^*_{K-N}}(\cdot) - \tilde\pi_{\Gamma_{K-N}}(\cdot)\|_{TV}\Big] \\
&= \mathbb E\Big[\|\tilde P^N_{\Gamma_{K-N}}(\tilde X_{K-N}, \cdot) - \tilde\pi_{\Gamma_{K-N}}(\cdot)\|_{TV}\Big] \\
&\leq \mathbb E[\varepsilon] = \varepsilon.
\end{aligned} \tag{2.24}$$

Putting together (2.15) with (2.18), (2.23) and (2.24), yields

$$\|A_K^{(\tilde x,\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \leq \Diamond_K^{(1)} + \Diamond_K^{(2)} + \Diamond_K^{(3)} \leq 3\varepsilon = \delta, \tag{2.25}$$

as required. □

2.2.2 Non-uniform case

In fact assumption a) of Theorem 2 can be relaxed, namely, it is enough to assume that $\|\tilde P^N_{\Gamma_{K-N}}(\tilde X_{K-N}, \cdot) - \tilde\pi_{\Gamma_{K-N}}(\cdot)\|_{TV} \leq \varepsilon$ holds for large N with probability close to 1. To introduce this concept formally, let $M_\varepsilon(\tilde x, \gamma)$ be defined as

$$M_\varepsilon(\tilde x, \gamma) = \inf\{k \geq 1 : \|\tilde P^k_\gamma(\tilde x, \cdot) - \tilde\pi_\gamma(\cdot)\|_{TV} \leq \varepsilon\}.$$

The following result is analogous to Theorem 2 of [Roberts and Rosenthal, 2007].

Theorem 3. *Consider an Auxiliary Variable Adaptive MCMC algorithm, like in Theorem 2. Replace condition a) of this theorem with the following assumption.*

(a) (Containment.) For all $\varepsilon > 0$ and all $\tilde{\delta} > 0$, there exists $N = N(\varepsilon, \tilde{\delta})$ such that

$$\mathbb{P}\left(M_\varepsilon(\tilde{X}_n, \Gamma_n) > N | \tilde{X}_0 = \tilde{x}, \Gamma_0 = \gamma\right) \leq \tilde{\delta}$$

for all $n \in \mathbb{N}$.

This condition (together with diminishing adaptation) implies ergodicity of the algorithm in the sense of Definition 1.

Proof. Fix $\delta > 0$. Similarly to the proof of Theorem 2, we aim to construct a target time $K^* = K^*(\delta)$ satisfying (2.17). Let $\varepsilon = \delta/4$ and choose $N = N(\varepsilon, \varepsilon)$ such that condition a) is satisfied for $\tilde{\delta} := \varepsilon$. The reasoning used in the proof of Theorem 2 remains valid until (2.24), where we cannot bound $\|\tilde{P}^N_{\Gamma_{K-N}}(\tilde{X}_{K-N}, \cdot) - \tilde{\pi}_{\Gamma_{K-N}}(\cdot)\|_{TV}$ by ε. Let $G := \{M_\varepsilon(\tilde{X}_{K-N}, \Gamma_{K-N}) \leq N\}$, then $\mathbb{P}(G) \geq 1 - \varepsilon$. Therefore, for (2.24) we get

$$\begin{aligned} & \mathbb{E}\left[\|\tilde{P}^N_{\Gamma_{K-N}}(\tilde{X}_{K-N}, \cdot) - \tilde{\pi}_{\Gamma_{K-N}}(\cdot)\|_{TV}\right] \\ = \ & \mathbb{E}\left[\|\tilde{P}^N_{\Gamma_{K-N}}(\tilde{X}_{K-N}, \cdot) - \tilde{\pi}_{\Gamma_{K-N}}(\cdot)\|_{TV}\mathbb{I}_G\right] + \mathbb{E}\left[\|\tilde{P}^N_{\Gamma_{K-N}}(\tilde{X}_{K-N}, \cdot) - \tilde{\pi}_{\Gamma_{K-N}}(\cdot)\|_{TV}\mathbb{I}_{G'}\right] \\ \leq \ & \varepsilon \cdot \mathbb{P}(G) + 1 \cdot \mathbb{P}(G') \leq 2\varepsilon, \end{aligned} \tag{2.26}$$

where in the first inequality we have used that on G

$$\|\tilde{P}^N_{\Gamma_{K-N}}(\tilde{X}_{K-N}, \cdot) - \tilde{\pi}_{\Gamma_{K-N}}(\cdot)\|_{TV} \leq \varepsilon$$

and that the total variation is a function bounded by 1. The second inequality follows from the fact that $\mathbb{P}(G') \leq \varepsilon$. We put together (2.15), (2.18), (2.23) and (2.26) to obtain

$$\|A_K^{(\tilde{x},\gamma)}(\cdot) - \pi(\cdot)\|_{TV} \leq \diamondsuit_K^{(1)} + \diamondsuit_K^{(2)} + \diamondsuit_K^{(3)} \leq 4\varepsilon = \delta,$$

which completes the proof. □

2.3 Overview of the assumptions

In order to prove ergodic results for Adaptive MCMC for Multimodal Distributions, we consider a slightly modified version of this algorithm, presented in Algorithm 2. The modifications are two-fold: firstly, we update the parameters only if the most recent sample (X_n, i_n) belongs to some fixed compact set A_{i_n} and secondly, we adapt them "increasingly rarely" (see [Chimisov et al., 2018]). The reasons for these modifications will become clearer in the sequel.

We assume herein that our target density π is super-exponential, i.e. (following the definition introduced in [Jarner and Hansen, 2000]), it is positive and has continuous first derivatives such that

$$\lim_{|x|\to\infty} \frac{x}{|x|} \cdot \nabla \log \pi(x) = -\infty. \tag{2.27}$$

Recall that we are working with the collection of distributions $\{\tilde{\pi}_\gamma(\cdot)\}_{\gamma\in\mathcal{Y}}$ on $\tilde{\mathcal{X}} := \mathcal{X} \times \mathcal{I}$, which corresponds to the notation introduced for the Auxiliary Variable Adaptive MCMC class in Section 2.2, for Φ equal to the space of modes $\mathcal{I}$. Moreover, for each $B \in \mathcal{B}(\mathcal{X})$ and $\gamma \in \mathcal{Y}$

$$\tilde{\pi}_\gamma(B \times \mathcal{I}) = \int_B \sum_{i\in\mathcal{I}} \pi(x) \frac{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)}{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)} dx = \int_B \pi(x) \cdot 1 = \pi(B).$$

Recall that we allow for an adaptation scheme for three lists of parameters: covariance matrices (used both for adapting the target distribution $\tilde{\pi}_\gamma$ and the proposal distributions), weights $w_{\gamma,i}$ and probabilities $a_{\gamma,i}$ of proposing mode i in a jump. Hence, formally $\mathcal{Y}$ refers to the product space of $\Sigma_{\gamma,1}, \ldots, \Sigma_{\gamma,N}$, $w_{\gamma,1}, \ldots, w_{\gamma,N}$ and $a_{\gamma,1}, \ldots, a_{\gamma,N}$ restricted by $\sum_{j\in\mathcal{I}} w_{\gamma,j} = 1$ and $\sum_{j\in\mathcal{I}} a_{\gamma,j} = 1$ for each $\gamma \in \mathcal{Y}$.

Let $\tilde{P}_{\gamma,L,i}$ denote the kernel associated with the local move around mode i and analogously, let $\tilde{P}_{\gamma,J,i}$ be the kernel of the jump to mode i. The full kernel $\tilde{P}_\gamma$ is thus defined as

$$\tilde{P}_\gamma\left((x,i),(dy,k)\right) := (1-\epsilon)\tilde{P}_{\gamma,L,i}\left((x,i),(dy,k)\right)\delta_{i=k} + \epsilon a_{\gamma,k}\tilde{P}_{\gamma,J,k}\left((x,i),(dy,k)\right).$$

It is easily checked that the acceptance probabilities (2.2) and (2.3) ensure that detailed balance holds for all the above kernels, admitting $\tilde{\pi}_\gamma$ as their invariant distributions. They also satisfy the Harris ergodicity condition. The above discussion shows that the algorithm indeed falls into the category of the Auxiliary Variable Adaptive MCMC, so Theorems 2 and 3 can be used to establish its ergodicity.

Recall that Algorithm 2 is constructed in such a way that all the covariance matrices $\Sigma_{\gamma,i}$ are, after exceeding a pre-specified number of samples, based on samples belonging to a compact set A_i. This implies that these matrices are bounded from above. Since we keep adding βI_d to the covariance matrix at each step, they are also bounded from below. Recall also that the covariance matrices for the local proposal distributions are scaled by a fixed factor $2.38^2/d$. Consequently, there exist positive constants m and M for which

$$mI_d \preceq \Sigma_{\gamma,i} \preceq MI_d \text{ and } mI_d \preceq 2.38^2/d\Sigma_{\gamma,i} \preceq MI_d \text{ for all } \gamma \in \mathcal{Y} \text{ and } i \in \mathcal{I}. \tag{2.28}$$

As for the adaptive scheme for $w_{\gamma,i}$ and $a_{\gamma,i}$, we only require that these values be bounded away from 0, i.e. there exist ϵ_a and ϵ_w such that

$$w_{\gamma,i} > \epsilon_w \quad \text{and} \quad a_{\gamma,i} > \epsilon_a \quad \text{for all } \gamma \in \mathcal{Y} \text{ and } i \in \mathcal{I}. \tag{2.29}$$

Therefore, the parameter space $\mathcal{Y}$ may be considered as compact and this will be a crucial assumption exploited in the subsequent sections.

Algorithm 2 Adaptive MCMC for Multimodal Distributions modified (iteration $n+1$)

```
Input: the list of modes $\{\mu_1, \dots \mu_N\}$, the parameter $\gamma_{n+1} = \{(\Sigma_1, \dots, \Sigma_N), (w_1, \dots, w_N), (a_1, \dots, a_N)\}$, empirical means and covariance matrices around each mode, the starting point $(x_n, i_n)$, constants $\alpha$, $\beta$ and $\epsilon$ and the constant for the optimal acceptance rate opt acc s.t. $0 < \alpha, \beta, \epsilon, \text{opt acc} < 1$, a positive integer $AC_1$, the dimension of the state space $d$, compact sets $A_1, \dots, A_N$, the sequence of lags $n_k$, a positive constant $\kappa^*$, a sequence of integers $N^*$.
Generate $u_1 \sim U[0,1]$.
if $u_1 > \epsilon$ then
    Local Move:
    Propose a new value $y \sim R_{L,i_n}(x_n, \cdot)$.
    Accept $y$ with probability $\min\left[1, \frac{\pi(y)Q_i(\mu_i,\Sigma_i)(y)}{\pi(x)Q_i(\mu_i,\Sigma_i)(x)} \frac{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(x)}{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(y)}\right]$.
    if $y$ accepted then
        $(x_{n+1}, i_{n+1}) = (y, i_n)$.
    else
        $(x_{n+1}, i_{n+1}) = (x_n, i_n)$.
    end if
else
    Jump Move:
    Propose a new mode $k \sim (a_1, \dots, a_N)$.
    Propose a new value $y \sim R_{J,k}(\cdot)$.
    Accept $(y,k)$ with probability $\min\left[1, \frac{\pi(y)w_k Q_k(\mu_k,\Sigma_k)(y)}{\pi(x)w_i Q_i(\mu_i,\Sigma_i)(x)} \frac{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(x)}{\sum_{j\in\mathcal{I}} w_j Q_j(\mu_j,\Sigma_j)(y)} \frac{a_i R_{J,i}(x)}{a_k R_{J,k}(y)}\right]$.
    if $(y,k)$ accepted then
        $(x_{n+1}, i_{n+1}) = (y, k)$.
    else
        $(x_{n+1}, i_{n+1}) = (x_n, i_n)$.
    end if
end if
if $X_{n+1} \in A_{i_{n+1}}$ then
    Update the empirical mean and covariance matrix in mode $i_{n+1}$.
end if
if the number of samples in mode $i_n < AC_1$ and $n+1$ is equal to $N_j^*$ for some $j$ then
    if Local Move and $X_{n+1} \in A_{i_{n+1}}$ then
        $\tilde{\Sigma}_{i_n} = \exp\Big((\text{number of samples in mode } i_n)^{-\alpha}(\text{the last acceptance probability - opt acc})\Big)\tilde{\Sigma}_{i_n}$.
        $\Sigma_{i_n} = \tilde{\Sigma}_{i_n} + \beta I_d$.
        Update $w_i$ and $a_i$ for $i = 1, \dots, N$.
        Sample $n_{j+1}^* = n_{j+1} + \text{Uniform}[0, \lfloor (j+1)^{\kappa^*} \rfloor]$.
        $N^* = N^* \cup N_{j+1}^*$ for $N_{j+1}^* = N_j^* + n_{j+1}^*$.
    end if
else
    if $n+1$ is equal to $N_j^*$ for some $j$ and $X_{n+1} \in A_{i_{n+1}}$ then
        $\Sigma_{i_{n+1}} =$ empirical covariance matrix in mode $i_{n+1} + \beta I_d$.
        Update $w_i$ and $a_i$ for $i = 1, \dots, N$.
        Sample $n_{j+1}^* = n_{j+1} + \text{Uniform}[0, \lfloor (j+1)^{\kappa^*} \rfloor]$.
        $N^* = N^* \cup N_{j+1}^*$ for $N_{j+1}^* = N_j^* + n_{j+1}^*$.
    end if
end if
$\gamma_{n+2} = \{(\Sigma_1, \dots, \Sigma_N), (w_1, \dots, w_N), (a_1, \dots, a_N)\}$.
return The new sample $(x_{n+1}, i_{n+1})$, the parameter $\gamma_{n+2}$, the list of matrices $\left(\tilde{\Sigma}_1, \dots, \tilde{\Sigma}_N\right)$, the sequence $N^*$.
```

2.4 Diminishing adaptation

As mentioned in Section 2.3, in Algorithm 2 we applied the idea of Adaptive increasingly rarely MCMC introduced in [Chimisov et al., 2018]. Originally, AirMCMC relies on updating the parameters of an MCMC algorithm only at pre-specified times N_j with and increasing sequence of lags n_k between them. N_j is therefore defined as

$$N_j = \sum_{k=1}^{j} n_k \quad \text{with } N_0 = 0 \text{ and } n_0 = 0.$$

[Chimisov et al., 2018] proposed using any scheme for the sequence $\{n_k\}_{k>1}$ that satisfies

$$c_2 k^{\kappa} \geq n_k \geq c_1 k^{\kappa}$$

for some positive c_1, c_2 and κ. This scheme, however, does not ensure that the random variable

$$D_n = \sup_{\tilde{x} \in \tilde{\mathcal{X}}} \|\tilde{P}_{\Gamma_{n+1}}(\tilde{x}, \cdot) - \tilde{P}_{\Gamma_n}(\tilde{x}, \cdot)\|_{TV}$$

converges to 0 in probability. To fix this issue, the following modification is introduced. The updates happen at times N_j^*, where

$$N_j^* = \sum_{k=1}^{j} n_k^* \quad \text{with } N_0^* = 0 \text{ and } n_0^* = 0.$$

and

$$n_k^* = n_k + \text{Uniform}[0, \lfloor k^{\kappa^*} \rfloor].$$

for some $\kappa^* \in (0, \kappa)$. Observe that if D_n can only be positive if $n+1 \in \{N_j^*\}_{j \geq 1}$. Furthermore, if $n + 1 > N_k$ then

$$\mathbb{P}(D_n > 0) \leq \frac{1}{\lfloor k^{\kappa^*} \rfloor},$$

so in particular D_n goes to 0 as n tends to infinity. As a result, diminishing adaptation is satisfied for Algorithm 2.

2.5 Uniformly ergodic case

In this section we consider the case when the proposal distributions for the jump moves $R_{\gamma,J,i}$ have heavier tails than the target distribution π for all $i \in \mathcal{I}$ and $\gamma \in \mathcal{Y}$, i.e.

$$\sup_{x \in \mathcal{X}} \sup_{\gamma \in \mathcal{Y}} \frac{\pi(x)}{R_{\gamma,J,i}(x)} < \infty \quad \text{for each } i \in \mathcal{I}. \tag{2.30}$$

We shall prove that under this assumption simultaneous uniform ergodicity is satisfied for Algorithm 2 and consequently, by Theorem 2, the algorithm is ergodic.

Theorem 4. *Consider Algorithm 2 and assume additionally that the relationship between the target distribution π and the proposal distributions $R_{\gamma,J,i}$ follows* (2.30). *Then Algorithm 2 is ergodic.*

Proof. Diminishing adaptation has been addressed in Section 2.4, so it is enough to prove that simultaneous uniform ergodicity holds. Note that assumption (2.30) implies that for some positive constant c_1

$$\frac{R_{\gamma,J,k}(y)}{\tilde{\pi}(y,k)} = \frac{R_{\gamma,J,k}(y)}{\pi(y)} \frac{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j,\Sigma_{\gamma,j})(y)}{w_{\gamma,k} Q_k(\mu_k,\Sigma_{\gamma,k})(y)} > \frac{R_{\gamma,J,k}(y)}{\pi(y)} > c_1 \tag{2.31}$$

for each $k \in \mathcal{I}$, $y \in \mathcal{X}$ and $\gamma \in \mathcal{Y}$. For any $(x,i) \in \mathcal{X} \times \mathcal{I}$, any set $\hat{C} \subset \mathcal{X}$ and any $k \in \mathcal{I}$ we can compute

$$\begin{aligned}
\int_{\hat{C}} \tilde{P}_\gamma\left((x,i),(dy,k)\right) \geq & \epsilon a_{\gamma,k} \int_{\hat{C}} \tilde{P}_{\gamma,J,k}\left((x,i),(dy,k)\right) \\
= & \epsilon a_{\gamma,k} \int_{\hat{C}} R_{\gamma,J,k}(y) \min\left[1, \frac{\tilde{\pi}_\gamma(y,k) a_{\gamma,i} R_{\gamma,J,i}(x)}{\tilde{\pi}_\gamma(x,i) a_{\gamma,k} R_{\gamma,J,k}(y)}\right] dy \\
\geq & \epsilon \int_{\hat{C}} \min\left[a_{\gamma,k} R_{\gamma,J,k}(y), \tilde{\pi}_\gamma(y,k) a_{\gamma,i} \frac{R_{\gamma,J,i}(x)}{\tilde{\pi}_\gamma(x,i)}\right] dy \\
\geq & \epsilon\epsilon_a \int_{\hat{C}} \min\left[c_1 \tilde{\pi}_\gamma(y,k), c_1 \tilde{\pi}_\gamma(y,k)\right] dy \\
= & \epsilon\epsilon_a c_1 \int_{\hat{C}} \tilde{\pi}_\gamma(y,k) dy.
\end{aligned}$$

Any set $C \subset \mathcal{X} \times \mathcal{I}$ may be decomposed as $C = \bigcup_{k\in\mathcal{I}} \hat{C}_k \times \{k\}$, therefore

$$\sum_{k\in\mathcal{I}} \int_{\hat{C}_k} \tilde{P}_\gamma\left((x,i),(dy,k)\right) \geq \sum_{k\in\mathcal{I}} \epsilon\epsilon_a c_1 \int_{\hat{C}_k} \tilde{\pi}_\gamma(y,k) dy = \epsilon\epsilon_a c_1 \tilde{\pi}_\gamma(C). \tag{2.32}$$

Since $\tilde{\pi}_\gamma$ is a probability measure on $\mathcal{X} \times I$ for each $\gamma \in \mathcal{Y}$ and (2.32) holds for all $(x,i) \in \mathcal{X} \times I$, by Theorem 8 of [Roberts and Rosenthal, 2004] we have

$$\|\tilde{P}_\gamma^n((x,i),\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \leq (1 - \epsilon\epsilon_a c_1)^n \quad \text{for all } (x,i) \in \mathcal{X} \times \mathcal{I} \text{ and } \gamma \in \mathcal{Y},$$

which completes the proof. □

If $R_{\gamma,J,k}$ follows the t distribution, condition (2.30) is implied for example by assumption (2.27) (about the super-exponential target pdf). This statement, however, depends crucially on $\mathcal{Y}$ being compact, in particular on condition (2.28). In fact, if there was no upper bound for the determinant of the covariance matrices used in the algorithm, we would have

$$\sup_{\gamma\in\mathcal{Y}} \frac{\pi(\mu_i)}{R_{\gamma,J,i}(\mu_i)} \propto \sup_{\gamma\in\mathcal{Y}} \pi(\mu_i) \det \Sigma_{\gamma,i}^{1/2} = \infty,$$

which would contradict (2.30).

2.6 Geometrically ergodic case

When the tails of the distribution π are heavier then the tails of the proposal distributions $R_{\gamma,J,i}$, simultaneous uniform ergodicity does not hold. However, it turns out that under some additional assumptions Algorithm 2 is still ergodic, as it satisfies the assumptions of Theorem 3. The following lemma will be essential for establishing ergodicity.

Lemma 5. *Assume that the following conditions are satisfied.*

a) For each $\gamma \in \mathcal{Y}$ $\|\tilde{P}^k_\gamma(\tilde{x},\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \to 0$ as $k \to \infty$.

b) There exists a set $C \subset \tilde{\mathcal{X}}$ and $\delta > 0$ such that for each γ we can find a probability measure ν_γ on C that satisfies

$$\tilde{P}_\gamma(\tilde{x},\cdot) \geq \delta\nu_\gamma(\cdot) \quad \textit{for all } \tilde{x} \in C.$$

c) There exists $\lambda < 1$, $b < \infty$ and a collection of functions $V_{\tilde{\pi}_\gamma} : \tilde{\mathcal{X}} \to [1,\infty)$ for $\gamma \in \mathcal{Y}$, such that

$$\sup_{\gamma\in\mathcal{Y}} \sup_{\tilde{x}\in C} V_{\tilde{\pi}_\gamma}(\tilde{x}) = v < \infty$$

and the following simultaneous drift condition is satisfied:

$$\tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x}) \leq \lambda V_{\tilde{\pi}_\gamma}(\tilde{x}) + b\mathbb{I}_C(\tilde{x}) \quad \textit{for all } \tilde{x} \in \tilde{\mathcal{X}} \textit{ and } \gamma \in \mathcal{Y}, \tag{2.33}$$

where for $\tilde{x} = (x,i)$

$$\tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x}) := \mathbb{E}\left(V_{\tilde{\pi}_\gamma}(\tilde{X}_{n+1}) \big| \tilde{X}_n = \tilde{x}, \Gamma_n = \gamma\right).$$

Moreover, $V_{\tilde{\pi}_\gamma}(\tilde{x})$ is bounded on compact sets as a function of $(\tilde{x},\gamma)$.

d) $\mathcal{Y}$ is compact in some topology.

e) There exists a compact set A such that if $X_n \notin A$, then $\Gamma_{n+1} = \Gamma_n$.

f) $\mathbb{E}V_{\tilde{\pi}_{\Gamma_0}}(\tilde{X}_0) < \infty$.

Then for all $\varepsilon > 0$ and all $\delta > 0$, there exists N such that

$$\mathbb{P}\left(M_\varepsilon(\tilde{X}_n,\Gamma_n) > N | \tilde{X}_0 = \tilde{x}_0, \Gamma_0 = \gamma_0\right) \leq \delta$$

where

$$M_\varepsilon(\tilde{x},\gamma) = \inf\{k \geq 1 : \|\tilde{P}^k_\gamma(\tilde{x},\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \leq \varepsilon\}.$$

Proof. By Proposition 3 in [Roberts and Rosenthal, 2007] and assumptions a) – c) of the lemma for each $\gamma \in \mathcal{Y}$ there exists $K < \infty$ and $\rho < 1$, depending only on λ, b, v and δ, such that

$$\|\tilde{P}^k_\gamma(\tilde{x},\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \leq KV_{\tilde{\pi}_\gamma}(\tilde{x})\rho^k. \tag{2.34}$$

We will prove that the sequence $V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n)$ is bounded in probability. By Lemma 3 in [Roberts and Rosenthal, 2007], it suffices to show that $\sup_n \mathbb{E}V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) < \infty$. Firstly, let us show that $\tilde{P}V_{\tilde{\pi}_\gamma}(\tilde{x})$ is bounded for $\gamma \in \mathcal{Y}$ and $\tilde{x} \in A$. Note that

$$\sup_{\gamma\in\mathcal{Y}} \sup_{\tilde{x}\in A} \tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x}) = \sup_{\gamma\in\mathcal{Y}} \sup_{\tilde{x}\in A} \left(\frac{\tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x})}{V_{\tilde{\pi}_\gamma}(\tilde{x})} V_{\tilde{\pi}_\gamma}(\tilde{x})\right) \leq \sup_{\gamma\in\mathcal{Y}} \sup_{\tilde{x}\in\mathcal{X}} \frac{\tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x})}{V_{\tilde{\pi}_\gamma}(\tilde{x})} \sup_{\gamma\in\mathcal{Y}} \sup_{\tilde{x}\in A} V_{\tilde{\pi}_\gamma}(\tilde{x}).$$

Since A and $\mathcal{Y}$ were assumed to be compact, $\sup_{\gamma\in\mathcal{Y}}\sup_{\tilde{x}\in A} V_{\tilde{\pi}_\gamma}(\tilde{x}) < \infty$. Additionally, the drift condition (2.33) yields

$$\sup_{\gamma\in\mathcal{Y}}\sup_{\tilde{x}\in\mathcal{X}} \frac{\tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x})}{V_{\tilde{\pi}_\gamma}(\tilde{x})} \leq \sup_{\gamma\in\mathcal{Y}}\sup_{\tilde{x}\in\mathcal{X}} \frac{\lambda V_{\tilde{\pi}_\gamma}(\tilde{x}) + b}{V_{\tilde{\pi}_\gamma}(\tilde{x})} \leq \lambda + b.$$

Therefore we can define $M := \sup_{\gamma\in\mathcal{Y}}\sup_{\tilde{x}\in A} \tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x}) < \infty$. It follows that

$$\begin{aligned} E\left(V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right) =& \mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right)\mathbb{I}_{\tilde{X}_n\in A} + \mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right)\mathbb{I}_{\tilde{X}_n\notin A} \\ =& \mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right)\mathbb{I}_{\tilde{X}_n\in A} + \mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right)\mathbb{I}_{\tilde{X}_n\notin A} \\ \leq& \sup_{\tilde{x}\in A} \tilde{P}_{\Gamma_{n+1}} V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{x}) + \mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right) \\ \leq& \sup_{\gamma\in\mathcal{Y}}\sup_{\tilde{x}\in A} \tilde{P}_\gamma V_{\tilde{\pi}_\gamma}(\tilde{x}) + \lambda V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) + b \leq M + \lambda V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) + b. \end{aligned} \tag{2.35}$$

By the law of total expectation,

$$\mathbb{E}V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1}) = \mathbb{E}\mathbb{E}\left(V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1})\middle|\tilde{X}_n,\Gamma_n\right),$$

which combined with (2.35) gives

$$\mathbb{E}V_{\tilde{\pi}_{\Gamma_{n+1}}}(\tilde{X}_{n+1}) \leq \lambda\mathbb{E}V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) + M + b.$$

This implies, using Lemma 2 in [Roberts and Rosenthal, 2007] that

$$\sup_n \mathbb{E}V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) \leq \max\left[\mathbb{E}V_{\tilde{\pi}_{\Gamma_0}}(X_0), \frac{M+b}{1-\lambda}\right].$$

Lemma 5 will now follow from combining the fact that the sequence $V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n)$ is bounded in probability with (2.34). Note that for any fixed ε and δ, there exists N such that

$$\begin{aligned} \mathbb{P}\left(M_\varepsilon(\tilde{X}_n,\Gamma_n) \leq N\right) &= \mathbb{P}\left(\|\tilde{P}^N_{\Gamma_n}(\tilde{X}_n,\cdot) - \tilde{\pi}_\gamma(\cdot)\|_{TV} \leq \varepsilon\right) \geq \mathbb{P}\left(KV_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n)\rho^N \leq \varepsilon\right) \\ &= \mathbb{P}\left(\log V_{\tilde{\pi}_{\Gamma_n}}(\tilde{X}_n) \leq \log\frac{\varepsilon}{K} - N\log\rho\right) \geq 1-\delta \end{aligned}$$

for all $n \in \mathbb{N}$. The last inequality holds since $\log\frac{\varepsilon}{K} - N\log\rho \to \infty$ as $N \to \infty$ and $\log V_{\Gamma_n}(\tilde{X}_n)$, as well as $V_{\Gamma_n}(\tilde{X}_n)$, is bounded in probability. □

We now proceed to present the main result of this section.

Theorem 6. *Consider Algorithm 2 and assume additionally that the following conditions are satisfied.*

a) The target distribution π is super-exponential, i.e. it is positive with continuous first derivatives and satisfies (2.27)*.*

b) Q_i is the multivariate t distribution for all $i \in \mathcal{I}$.

c) For some $s_J \in (0,1]$ the relationship between the target distribution and the proposal distributions for jumps is given by

$$\sup_{x\in\mathcal{X}} \sup_{\gamma\in\mathcal{Y}} \frac{R_{\gamma,J,i}(x)}{\pi(x)^{s_J}} < \infty \quad \textit{for each } i \in \mathcal{I}. \tag{2.36}$$

d) Let $r_{\gamma,i}(x)$ be the rejection set for the local moves, i.e. $r_{\gamma,i}(x) = \{y \in \mathcal{X} : \tilde{\pi}_\gamma(y,i) < \tilde{\pi}_\gamma(x,i)\}$. We assume that

$$\limsup_{|x|\to\infty} \sup_{\gamma\in\mathcal{Y}} \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)dy < 1 \quad \textit{for each } i \in \mathcal{I}. \tag{2.37}$$

Then Algorithm 2 is ergodic.

We will show that the assumptions of Theorem 3 are satisfied. Since diminishing adaptation was discussed in Section 2.4, it suffices to prove that the containment condition holds, which we will do using Lemma 5. The proof is organised as follows. In the first part we show that conditions a)–b) and d)–f) are satisfied – these will mainly follow directly from the construction of the algorithm. Secondly, we show that drift condition expressed in assumption c) of Lemma 5 is satisfied under the additional assumptions a), b) and c) of Theorem 6. We first focus on obtaining the appropriate result for the local kernels and subsequently we combine it with the result for jumps.

Assumptions a)–b) and d)–f) of Lemma 5

Proof. Assumptions a) and d) were discussed in Section 2.3. Assumption e) follows directly from the construction of the algorithm for $A := \bigcup_{i\in\mathcal{I}} A_i \times \{i\}$. Assumption f) holds trivially, since X_0 and Γ_0 are deterministic (chosen by the user of the algorithm). As for assumption b), let C be a set of the form

$$C = \hat{C} \times \mathcal{I} \quad \text{for some compact } \hat{C} \subset \mathcal{X}. \tag{2.38}$$

This condition will be imposed later in the proof of the theorem by taking $\hat{C} = B(\mathbf{0}, R)$ for some $R > 0$. We shall consider two cases. In the scenario when $R_{\gamma,J,i}$ follows the normal or the t distribution, we can take

$$\nu_\gamma := \text{Uniform}(C) \quad \text{for each } \gamma \in \mathcal{Y},$$

i.e. $\nu_\gamma(\hat{A} \times \{i\}) = \frac{1}{N}\frac{\mu^{\text{Leb}}(\hat{A})}{\mu^{\text{Leb}}(\hat{C})}$ for any $\hat{A} \subseteq \hat{C}$. To find an appropriate value of δ, fix an arbitrary set $\hat{B} \times \{k\}$ for $\hat{B} \subseteq \hat{C}$ and $k \in \mathcal{I}$ and observe that for any $\tilde{x} \in C$

$$\begin{aligned} \tilde{P}_\gamma(\tilde{x}, \hat{B} \times \{k\}) \geq& \epsilon a_{\gamma,k} \tilde{P}_{\gamma,J,k}(\tilde{x}, \hat{B} \times \{k\}) = \epsilon a_{\gamma,k} \int_{\hat{B}} R_{\gamma,J,k}(y) \min\left[1, \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} \frac{a_{\gamma,i} R_{\gamma,J,i}(x)}{a_{\gamma,k} R_{\gamma,J,k}(y)}\right] dy \\ \geq& \epsilon\epsilon_a \int_{\hat{B}} \min\left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x)\right] dy \\ \geq& \epsilon\epsilon_a \underbrace{\inf_{\gamma\in\mathcal{Y}} \min_{i,k\in\mathcal{I}} \inf_{x,y\in\hat{C}} \min\left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x)\right]}_{>0 \text{ for } R_{\gamma,J,k} \text{ and } R_{\gamma,J,i} \text{ normally or } t\text{-distributed}} \mu^{\text{Leb}}(\hat{B}). \end{aligned} \tag{2.39}$$

If we let

$$\delta := N\mu^{\text{Leb}}(\hat{C})\epsilon\epsilon_a \inf_{\gamma\in\mathcal{Y}} \min_{i,k\in\mathcal{I}} \inf_{x,y\in\hat{C}} \min\left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x)\right],$$

a decomposition similar to (2.32) shows that for any $B \subseteq C$

$$\tilde{P}_\gamma(\tilde{x}, B) \geq \delta\nu_\gamma(B).$$

If, however, $R_{\gamma,J,i}$ follows the truncated normal distribution, as described in Section 2.1, we can define ν_γ as the uniform distribution on

$$D := C \cap \left(\bigcup_{i \in \mathcal{I}} \bigcap_{\gamma \in \mathcal{Y}} E_{\gamma,i} \times \{i\} \right) = \bigcup_{i \in \mathcal{I}} \underbrace{\left(\hat{C} \cap \bigcap_{\gamma \in \mathcal{Y}} E_{\gamma,i} \right)}_{:=\hat{D}_i} \times \{i\}, \tag{2.40}$$

for $E_{\gamma,i}$ introduced in (2.4) (and 0 on $C \setminus D$), that is, $\nu_\gamma(\hat{A} \times \{i\}) = \frac{\mu^{\text{Leb}}(\hat{A} \cap \hat{D}_i)}{\sum_{i \in \mathcal{I}} \mu^{\text{Leb}}(\hat{D}_i)}$ for any $\hat{A} \subseteq \hat{C}$. Observe that since $\mathcal{Y}$ is compact, $\bigcap_{\gamma \in \mathcal{Y}} E_{\gamma,i}$ has a positive Lebesgue measure for each $i \in \mathcal{I}$. The radius R of $\hat{C} = B(\mathbf{0}, R)$ may be taken large enough so that the Lebesgue measure of $\hat{D}_i$ is also positive for each $i \in \mathcal{I}$, the measure ν_γ will be then well-defined. Calculations similar to (2.39) show that for any $\hat{B} \subseteq \hat{C}$

$$\begin{aligned}
\tilde{P}_\gamma(\tilde{x}, \hat{B} \times \{k\}) \geq & \epsilon\epsilon_a \int_{\hat{B}} \min \left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x) \right] dy \\
\geq & \epsilon\epsilon_a \int_{\hat{B} \cap \hat{D}_k} \min \left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x) \right] dy \\
\geq & \epsilon\epsilon_a \inf_{\gamma \in \mathcal{Y}} \min_{i,k \in \mathcal{I}} \inf_{x,y \in \hat{D}_k} \min \left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x) \right] \mu^{\text{Leb}}(\hat{B} \cap \hat{D}_k).
\end{aligned}$$

Setting

$$\delta := \left(\sum_{i \in \mathcal{I}} \mu^{\text{Leb}} \left(\hat{D}_i \right) \right) \epsilon\epsilon_a \inf_{\gamma \in \mathcal{Y}} \min_{i,k \in \mathcal{I}} \inf_{x,y \in \hat{D}_k} \min \left[R_{\gamma,J,k}(y), \frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)} R_{\gamma,J,i}(x) \right]$$

will complete the proof. □

Assumption c) of Lemma 5 (local kernels)

Proof. Fix $s \in (0,1)$ and let $V_{\tilde{\pi}_\gamma}(\tilde{x}) := c\tilde{\pi}_\gamma(\tilde{x})^{-s} = c\tilde{\pi}_\gamma((x,i))^{-s}$ for c such that $c\pi(x)^{-s} \geq 1$ (thus enforcing $V_{\tilde{\pi}_\gamma}(\tilde{x}) > 1$). This is a jointly continuous function of (x, γ) so it is bounded on compact sets in $\mathcal{X} \times \mathcal{Y}$ for each $i \in \mathcal{I}$. Therefore, it is also bounded on compact sets in $\tilde{\mathcal{X}} \times \mathcal{Y}$, in particular on C if C is of the form (2.38). The proof will be continued for $s = \frac{1}{2}$ but analogous reasoning would be valid for any $s \in (0,1)$.

We will prove that there exists $\lambda_L < 1$ such that for the local move kernels we have

$$\limsup_{|x| \to \infty} \sup_{\gamma \in \mathcal{Y}} \frac{\tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}((x,i))}{V_{\tilde{\pi}_\gamma}((x,i))} \leq \lambda_L \tag{2.41}$$

for all $i \in \mathcal{I}$. We will proceed in the spirit of the proof of Theorem 4.1 of [Jarner and Hansen, 2000]. Following the notation used there, let $C_{\pi(x)}(\delta)$ denote the radial δ-zone around $C_{\pi(x)}$, where $C_{\pi(x)}$ is the contour manifold corresponding to $\pi(x)$. Define also

$$\lambda_{L,i} := \limsup_{|x|\to\infty} \sup_{\gamma\in\mathcal{Y}} \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)dy. \tag{2.42}$$

By assumption (2.37) $\lambda_{L,i} < 1$.

Fix $i \in \mathcal{I}$ and $\epsilon > 0$. We will show that for sufficiently large x

$$\frac{\tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}\left((x,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)} \leq \lambda_{i,L} + 3\epsilon + \epsilon^{1/2}. \tag{2.43}$$

The idea of this proof is to split $\mathcal{X}$ into disjoint sets $\mathcal{X}\backslash B(x,K)$, $B(x,K)\cap C_{\pi(x)}(\delta)$ and $B(x,K)\backslash C_{\pi(x)}(\delta)$ and show that for any x with a sufficiently large norm the integral representing acceptance, that is, of the function $R_{\gamma,L,i}(y)\min\left[1, \frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right]\frac{V_{\tilde{\pi}_\gamma}((y,i))}{V_{\tilde{\pi}_\gamma}((x,i))}$ on those sets is bounded from above by ϵ, ϵ and $\epsilon^{1/2}$, respectively. We fix the values of K and δ below. As for the rejection part, we use (2.42) to show that the corresponding integral is bounded by $\lambda_{L,i} + \epsilon$, for all x at a sufficient distance from $\mathbf{0}$. Putting all these upper bounds together, we obtain the required $\lambda_{i,L} + 3\epsilon + \epsilon^{1/2}$.

Firstly, observe that there exists K such that

$$\sup_{\gamma\in\mathcal{Y}} \int_{\mathcal{X}\backslash B(x,K)} R_{\gamma,L,i}(x,y)dy \leq \mathbb{P}(Z \in \mathcal{X} \setminus B(x,K)) \leq \epsilon, \tag{2.44}$$

where $Z \sim R_{\gamma,L,i}(x,\Sigma)$ for $\Sigma = MI_d$.

Furthermore, we know from the proof of Theorem 4.1 of [Jarner and Hansen, 2000] that under assumption (2.27) for any δ and K

$$\mu^{\text{Leb}}\left(B(x,K)\cap C_{\pi(x)}(\delta)\right) \leq \delta\left(\frac{|x|+K}{|x|-K}\right)^{d-1}\frac{\mu^{\text{Leb}}\left(B(x,3K)\right)}{K}. \tag{2.45}$$

Since $\lim_{x\to\infty}\left(\frac{|x|+K}{|x|-K}\right)^{d-1} = 1$, there exists $R_1 > 0$ such that for $|x| > R_1$

$$\left(\frac{|x|+K}{|x|-K}\right)^{d-1} < 1+\epsilon.$$

Now let us choose δ such that for $|x| > R_1$

$$\mu^{\text{Leb}}\left(C_{\pi(x)}(\delta)\cap B(x,K)\right) \leq \epsilon\frac{1}{\sup_{\gamma\in\mathcal{Y}}\sup_{y\in\mathcal{X}} R_{\gamma,L,i}(x,y)} \leq \epsilon\frac{1}{g_m(\mathbf{0})},$$

where $g_m(\mathbf{0})$ is the value at $\mathbf{0}$ of the pdf of $R_{\gamma,L,i}(\mathbf{0},\Sigma)$ with $\Sigma = mI_d$, and so

$$\sup_{\gamma\in\mathcal{Y}} \int_{C_{\pi(x)}(\delta)\cap B(x,K)} R_{\gamma,L,i}(x,y)dy \leq \epsilon. \tag{2.46}$$

Let $r(x) = \{y \in \mathcal{X} : \pi(y) < \pi(x)\}$ and $a(x) = \{y \in \mathcal{X} : \pi(y) \geq \pi(x)\}$. We now split $B(x,K)\setminus C_{\pi(x)}(\delta)$ into $(r(x)\cap B(x,K))\setminus C_{\pi(x)}(\delta)$ and $(a(x)\cap B(x,K))\setminus C_{\pi(x)}(\delta)$ and we estimate the value of

$\min\left[1, \frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right] \frac{V_{\tilde{\pi}_\gamma}((y,i))}{V_{\tilde{\pi}_\gamma}((x,i))}$ on each of those sets separately. Fix $\tilde{K}$ such that

$$\frac{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)} \leq \tilde{K} \quad \text{for all } x \in \mathcal{X} \text{ and } \gamma \in \mathcal{Y}. \tag{2.47}$$

This is possible by the assumption that Q_j follow the multivariate t distribution for $j \in \mathcal{I}$ (see Appendix A.2). Since π is super-exponential, there exists R_2 so large that for $|x| > R_2$:

1) If $y \in (r(x) \cap B(x,K)) \setminus C_{\pi(x)}(\delta)$, then $\frac{\pi(y)}{\pi(x)} \leq \frac{\epsilon}{\tilde{K}}$.

2) If $y \in (a(x) \cap B(x,K)) \setminus C_{\pi(x)}(\delta)$, then $\frac{\pi(x)}{\pi(y)} \leq \frac{\epsilon}{\tilde{K}}$.

In the first case we have (using (2.47)):

$$\begin{aligned}\frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)} =& \frac{\pi(y)}{\pi(x)} \frac{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)} \frac{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(y)}{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(y)} \\ \leq& \frac{\pi(y)}{\pi(x)} \frac{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)} \leq \tilde{K} \frac{\pi(y)}{\pi(x)} \leq \epsilon.\end{aligned}$$

Similarly for $y \in (a(x) \cap B(x,K)) \setminus C_{\pi(x)}(\delta)$ we would get $\frac{\tilde{\pi}_\gamma(x,i)}{\tilde{\pi}_\gamma(y,i)} \leq \epsilon$. Hence, on $B(x,K) \setminus C_{\pi(x)}(\delta)$ we have

$$\min\left[1, \frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right] \frac{V_{\tilde{\pi}_\gamma}((y,i))}{V_{\tilde{\pi}_\gamma}((x,i))} = \min\left[\frac{\tilde{\pi}_\gamma(x,i)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}, \frac{\tilde{\pi}_\gamma(y,i)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right] \leq \epsilon^{1/2}. \tag{2.48}$$

Furthermore, by assumption (2.42) we can choose R_3 such that for $|x| > R_3$

$$\sup_{\gamma\in\mathcal{Y}} \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)dy \leq \lambda_{L,i} + \epsilon. \tag{2.49}$$

Finally, for $|x| > \max\left[R_1, R_2, R_3\right]$ we obtain

$$\begin{aligned}
\frac{\tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}\left((x,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)} &= \int_{\mathcal{X}} R_{\gamma,L,i}(y) \min\left[1, \frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right] \frac{V_{\tilde{\pi}_\gamma}\left((y,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)} dy \\
&\quad + \int_{\mathcal{X}} R_{\gamma,L,i}(x,y) \left(1 - \min\left[1, \frac{\tilde{\pi}_\gamma\left(y,i\right)}{\tilde{\pi}_\gamma\left(x,i\right)}\right]\right) dy \\
&= \int_{\mathcal{X}} R_{\gamma,L,i}(x,y) \min\left[\frac{\tilde{\pi}_\gamma\left(x,i\right)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}, \frac{\tilde{\pi}_\gamma\left(y,i\right)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right] dy \\
&\quad + \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y) \left(1 - \min\left[1, \frac{\tilde{\pi}_\gamma\left(y,i\right)}{\tilde{\pi}_\gamma\left(x,i\right)}\right]\right) dy \\
&= \int_{\mathcal{X}\setminus B(x,K)} R_{\gamma,L,i}(x,y) \min\left[\frac{\tilde{\pi}_\gamma\left(x,i\right)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}, \frac{\tilde{\pi}_\gamma\left(y,i\right)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right] dy \\
&\quad + \int_{B(x,K)\cap C_{\pi(x)}(\delta)} R_{\gamma,L,i}(x,y) \min\left[\frac{\tilde{\pi}_\gamma\left(x,i\right)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}, \frac{\tilde{\pi}_\gamma\left(y,i\right)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right] dy \\
\text{(see (2.48))} &\qquad + \int_{B(x,K)\setminus C_{\pi(x)}(\delta)} R_{\gamma,L,i}(x,y) \min\left[\frac{\tilde{\pi}_\gamma\left(x,i\right)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}, \frac{\tilde{\pi}_\gamma\left(y,i\right)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right] dy \\
&\qquad\quad + \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y) \left(1 - \frac{\tilde{\pi}_\gamma\left(y,i\right)}{\tilde{\pi}_\gamma\left(x,i\right)}\right) dy \\
(\leq \epsilon \text{ by } (2.44)) &\leq \int_{\mathcal{X}\setminus B(x,K)} R_{\gamma,L,i}(x,y) dy \\
(\leq \epsilon \text{ by } (2.46)) &\quad + \int_{B(x,K)\cap C_{\pi(x)}(\delta)} R_{\gamma,L,i}(x,y) dy \\
(\leq \epsilon^{1/2}) &\qquad + \int_{B(x,K)\setminus C_{\pi(x)}(\delta)} R_{\gamma,L,i}(x,y) \epsilon^{1/2} dy \\
(\leq \lambda_{L,i} + \epsilon \text{ by } (2.49)) &\qquad\quad + \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y) dy \\
&\leq \lambda_{L,i} + 3\epsilon + \epsilon^{1/2},
\end{aligned}$$

which ends the proof of (2.43). Consequently, by setting $\lambda_L := \max_{i\in\mathcal{I}} \lambda_{i,L}$, we obtain (2.41). Observe that (2.41) implies that there exists $R_L > 0$ such that if $|x| > R_L$, then

$$\tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}\left((x,i)\right) \leq \lambda_L V_{\tilde{\pi}_\gamma}\left((x,i)\right).$$

For $|x| \leq R_L$ we have

$$\sup_{|x|<R_L} \sup_{\gamma\in\mathcal{Y}} \tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}\left((x,i)\right) \leq \sup_{|x|<R_L} \sup_{\gamma\in\mathcal{Y}} \frac{\tilde{P}_{\gamma,L,i} V_{\tilde{\pi}_\gamma}\left((x,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)} \sup_{|x|<R_L} \sup_{\gamma\in\mathcal{Y}} V_{\tilde{\pi}_\gamma}\left((x,i)\right).$$

Now analogously to $r_{\gamma,i}(x)$, let us define the acceptance region for $\tilde{\pi}_\gamma$ as

$$a_{\gamma,i}(x) = \{y \in \mathcal{X} : \tilde{\pi}_\gamma(y,i) \geq \tilde{\pi}_\gamma(x,i)\}. \tag{2.50}$$

Note that

$$\begin{aligned}
\frac{\tilde{P}_{\gamma,L,i}V_{\tilde{\pi}_\gamma}\left((x,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)} =& \int_{a_{\gamma,i}(x)} R_{\gamma,i}(x,y)\frac{V_{\tilde{\pi}_\gamma}\left((y,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)}dy \\
&+ \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)\frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\frac{V_{\tilde{\pi}_\gamma}\left((y,i)\right)}{V_{\tilde{\pi}_\gamma}\left((x,i)\right)}dy \\
&\quad + \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)\left(1-\frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}\right)dy \\
=& \int_{a_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)\frac{\tilde{\pi}_\gamma\left(x,i\right)^{1/2}}{\tilde{\pi}_\gamma(y,i)^{1/2}}dy \\
&+ \int_{r_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)\left(1-\frac{\tilde{\pi}_\gamma(y,i)}{\tilde{\pi}_\gamma(x,i)}+\frac{\tilde{\pi}_\gamma(y,i)^{1/2}}{\tilde{\pi}_\gamma(x,i)^{1/2}}\right)dy \\
\leq& 2\int_{\mathcal{X}} R_{\gamma,L,i}(x,y)dy = 2.
\end{aligned}$$

Besides

$$\sup_{|x|<R_L}\sup_{\gamma\in\mathcal{Y}} V_{\tilde{\pi}_\gamma}\left((x,i)\right) < \infty$$

as for each i the function $V_{\tilde{\pi}_\gamma}\left((x,i)\right)$ is jointly continuous with respect to x and γ. By setting

$$b_L := 2\sup_{|x|<R_L}\sup_{\gamma\in\mathcal{Y}} V_{\tilde{\pi}_\gamma}\left((x,i)\right)$$

we obtain

$$\tilde{P}_{\gamma,L,i}V_{\tilde{\pi}_\gamma}\left((x,i)\right) \leq \lambda_L V_{\tilde{\pi}_\gamma}\left((x,i)\right) + b_L \mathbb{I}_{B(\mathbf{0},R_L)\times\mathcal{I}} \tag{2.51}$$

for all $(x,i)\in\mathcal{X}\times\mathcal{I}$. □

Assumption c) of Lemma 5 (jump kernels)

Proof. Recall that for any $s\in(0,1)$ if $V_{\tilde{\pi}_\gamma}\left((x,i)\right) = c\tilde{\pi}_\gamma(x,i)^{-s}$, then (2.51) holds for some λ_L, b_L and R_L. Furthermore,

$$\begin{aligned}
&\int_{\mathcal{X}} \tilde{P}_{\gamma,J,k}\left((x,i),(dy,k)\right)V_{\tilde{\pi}_\gamma}\left((y,k)\right) \\
=& \int_{\mathcal{X}} R_{\gamma,J,k}(y)\min\left[1,\frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)}\frac{a_{\gamma,i}R_{\gamma,J,i}(x)}{a_{\gamma,k}R_{\gamma,J,k}(y)}\right]V_{\tilde{\pi}_\gamma}\left((y,k)\right)dy \\
&+\left(1-\int_{\mathcal{X}} R_{\gamma,J,k}(y)\min\left[1,\frac{\tilde{\pi}_\gamma(y,k)}{\tilde{\pi}_\gamma(x,i)}\frac{a_{\gamma,i}R_{\gamma,J,i}(x)}{a_{\gamma,k}R_{\gamma,J,k}(y)}\right]dy\right)V_{\tilde{\pi}_\gamma}\left((x,i)\right) \\
\leq& \int_{\mathcal{X}} R_{\gamma,J,k}(y)V_{\tilde{\pi}_\gamma}\left((y,k)\right)dy + V_{\tilde{\pi}_\gamma}\left((x,i)\right).
\end{aligned} \tag{2.52}$$

By assumption (2.36) there exists a constant c_2 such that $\frac{R_{\gamma,J,i}(x)}{\pi(x)^{s_J}} < c_2$ for each $x\in\mathcal{X}$, $i\in\mathcal{I}$ and $\gamma\in\mathcal{Y}$ and as a consequence,

$$\frac{R_{\gamma,J,i}(x)}{\tilde{\pi}_\gamma(x,i)^{s_J}} = \frac{R_{\gamma,J,i}(x)}{\pi(x)^{s_J}}\left(\frac{\sum_{j\in\mathcal{I}} w_{\gamma,j}Q_j(\mu_j,\Sigma_{\gamma,j})(x)}{w_{\gamma,i}Q_i(\mu_i,\Sigma_{\gamma,i})(x)}\right)^{s_J} \leq c_2\tilde{K}^{s_J},$$

where the last inequality follows from (2.47). Fix $s < s_J$ and observe that

$$\begin{aligned} b_J := \sup_{\gamma\in\mathcal{Y}} \max_{k\in\mathcal{I}} \int_{\mathcal{X}} R_{\gamma,J,k}(y) V_{\tilde{\pi}_\gamma}\left((y,k)\right) dy =& \sup_{\gamma\in\mathcal{Y}} \max_{k\in\mathcal{I}} \int_{\mathcal{X}} R_{\gamma,J,k}(y)\tilde{\pi}_\gamma(y,k)^{-s} dy \\ =& \sup_{\gamma\in\mathcal{Y}} \max_{k\in\mathcal{I}} \int_{\mathcal{X}} \frac{R_{\gamma,J,k}(y)}{\tilde{\pi}_\gamma(y,k)^{s_J}} \tilde{\pi}_\gamma(y,k)^{s_J-s} dy \\ \leq& \sup_{\gamma\in\mathcal{Y}} \max_{k\in\mathcal{I}} c_2 \tilde{K}^{s_J} \int_{\mathcal{X}} \tilde{\pi}_\gamma(y,k)^{s_J-s} dy \\ \leq& c_2 \tilde{K}^{s_J} \int_{\mathcal{X}} \pi(y)^{s_J-s} dy < \infty, \end{aligned} \tag{2.53}$$

where the last inequality follows from π being super-exponential and $s_J - s$ positive.

Now recall that

$$\begin{aligned} \tilde{P}_\gamma V_{\tilde{\pi}_\gamma}\left((x,i)\right) =& \sum_{k\in\mathcal{I}} \int_{\mathcal{X}} \tilde{P}_\gamma\left((x,i),(dy,k)\right) V_{\tilde{\pi}_\gamma}\left((y,k)\right) = (1-\epsilon) \int_{\mathcal{X}} \tilde{P}_{\gamma,L,i}\left((x,i),(dy,i)\right) V_{\tilde{\pi}_\gamma}\left((y,i)\right) \\ &+ \epsilon \sum_{k\in\mathcal{I}} a_k \int_{\mathcal{X}} \tilde{P}_{\gamma,J,k}\left((x,i),(dy,k)\right) V_{\tilde{\pi}_\gamma}\left((y,k)\right). \end{aligned} \tag{2.54}$$

Putting together (2.51), (2.52), (2.53) and (2.54) yields

$$\begin{aligned} \tilde{P}_\gamma V_{\tilde{\pi}_\gamma}\left((x,i)\right) \leq& (1-\epsilon)\lambda_L V_{\tilde{\pi}_\gamma}\left((x,i)\right) + (1-\epsilon) b_L \mathbb{I}_{B(\mathbf{0},R_L)\times\mathcal{I}} + \epsilon \sum_{k\in\mathcal{I}} a_{\gamma,k} b_J + \epsilon V_{\tilde{\pi}_\gamma}\left((x,i)\right) \\ =& \left((1-\epsilon)\lambda_L + \epsilon\right) V_{\tilde{\pi}_\gamma}\left((x,i)\right) + (1-\epsilon) b_L \mathbb{I}_{B(\mathbf{0},R_L)\times\mathcal{I}} + \epsilon b_J. \end{aligned}$$

To obtain the drift condition as given by (2.33), fix λ such that

$$((1-\epsilon)\lambda_L + \epsilon) < \lambda < 1.$$

For $|x|$ larger than some R_J we have

$$\epsilon b_J < \left(\lambda - ((1-\epsilon)\lambda_L + \epsilon)\right) c\pi(x)^{-s} < \left(\lambda - ((1-\epsilon)\lambda_L + \epsilon)\right) V_{\tilde{\pi}_\gamma}\left((x,i)\right).$$

Let $b := (1-\epsilon)b_L + \epsilon b_J$ and $R := \max[R_L, R_J]$ and $C := B(\mathbf{0}, R) \times I$. Drift condition (2.33) follows. Note that the set C defined as above has the form (2.38), as required. If $R_{\gamma,J,i}$ follow the truncated normal distribution, we may additionally need to enlarge R so that all the sets $\hat{D}_i$ defined in (2.40) have a positive Lebesgue measure □

If $R_{\gamma,J,i}$ is assumed to follow the truncated normal distribution, as described in Section 2.3, then condition c) of Theorem 6 is automatically satisfied for $s_J = 1$, for any π positive on $\mathcal{X}$. Again, the assumption of $\mathcal{Y}$ being compact is vital here, since this ensures that all the ellipsoids defined in (2.4) are contained within a compact set

$$E = \bigcup_{i \in \mathcal{I}} E_i \quad \text{where} \quad E_i = \{x \in \mathcal{X} : (x - \mu_i)^T M^{-1} (x - \mu_i) \leq \delta\}.$$

Then for each $\gamma \in \mathcal{Y}$ and each $i \in \mathcal{I}$ the value of the density of $R_{\gamma,J,i}$ is equal to 0 outside E and consequently

$$\sup_{x \in \mathcal{X}} \sup_{\gamma \in \mathcal{Y}} \frac{R_{\gamma,J,i}(x)}{\pi(x)} = \sup_{x \in E} \sup_{\gamma \in \mathcal{Y}} \frac{R_{\gamma,J,i}(x)}{\pi(x)} < \infty,$$

as required.

The following lemmas are useful in verifying assumption d) of Theorem 6.

Lemma 7. *Let $r(x) = \{y \in \mathcal{X} : \pi(y) < \pi(x)\}$ and $a(x) = \{y \in \mathcal{X} : \pi(y) \geq \pi(x)\}$. Consider Algorithm 2 together with the assumptions described in Section 2.3 and assume additionally that the following conditions are satisfied.*

a) The target distribution π is super-exponential, i.e. it is positive with continuous first derivatives and satisfies (2.27)*.*

b) Q_i is the multivariate t distribution for all $i \in \mathcal{I}$.

c) For some $\gamma^* \in \mathcal{Y}$

$$\limsup_{|x| \to \infty} \int_{r(x)} R_{\gamma^*,L,i}(x, y) dy < 1 \quad \textit{for each } i \in \mathcal{I}. \tag{2.55}$$

Then condition (2.37) *holds.*

Proof. Fix $i \in \mathcal{I}$ and let ϵ_L be such that for $|x|$ larger than some R_0

$$\int_{a(x)} R_{\gamma^*,L,i}(x, y) \geq \epsilon_L,$$

(such ϵ_L can be found due to assumption (2.55)). Hence, for K sufficiently large

$$\int_{a(x) \cap B(x,K)} R_{\gamma^*,L,i}(x, y) \geq \frac{\epsilon_L}{2},$$

which implies that for any $|x| > R_0$

$$\mu^{\text{Leb}}\left(a(x) \cap B(x, K)\right) \geq \frac{\epsilon_L}{2 \sup_{y \in B(x,K)} R_{\gamma^*,L,i}(x, y)} = \frac{\epsilon_L}{2 \sup_{y \in B(\mathbf{0},K)} R_{\gamma^*,L,i}(\mathbf{0}, y)}$$

and consequently

$$\begin{aligned} &\inf_{\gamma \in \mathcal{Y}} \int_{a(x) \cap B(x,K)} R_{\gamma,L,i}(x, y) dy \\ \geq &\mu^{\text{Leb}}\left(a(x) \cap B(x, K)\right) \inf_{\gamma \in \mathcal{Y}} \inf_{y \in B(x,K)} R_{\gamma,L,i}(x, y) \\ \geq &\frac{\epsilon_L \inf_{\gamma \in \mathcal{Y}} \inf_{y \in B(\mathbf{0},K)} R_{\gamma,L,i}(\mathbf{0}, y)}{2 \sup_{y \in B(\mathbf{0},K)} R_{\gamma^*,L,i}(\mathbf{0}, y)} =: \tilde{\epsilon}_L > 0. \end{aligned} \tag{2.56}$$

Recall that $R_{\gamma,L,i}$ follows either the normal or the t distribution, so $\tilde{\epsilon}_L$ is indeed positive.

Let the acceptance region $a_{\gamma,i}(x)$ be given by (2.50). We will show that for $|x|$ sufficiently large and for each $\gamma \in \mathcal{Y}$

$$\int_{a_{\gamma,i}(x)} R_{\gamma,L,i}(x,y)dy \geq \frac{\tilde{\epsilon}_L}{2},$$

which will prove the claim. We shall now repeat similar arguments to those used in the proof of Theorem 6, in the part for the local kernels. Firstly, we use formula (2.45) to conclude that for $|x|$ larger than some R_1 (which may depend on K) and for sufficiently small δ (which may depend on K, R_1 and $\tilde{\epsilon}_L$), we have

$$\sup_{\gamma\in\mathcal{Y}} \int_{C_{\pi(x)}(\delta)\cap B(x,K)} R_{\gamma,L,i}(x,y)dy \leq \frac{\tilde{\epsilon}_L}{2}. \tag{2.57}$$

We put (2.56) together with (2.57) to obtain

$$\inf_{\gamma\in\mathcal{Y}} \int_{(a(x)\cap B(x,K))\setminus C_{\pi(x)}(\delta)} R_{\gamma,L,i}\left(x,y\right)dy \geq \frac{\tilde{\epsilon}_L}{2}$$

for $|x| > \max\left[R_0, R_1\right]$. Now recall that for each δ there exists such R_2 that for $|x| > R_2$ if $y \in (a(x) \cap B(x,K)) \setminus C_{\pi(x)}(\delta)$ then $\frac{\pi(y)}{\pi(x)} \geq \tilde{K}$ for $\tilde{K}$ defined as in (2.47). Therefore in particular $y \in a_{\gamma,i}(x)$ for each $\gamma \in \mathcal{Y}$. Finally, for $|x| > \max\left[R_0, R_1, R_2\right]$ we have

$$\inf_{\gamma\in\mathcal{Y}} \int_{a_{\gamma,i}(x)} R_{\gamma,L,i}\left(x,y\right)dy \geq \inf_{\gamma\in\mathcal{Y}} \int_{(a(x)\cap B(x,K))\setminus C_{\pi(x)}(\delta)} R_{\gamma,L,i}\left(x,y\right)dy \geq \frac{\tilde{\epsilon}_L}{2},$$

which ends the proof. □

Lemma 8. *Assume that the following conditions hold.*

a) The target distribution π is super-exponential, i.e. it is positive with continuous first derivatives and satisfies (2.27)*.*

b) Q_i is the multivariate t distribution for all $i \in \mathcal{I}$.

c) The target distribution satisfies

$$\limsup_{|x|\to\infty} \frac{x}{|x|} \cdot \frac{\nabla\pi(x)}{|\nabla\pi(x)|} < 0. \tag{2.58}$$

Then condition d) of Theorem 6 is satisfied.

Proof. Fix any $\gamma \in \mathcal{Y}$ and $i \in \mathcal{I}$. Note that $R_{\gamma,L,i}(\mathbf{0},\cdot)$ is bounded away from 0 on some region around $\mathbf{0}$ and satisfies $R_{\gamma,L,i}(x,y) = R_{\gamma,L,i}(y,x)$. To prove the required result, we will use analogous arguments to those from the proof of Theorem 4.3 of [Jarner and Hansen, 2000]. Let $\epsilon > 0$ and R be such that for $|x| > R$

$$\frac{x}{|x|} \cdot \frac{\nabla\pi(x)}{|\nabla\pi(x)|} \leq -\epsilon.$$

Fix $K > 0$ and define the cone $W(x)$ as

$$W(x) := \left\{x - a\xi : 0 < a < K, \xi \in S^{k-1}, \left|\xi - \frac{x}{|x|}\right| \leq \frac{\epsilon}{2}\right\}.$$

[Jarner and Hansen, 2000] show that for x sufficiently large $W(x) \subset a(x)$. What is more,

$$\liminf_{|x|\to\infty} \int_{W(x)} R_{\gamma,L,i}(x,y)dy > 0 \quad \text{and so} \quad \liminf_{|x|\to\infty} \int_{a(x)} R_{\gamma,L,i}(x,y)dy > 0.$$

Hence, since i was chosen arbitrarily, assumption (2.55) is satisfied for $\gamma^* := \gamma$.

We would like to point out here that originally Theorem 4.3 of [Jarner and Hansen, 2000] was proved under a stronger assumption, that is, $R_{\gamma,L,i}(x,y) = R_{\gamma,L,i}(|x-y|)$. However, careful inspection of this proof shows that it is enough to assume that $R_{\gamma,L,i}(x,y) = R_{\gamma,L,i}(y,x)$. □

The following corollary shows that if Q_i is assumed to follow the t distribution and the proposal distributions for jumps are chosen appropriately (see assumption c) of Theorem 6), Algorithm 2 is successful at targeting mixtures of normal distributions.

Corollary 9. *Let the target distribution π be given by*

$$\pi(x) \propto w_1 \exp\left(-p_1(x)\right) + \ldots + w_n \exp\left(-p_n(x)\right),$$

where $w_i > 0$ and p_i is a polynomial of order ≥ 2 for each $i = 1, \ldots, n$. If additionally Q_i follows the multivariate t distribution for all $i \in \mathcal{I}$, the assumptions of Lemma 8 are satisfied.

Proof. We will again refer multiple times to [Jarner and Hansen, 2000]. Firstly, by Theorem 4.4 of the cited paper, if π_1 and π_2 are super-exponential and satisfy (2.58), then also $a_1\pi_1 + a_2\pi_2$ is super-exponential and satisfies (2.58) for positive a_1 and a_2. By Theorem 4.6 of the same paper, each density of the form $\pi(x) \propto \exp\left(-p(x)\right)$ is super-exponential and satisfies (2.58), if p is a polynomial of order ≥ 2. Therefore, the assumptions of Lemma 8 hold, as required. □

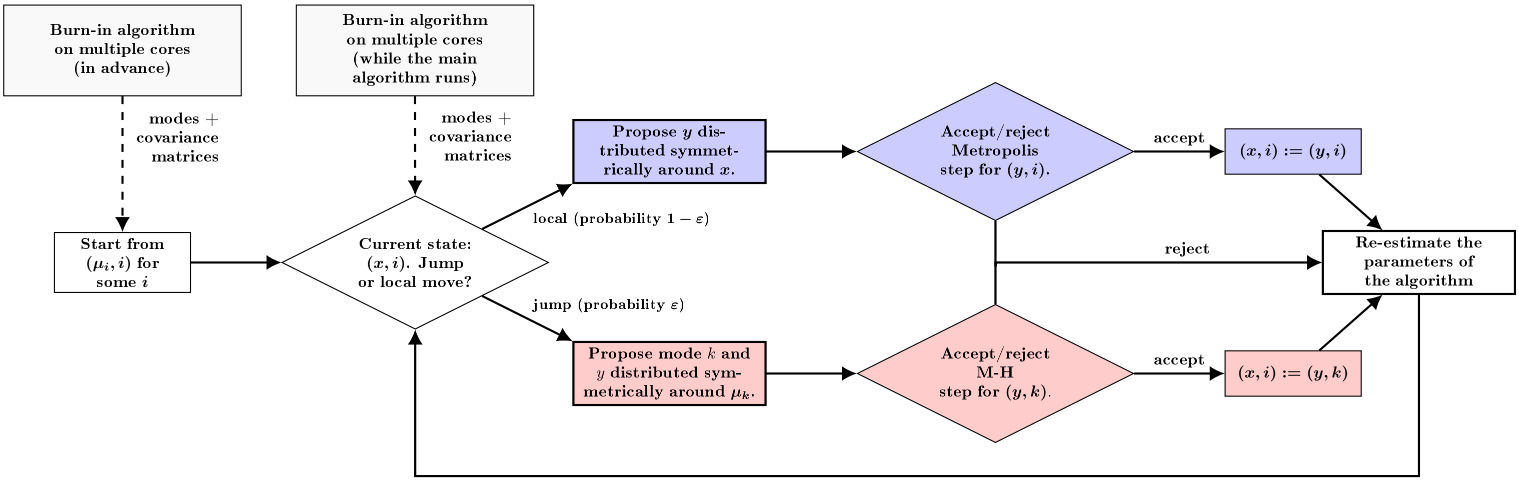
Chapter 3

Implementation and examples

In Chapter 2 we based our analysis on the assumption that the target distribution π has N known modes. Additionally, the fact that the algorithm is supposed to learn the covariance matrices around modes based on the samples collected in those modes assumes implicitly that the first jump to each of those modes will be accepted (since the updates of the parameters happen only at local moves). However, if the initial covariance matrices do not represent well the shapes of the modes, it may take a long time for the algorithm to accept a jump to some of the modes. Below we attempt to address the issues mentioned above and other problems related to the implementation of our algorithm, admitting at the same time that some of them are not fully resolved. In the second part of this chapter we demonstrate the performance of our algorithm on two toy examples, using our `R` package `MultiMCMC` with the implementation of our method.

3.1 Implementation details and the burn-in algorithm

1. **General overview of the burn-in algorithm.** The modes of the target distribution are rarely known in advance, so a method of finding them and passing their locations to the MCMC sampler must be incorporated into the algorithm. The way we deal with this issue is the following. First we sample a certain number of starting points in the state space. We then run an optimisation algorithm for π from each of those points in parallel on multiple cores. This procedure discovers at least a part of the modes of π and thanks to that we can construct our initial target distribution $\tilde{\pi}$. At this point the MCMC sampler (Algorithm 2) is initialised on one of the cores. In parallel, the remaining cores keep running the optimisation searches. They communicate with the MCMC sampler every certain number of iterations to allow for including the recently discovered modes into the target distribution $\tilde{\pi}$ (assuming the modes have not been added earlier). In addition, before passing a mode to the main MCMC sampler, we run an initial estimation of its corresponding covariance matrix. These estimates are treated as the initial values of the matrices $\Sigma_1, \ldots, \Sigma_N$ that are later adapted in the main algorithm. The whole procedure of finding the local maxima of the target distribution and estimating the covariance matrices forms the burn-in algorithm



Burn-in algorithm on multiple cores (in advance)
Burn-in algorithm on multiple cores (while the main algorithm runs)
modes + covariance matrices
modes + covariance matrices
Start from (μ_i, i) for some i
Current state: (x, i). Jump or local move?
local (probability $1-\varepsilon$)
jump (probability ε)
Propose y distributed symmetrically around x.
Propose mode k and y distributed symmetrically around μ_k.
Accept/reject Metropolis step for (y, i).
Accept/reject M-H step for (y, k).
accept
accept
reject
$(x, i) := (y, i)$
$(x, i) := (y, k)$
Re-estimate the parameters of the algorithm

Figure 3.1: Flowchart illustrating the communication between the main algorithm and the burn-in algorithm (running before the main algorithm is initialised and while it runs).

summarized by Algorithm 3. We elaborate on the details of different stages of this procedure and associated issues below, in points 2-5. The flowchart illustrating how the full algorithm works is shown in Figure 3.1.

The current version of our package, however, allows only for running the burn-in algorithm before the main one, there is no subsequent communication between the cores, therefore the examples shown later in this chapter are based on a pre-defined number of optimisation searches. It is worth pointing out that in the version where the burn-in algorithm continues to run in parallel to the main MCMC sampler, Theorem 6 and Theorem 4 of Chapter 2 hold provided the burn-in algorithm stops adding new modes at a finite time with probability one.

2. **Starting points for the optimization procedure.** We propose to sample the starting points for optimisation searches uniformly on a compact set which is a product of intervals provided by the user
$$[L_1, U_1] \times \ldots \times [L_d, U_d],$$
where d is the dimension of the state space. We would like to point out that even though the exact locations of modes are unlikely to be known, in applied problems it is not unrealistic for the user to be able to provide reasonable values for the end points of the intervals. For example, if the considered problem is the Bayesian regression, the range of plausible values for the parameters of interest is often known.

   Since the optimisation technique we use later is deterministic and the number of modes is finite (by assumption), the state space splits into a finite number of domains of attraction of the modes. If the volume of the overlap between $[L_1, U_1] \times \ldots \times [L_d, U_d]$ and the domain of attraction is positive for each mode, then asymptotically all modes will be found, as we will have at least one starting point in each domain. Obviously the efficiency of this method decreases exponentially with the dimension. In high dimensions some domains of attraction will tend to have a tiny volume compared to the volume of $[L_1, U_1] \times \ldots \times [L_d, U_d]$ and therefore the number of starting points

Algorithm 3 Burn-in algorithm

```
Input: the end points of the intervals $L_1, U_1, \dots, L_d, U_d$ such that the starting points for optimisation are sampled from $[L_1, U_1] \times \dots \times [L_d, U_d]$, the number of starting points $n$, a small positive value $q$, the number of rounds $K$ for the initial estimation of the covariance matrices, parameters of the main MCMC sampler for each round.
Generate $s_1, \dots, s_n$ uniformly from $[L_1, U_1] \times \dots \times [L_d, U_d]$.
Run BFGS from $s_1, \dots s_n$ to maximize $\pi(x)$. Denote the optimum points by $m_1, \dots, m_n$.
Set $\mu_1 := m_1$, $N := 1$.
for $i = 2, \dots, n$ do
    if $\min_{j \in 1,\dots,N} \|\mu_j - m_i\| < q$ then
        $k := \arg\min_{j \in 1,\dots,N} \|\mu_j - m_i\|$
        if $\pi(\mu_k) < \pi(m_i)$ then
            $\mu_k := m_i$
        end if
    else
        $\mu_{N+1} := m_i$
        $N := N + 1$
    end if
end for
for $i = 1, \dots, K$ do
    Run the main algorithm with $\epsilon = 0$ (no jumps) from $\mu_1, \dots \mu_N$ in parallel.
    Exchange the knowledge about the matrices $\Sigma_1, \dots, \Sigma_N$ between modes.
end for
Let $\Sigma_1, \dots \Sigma_N$ be the empirical covariance matrices based on the samples collected during last round.
return The modes $\mu_1, \dots, \mu_N$ and the covariance matrices $\Sigma_1, \dots, \Sigma_N$ that will be used as input for the main MCMC sampler.
```

required to identify all modes may become prohibitively large. On the other hand, this method, straightforward as it is, has an advantage of being fully parallelisable, so a user may take advantage of having access to a large number of cores.

This step remains the main bottleneck of our algorithm, so ideas how to set up the starting points more efficiently across basins of attraction will be explored in the future. So far we have looked into two different ideas of how this step could be improved. Firstly, we tried replacing the standard uniform distribution over $[L_1, U_1] \times \dots \times [L_d, U_d]$ with Quasi Monte Carlo sequences in the hope that more evenly spaced starting points will alleviate the issue. Unfortunately, this method did not prove to be particularly helpful on the examples that we have tested.

The other idea that we have explored is as follows. Having found some of the modes using our standard method, we take a number of Random Walk Metropolis steps targeting π^{-1} using the modes found so far as starting points. Namely, given the current state x we propose a new state y from a distribution $q(\cdot|x)$ and accept/reject this move using the probability of acceptance $\min\left[1, \frac{\pi(x)}{\pi(y)}\right]$. This idea is hence very close to the downhill moves employed in the RAM algorithm of [Tak et al., 2017]. We make K such steps with K drawn from the geometric distribution. The procedure is then repeated for several different parameters of the geometric distribution to provide more variation in the number of steps taken. The distribution q is chosen to be heavy-tailed, e.g. the multivariate t distribution. The maximization procedure then runs from the new starting points obtained in the way described above.

The rationale behind this method is that the above procedure pushes the starting points away from the high probability regions of π to the valleys between the modes or to the tails of π. Intuitively, the new starting points are then more likely to reach all the domains of attraction, including those of the most distant modes. The potential advantage of this method is that it reduces the burden of choosing the right end points of the intervals $[L_i, U_i]$ for $i = 1, \dots, d$.

This method turned out to be useful for some of the examples, that is, the number of modes identified using this approach was larger than when using the standard one. However, this method is highly parametric and it is not clear how the parameters (the covariance matrix and the number of degrees of freedom of the proposal distribution, the parameter of the geometric distribution) could be optimised. In our experiments shown later in this chapter we therefore stick to the straightforward approach.

3. **Mode-finding via optimization procedure.** We propose to use the BFGS algorithm, which is the most popular quasi-Newton method named after its inventors, Broyden, Fletcher, Goldfarb and Shanno. Recall that the quasi-Newton methods rely on the same principle as the original Newton's method, but the Hessian matrix used for constructing subsequent iterations of the algorithm is replaced with its approximation. The estimate of the Hessian matrix is then updated at each step, based on the knowledge it has gained during the last iteration. The reason for this is that recalculating the matrix completely at each iteration would be prohibitively expensive or even impossible. Quasi-Newton methods differ in the way the approximation of the Hessian is updated. The BFGS formula for the update gained its popularity since it has proven to be very efficient and robust. Additionally, it exhibits good self-correcting properties, i.e. if at some iteration the approximation of the Hessian is poor, the algorithm will tend to correct this estimate within the next few steps. More details may be found in [Nocedal and Wright, 2006].

   In particular, we are using the implementation of BFGS from the `optim` function in `R`. We found this method to be better-suited for the purpose of our algorithm than the Monte Carlo based ones, such as simulated annealing. Firstly, it provides a way of assessing if the convergence has been successful and secondly, it ensures more accuracy.

4. **"Clustering" the modes.** As discussed above, for finding the modes we use a numerical procedure (BFGS). Starting this algorithm from different points belonging to the same basin of attraction will take us to points which are very close to the true local maximum, but numerically different. Hence, the algorithm needs to have a built-in mechanism which decides whether two numerically different vectors in fact correspond to the same mode or not. If we instead decided to include all the vectors our optimisation algorithm converged to, we would effectively slow down the main algorithm, since the modes discovered more often would end up being proposed more frequently in jump moves. This is an immanent issue of all optimisation-based methods, however, it does not seem to be addressed in the MCMC literature.

   We deal with this issue in a heuristic way, presented in lines 4-15 of Algorithm 3. This simple method, based essentially on classifying two vectors as "equal" if their Euclidean distance is smaller than some pre-specified value q, proved to be successful in the scenarios we have tested.

5. **Initial estimation of the covariance matrices.** Recall that our main MCMC sampler requires some initial values of the matrices $\Sigma_1, \dots \Sigma_N$ which are then adapted while the algorithm runs. The adaptation, however, happens only at local moves. This means that if i is the mode from which the algorithm starts, before any update of matrix Σ_k is possible for $k \neq i$, a jump to mode k must be accepted. If the initial value of Σ_k does not correspond to the true local shape of mode k, the probability of accepting the jump will be typically small. Therefore, it is important to provide such initial values of $\Sigma_1, \dots, \Sigma_N$ that capture the true covariance structures reasonably well.

   In order to find these estimates, we run our main algorithm in parallel starting from each of the modes $\mu_1, \dots, \mu_N$ with $\epsilon = 0$, which is equivalent to making only local steps around each of the modes. This implies that we run N chains and each of them adapts only the matrix Σ_i corresponding to the mode μ_i which was its starting point. We make K rounds of this procedure (typically using $K = 4$) and after each round we change the target distribution $\tilde{\pi}$ by exchanging the knowledge about the adapted covariance matrices between cores. As outlined in Algorithm 3, the final covariance matrices passed to the main MCMC sampler are calculated based on the samples collected in the final round. For each round we need to specify the number of iterations and the constants AC_1 and AC_2. As for the number of iterations, we propose to use an increasing sequence, spending around a half of the overall number of iterations on the last round. The remaining parameters (e.g. the values of α and β) are set to be the same as in the main algorithm.

   Another approach would be to simply run an Adaptive MCMC algorithm in parallel starting from each mode – this would prevent the expensive communication between cores. However, the method presented earlier has a property of "discouraging" the samples to move far away from their corresponding modes and possibly visiting other ones, which enables obtaining more reliable estimates of the local covariance matrices.

   To compare these two approaches, we tested them on a modified version of the example of a mixture of 20 bivariate Gaussian distributions considered in [Kou et al., 2006] (with equal covariance matrices). Instead of the normal distribution, we used the bivariate t distribution with 7 degrees of freedom for each of the components, preserving the covariance matrices from the original example. It turned out that using Adaptive MCMC led to severe overestimation of some of the matrices (even if the initial values were much smaller than the true ones), whereas the approach of Algorithm 3 worked very well (for the same overall number of iterations), independently from the initial values.

6. **Facilitating the movement between modes.** The optimal within-mode mixing is obtained when the acceptance rate of local moves is around 0.234, as for the standard Metropolis algorithm ([Roberts and Rosenthal, 2009]). To optimise the mixing between modes, we would like the acceptance rate of jumps to be as close as 1 as possible.

   Assume that the target distribution $\tilde{\pi}$ at the current iteration is given by

$$\tilde{\pi}(x, i) = \pi(x) \frac{w_i Q_i(\mu_i, \Sigma_i)(x)}{\sum_{j \in \mathcal{I}} w_j Q_j(\mu_j, \Sigma_j)(x)}$$

(for simplicity of notation we do not use the subscript γ as it does not play a role here).

One idea for proposing a new state for a jump move from mode i to mode k is to sample a point from a spherical distribution (normally or t-distributed) centred at μ_k with a covariance matrix Σ_k, independently from the current state of the chain, as presented in Algorithms 1 and 2 in Chapter 2. Alternatively, given that the current state is x associated with mode i, we may propose a "corresponding" point y in mode k such that

$$(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) = (y-\mu_k)^T\Sigma_k^{-1}(y-\mu_k).$$

The required equality is satisfied for

$$y := \mu_k + \Lambda_k\Lambda_i^{-1}(x-\mu_i), \tag{3.1}$$

where

$$\Sigma_i = \Lambda_i\Lambda_i^T \quad \text{and} \quad \Sigma_k = \Lambda_k\Lambda_k^T.$$

The formula for the acceptance probability given by

$$\alpha_J\left((x,i)\to(y,k)\right) = \min\left[1, \frac{\tilde{\pi}(y,k)}{\tilde{\pi}(x,i)}\frac{a_i\sqrt{\det\Sigma_k}}{a_k\sqrt{\det\Sigma_i}}\right] \tag{3.2}$$

ensures that detailed balance holds for the deterministic proposal (3.1). We prove this fact in Appendix A.1. As we shall see later, this approach proves to be less sensitive to the inaccurate estimation of the matrices $\Sigma_1,\dots\Sigma_N$. The acceptance rate of jumps based on Gaussian or t-distributed proposal drops dramatically in high dimensions when obtaining an accurate estimate of the covariance matrix requires a large number of iterations. For deterministic jumps this decrease of efficiency is less severe.

Below we give an intuitive explanation why the deterministic jumps method described here may lead to higher acceptance rates for jumps. Recall that for any Metropolis-Hastings algorithm with a target distribution π defined on $\mathcal{X}$ and proposal q, the acceptance probability is given by $\min\left[1, \frac{\pi(y)}{\pi(x)}\frac{q(x|y)}{q(y|x)}\right]$. Moreover, if the chain is in equilibrium the expected value of the expression $\frac{\pi(y)}{\pi(x)}\frac{q(x|y)}{q(y|x)}$ is equal to 1. Indeed,

$$\int_{\mathcal{X}}\int_{\mathcal{X}}\frac{\pi(y)}{\pi(x)}\frac{q(x|y)}{q(y|x)}q(y|x)dy\pi(x)dx = \int_{\mathcal{X}}\pi(x)dx\int_{\mathcal{X}}q(y|x)dy = 1.$$

Our algorithm, both in the version with deterministic jumps and with independently proposed jumps, as described in Section 2.1, is a Metropolis-Hastings algorithm, so in both cases the expected value of the relevant expression is 1. Recall also that if the jump is proposed independently from the normal distribution the acceptance probability is given by

$$\min\left[1, \frac{\tilde{\pi}(y,k)}{\tilde{\pi}(x,i)}\frac{a_i\sqrt{\det\Sigma_k}}{a_k\sqrt{\det\Sigma_i}}\frac{\exp\left(-(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)/2\right)}{\exp\left(-(y-\mu_k)^T\Sigma_k^{-1}(y-\mu_k)/2\right)}\right].$$

Intuitively, the additional term $\frac{\exp\left(-(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)/2\right)}{\exp\left(-(y-\mu_k)^T\Sigma_k^{-1}(y-\mu_k)/2\right)}$ for $y \sim N(\mu_k, \Sigma_k)$ introduces additional variation is the considered expression. Therefore, informally speaking, in case of the independent proposal the value of this term is often much larger than 1 and often much smaller than one, whereas in case of the deterministic jumps it is more stable. Since the overall acceptance probability must be smaller than one, values much larger than 1 have the same influence as those slightly larger than 1. At the same time, the occasional values much smaller than 1 have an impact on decreasing the average acceptance rate.

As for the theoretical aspects of this method, neither of the two approaches of Chapter 2 (uniform and non-uniform ergodicity) applies directly to the deterministic jumps. However, one could use both the heavy-tailed jump proposal as well as the deterministic jumps (with positive probabilities). Then analogous reasoning to the proof to Theorem 4 demonstrates that the algorithm is ergodic.

7. **Possible extensions.** Instead of using Metropolis steps for the local moves, one could use Hamiltonian Monte Carlo, which could possibly speed up the within-mode mixing. Theorem 4 would then hold, as it does not require any assumptions about the local sampler. Other choices of local kernels are also possible and their design can follow the vast general purpose MCMC literature.

   Another possible direction would be adapting the weights w_i for $i = 1, \dots, N$ and the probabilities a_i of proposing a move to mode i so they reflect the true weights of each component. Our idea was to keep a_i equal to w_i and increase the weight w_i each time mode i has been visited (at the same time keeping these values separated from 0). In the version with the burn-in algorithm running alongside the main MCMC sampler, we could boost the weight of the recently added mode in order to learn its covariance matrix more quickly and make up for the time it was not known to the main sampler. The disadvantage of this approach is that if a particular mode is difficult to jump to (e.g. because of its peculiar shape), we would decrease its weight compared to other modes, which would make it even more unlikely to jump to this mode. Therefore, in our examples we are using the safer approach of keeping both w_i and a_i for $i = 1, \dots, N$ fixed throughout the run of the algorithm.

3.2 Examples

In this section we compare the performance of our algorithm (`MultiMCMC`) against adaptive parallel tempering (`APT`) based on empirical results for two different multimodal examples. Adaptive parallel tempering ([Miasojedow et al., 2013]) was chosen here as it is the refined version of the most commonly used MCMC method for multimodal distributions (parallel tempering). What is more, this algorithm, unlike some other methods based on regeneration times ([Ahn et al., 2013], [Lan et al., 2014]), has a generic implementation, where the user only needs to provide the target density function. To obtain the results presented below we used the code provided as supplementary material to the paper introducing adaptive parallel tempering. Our package `MultiMCMC` has three implemented methods, varying in the way the jumps between modes are performed. They are either proposed independently from the normal distribution (Gaussian jumps) or the multivariate t distribution (t-distributed jumps), as described in Section 2.1, or they follow the scheme presented in point 6 of Section 3.1 (deterministic jumps). The first example is a mixture of two Gaussians motivated by [Woodard et al., 2009a] and the second one is a mixture of fifteen multivariate t distributions and five banana-shaped ones. Below we describe the details of our experiments and discuss the conclusions.

3.2.1 Mixture of two Gaussian distributions with unequal covariance matrices

As mentioned in Chapter 1, [Woodard et al., 2009a] studied the following target density

$$\pi(x) = \frac{1}{2}N\left(-\underbrace{(1,\ldots,1)}_{d}, \sigma_1^2 I_d\right) + \frac{1}{2}N\left(\underbrace{(1,\ldots,1)}_{d}, \sigma_2^2 I_d\right), \tag{3.3}$$

for $\sigma_1 \neq \sigma_2$. In particular, they showed that the parallel tempering algorithm will tend to stay in the wider mode and, if started in the wider mode, may take a long time before getting to the more narrow one.

We looked at the results for the target distribution (3.3) in several different dimensions d for $\sigma_1^2 = 0.5\sqrt{d/100}$ and $\sigma_2^2 = \sqrt{d/100}$. The results for our method shown below are based on 500,000 iterations of the main algorithm, preceded by

- for dimensions smaller than 80: 1500 BFGS runs started from points sampled uniformly on $[-2,2]^d$ and 50,000 iterations (overall in 4 rounds) of the burn-in algorithm (per mode) for initial covariance matrix estimation;

- for dimensions 80 and 130: 4000 BFGS runs started from points sampled uniformly on $[-2,2]^d$ and 100,000 iterations (overall in 4 rounds) of the burn-in algorithm (per mode) for initial covariance matrix estimation.

For the adaptive parallel tempering algorithm we used 1,800,000 iterations and 5 temperatures. In order to base our analysis on the same sample size of 500,000 for the two methods, after the initial burn-in period of 300,000 we applied thinning, looking at every third iteration. Recall that one iteration of parallel tempering involves a Metropolis-Hastings update at each temperature level followed by a temperature swap, therefore in our case the computational cost amounts to $1,800,000 \times 6$ MCMC steps. Additionally, despite the name of the algorithm, it is not easy to split the computation between cores, since communication is needed before each temperature swap. Apart from this computational advantage that we gave to adaptive parallel tempering, expecting that the more narrow mode would be more difficult to detect, we set the starting point to $-\underbrace{(1,\ldots,1)}_{d} \in \mathbb{R}^d$. The exact parameters used in our experiments are summarised in Table 3.1. For dimensions $d = 5$ and $d = 20$ we ran 20 chains independently for each method. For each of the 20 runs of our algorithm we used the same burn-in algorithm for the three jumps methods.

		Example 1	Example 2
Main algorithm (MultiMCMC)			
	number of iterations	500,000	500,000 / 1,000,000
	α	0.7	0.7
	β	0.03	10^{-7}
	ϵ	0.1	0.1
	AC_1	100,000	10,000
	AC_2	1000	1000
	optimal acceptance rate	0.234	0.234
	distributions Q_i	t with 7 df	t with 7 df
	local proposal	Gaussian	t-distributed
	$(w_1, \ldots, w_N)$	(0.7, 0.3)	equal weights
	$(a_1, \ldots, a_N)$	equal weights	equal weights
	df of the proposal (if t-distributed)	7	7
Burn-in algorithm (MultiMCMC)			
	number of BFGS runs	1500/4000	10,000/30,000/–
	initial covariance matrices	$0.04I_d$	$0.1I_d$
	q	$0.07\sqrt{d}$	$0.07\sqrt{d}$
round 1	number of iterations	1000	500
	AC_1	1000	500
	AC_2	–	–
round 2	number of iterations	9000	2000/5000
	AC_1	8000/9000	1500/5000
	AC_2	100/–	100/–
round 3	number of iterations	15,000/40,000	2500/14,500
	AC_1	10,000/20,000	1500/10,000
	AC_2	500	100
round 4	number of iterations	25,000/50,000	5000/25,000
	AC_1	15,000/30,000	3000/10,000
	AC_2	1000	100/500
Adaptive parallel tempering			
	number of iterations	1,800,000	2,100,000
	number of temperatures	5	4
	burn-in period	300,000	600,000
	thinning	every third sample	every third sample
	initial covariance matrices	$0.1I_d$	$0.1I_d$

Table 3.1: Settings of the experiments for Example 1 (mixture of two Gaussian distributions) and Example 2 (mixture of multivariate t and banana-shaped distributions). In case of Example 1 the smaller numbers of iterations correspond to dimensions 5, 20 and 50 and the larger ones – to dimensions 80 and 130. In case of Example 2 the smaller numbers of iterations correspond to dimensions 4 and 10 and the larger ones to dimension 50 (except for the number of BFGS searches – there were 10,000 of them for $d = 4$ and 30,000 for $d = 10$; the modes were assumed to be known for $d = 50$).

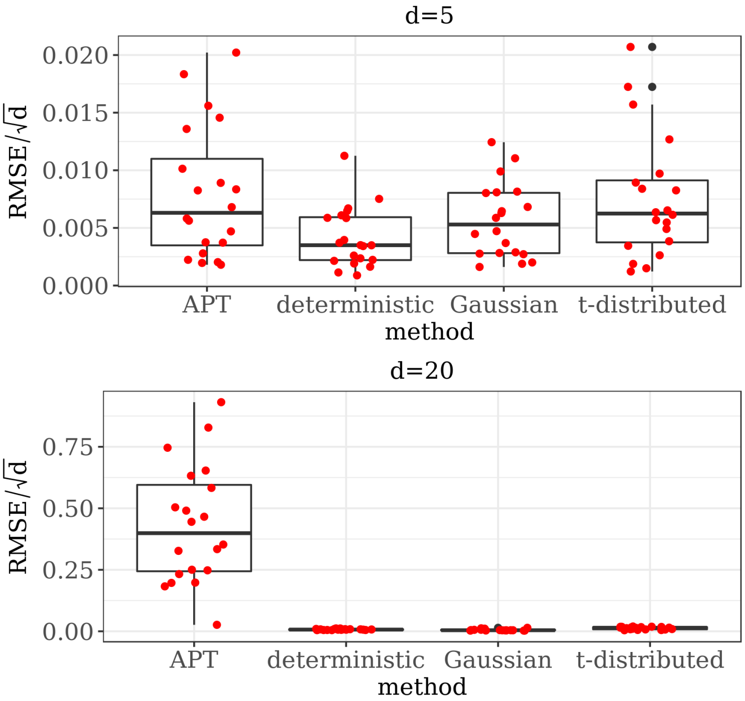


Figure 3.2: Boxplots of the values of RMSE/$\sqrt{d}$ for Example 1 across 20 runs of the experiment, dimensions 5 and 20. We compare the results for adaptive parallel tempering `APT` with three `MultiMCMC` methods: deterministic, Gaussian and t-distributed jumps.

To compare the performance of our method with adaptive parallel tempering, we computed the Root Mean Square Error divided by $\sqrt{d}$. That is, we calculated the Euclidean distance between the true expected value and the empirical one based on MCMC samples (divided by $\sqrt{d}$ to scale it appropriately for different dimensions). Figure 3.2 shows the boxplots of the value of RMSE/$\sqrt{d}$ for `APT` and `MultiMCMC` for three different methods of jumps. Figure 3.3 presents histograms of the means of all coordinates. We can conclude from these plots that whereas the methods give similar results in dimension 5, `MultiMCMC` in all its versions outperforms adaptive parallel tempering significantly in dimension 20, even with a much smaller computational budget.

	deterministic		Gaussian		t-distributed	
	Lowest	Highest	Lowest	Highest	Lowest	Highest
$d = 5$	0.64	0.96	0.60	0.82	0.62	0.84
$d = 20$	0.94	0.94	0.79	0.80	0.69	0.70
$d = 50$	0.91	0.93	0.50	0.51	0.43	0.44
$d = 80$	0.90	0.93	0.26	0.29	0.25	0.26
$d = 130$	0.76	0.87	0.00	0.01	0.02	0.02

Table 3.2: The lowest and the highest value (across 20 runs of the experiment) of the minimal acceptance rates of jump moves between any two modes for Example 1.

To analyse our method more systematically, we tested it on the same example in higher dimensions, again repeating the experiment 20 times. Figures 3.4 and 3.5 show analogous boxplots and histograms, respectively, for dimensions 50, 80, 130. It turns out that all three methods provide reliable results up to dimension 80, whereas the deterministic jumps and the t-distributed jumps work well even in

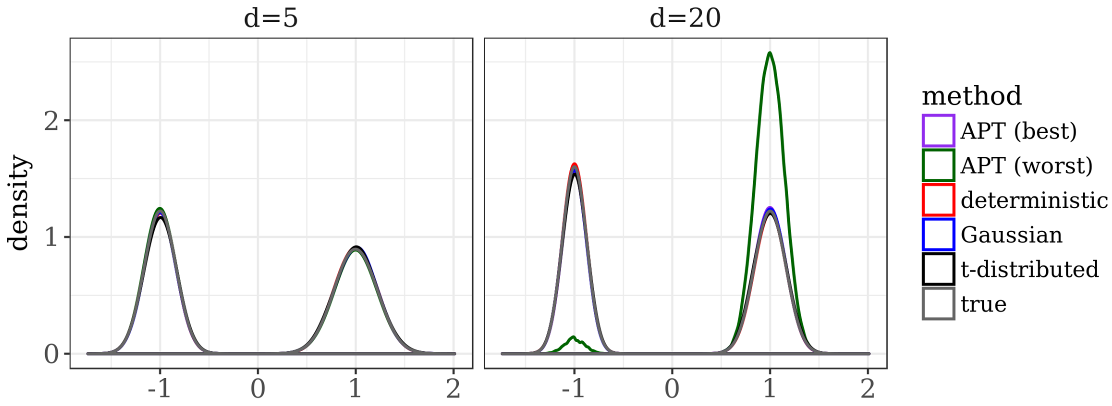


Figure 3.3: Density of the mean of all coordinates for Example 1, dimensions 5 and 20. We compare the results obtained for adaptive parallel tempering (`APT`) and the three methods (deterministic, Gaussian and t-distributed jumps) implemented in `MultiMCMC` with a sample generated from the true target distribution (`true`). `APT` (best) corresponds to the chain with the lowest value of RMSE/$\sqrt{d}$ across the 20 runs, `APT` (worst) – to the highest value. For each of the three `MultiMCMC` methods the chain with the highest value of RMSE/$\sqrt{d}$ is presented.

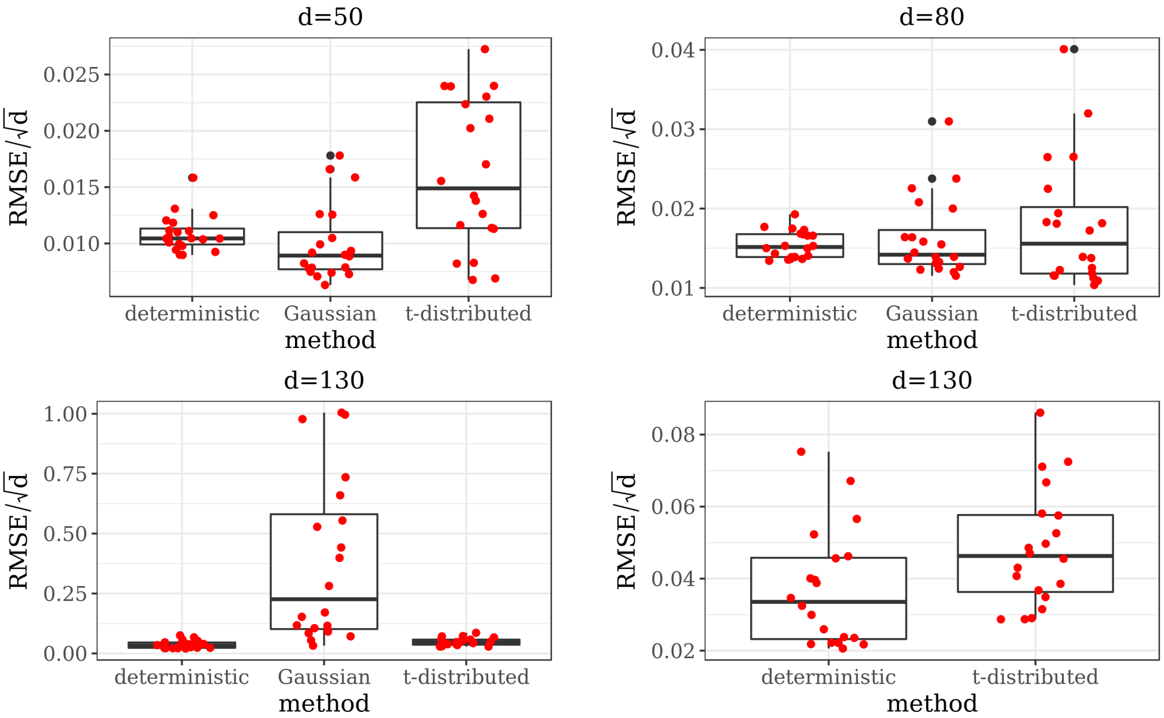


Figure 3.4: Boxplots of the values of RMSE/$\sqrt{d}$ for Example 1 across 20 runs of the experiment, dimensions 50, 80 and 130 for all the three methods for jumps implemented in `MultiMCMC`. Notice that each of the plots has a different scale of the y variable. The plot in the bottom right panel was added in order to visualise the differences between the deterministic and the t-distributed jumps in dimension 130.

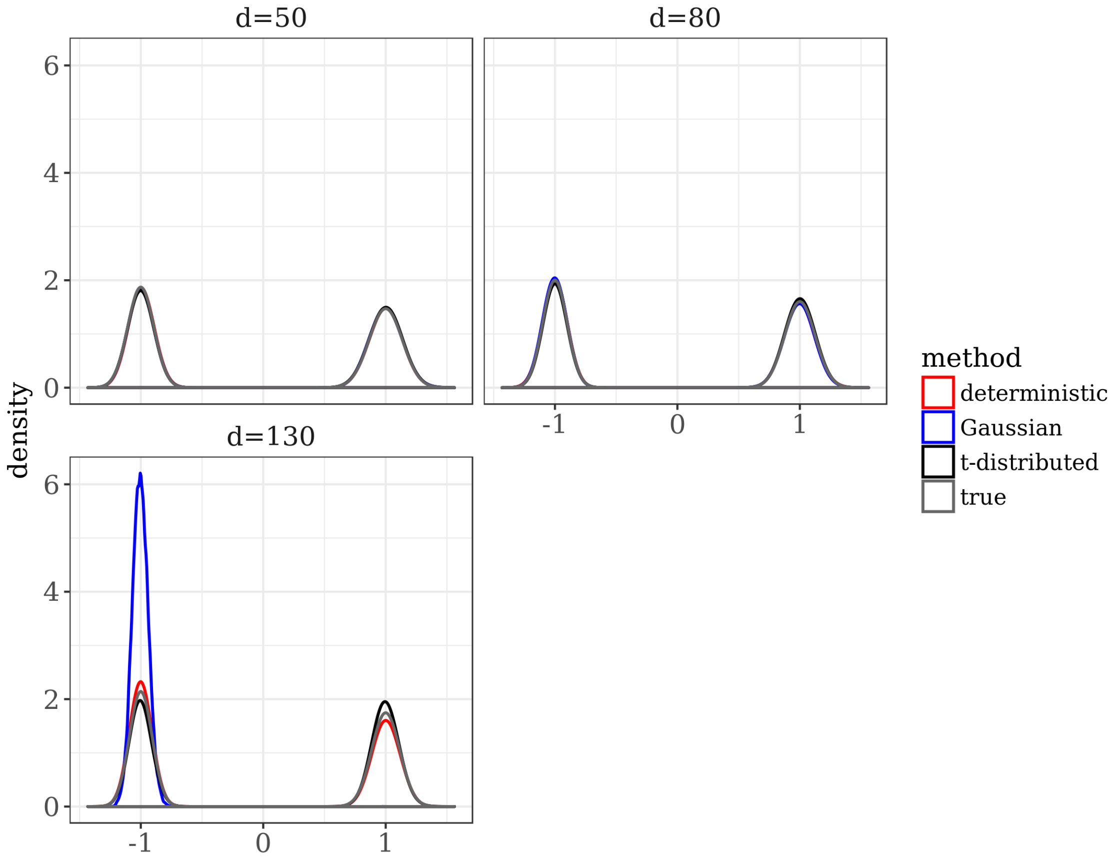


Figure 3.5: Density of the mean of all coordinates for Example 1, dimensions 50, 80 and 130. We compare the results obtained for the three methods (deterministic, Gaussian and t-distributed jumps) implemented in `MultiMCMC` with a sample generated from the true target distribution (`true`) For each of the three `MultiMCMC` methods the chain with the highest value of RMSE/$\sqrt{d}$ is presented.

dimension 130. The mode-finding was not particularly challenging here – both modes were discovered in all the experiments within the pre-specified number of BFGS runs.

Since the modes have a regular Gaussian shape, the within-mode mixing is a relatively simple task in this case, even in high dimensions. Indeed, our results show that the acceptance rate of local moves is close to 0.234 in both modes. What makes a difference is the mixing between modes, which can be measured by the acceptance rate of jumps. Therefore, for each of the 20 chains we calculated the minimal value of the acceptance rate of the jumps from one mode to another. The lowest and the highest values of this minimal acceptance rate across the 20 chains are reported in Table 3.2. The deterministic jumps proved to outperform the other two methods tremendously in high dimensions, with the acceptance rate ranging between 0.76 and 0.87 in dimension 130. Apparently this method is less sensitive to the accurate estimation of the covariance matrices than the Gaussian or t-distributed independent proposals.

3.2.2 Mixture of multivariate t and banana-shaped distributions

A classic example of a multimodal distribution is a mixture of 20 bivariate Gaussian distributions introduced in [Kou et al., 2006] (in two versions, with equal and unequal weights and covariance matrices). It was later studied also by [Miasojedow et al., 2013], [Tak et al., 2017] and [Nemeth et al., 2017]. Our algorithm works well on both versions, however, since the example is relatively simple and the existing methods already perform well on it, we do not expect our method to yield much improvement. Therefore, we decided to modify this example in the way described below in order to make it more challenging. Instead of the Gaussian distribution, the first five modes follow the banana-shaped distribution and the remaining ones – multivariate t with 7 degrees of freedom and the covariance matrices $0.01\sqrt{d}I_d$, where d is the dimension (the covariance matrices in the original example were given by $0.01I_2$). The weights are assumed to be equal to 0.05. We consider dimensions $d = 4$ and $d = 10$ by repeating the original coordinates of the centres of the modes twice and five times, respectively.

Recall the definition of the banana-shaped distribution introduced by [Haario et al., 1999b]. Let f be the density of the Gaussian distribution $N(0, C)$, where $C = \text{diag}(100, 1, \ldots, 1)$. Then the density of the banana-shaped distribution is given by

$$f_b = f \circ \phi_b,$$

where

$$\phi_b(x_1, \ldots, x_n) = (x_1, x_2 + bx_1^2 - 100b, x_3, \ldots, x_n). \tag{3.4}$$

In our case the constant b was set to 0.03. Furthermore, in dimension 4 the formula on the second coordinate of 3.4 was assigned to coordinate 2, 2, 4, 4, 4 for modes 1, 2, 3, 4, 5, respectively. In dimension 10 it was assigned to coordinate 2, 4, 6, 8, 10 for modes 1, 2, 3, 4, 5. We additionally multiplied each coordinate by $\sqrt{d}/10$ to decrease the variance.

Similarly to the previous example, we ran the algorithm 20 times for each method and we used a sample size of 500,000. The number of BFGS runs varied between 10,000 for dimension 4 and 30,000 for

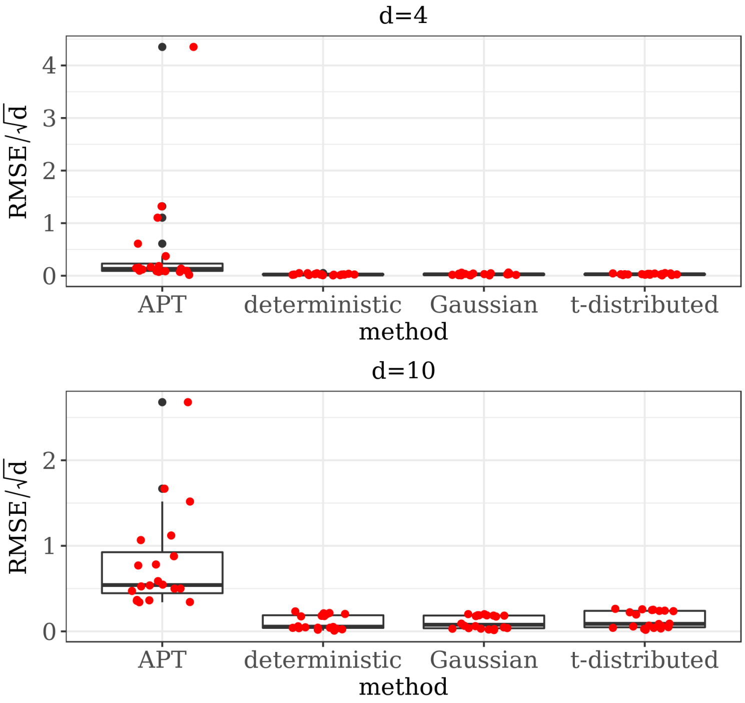


Figure 3.6: Boxplots of the values of RMSE/$\sqrt{d}$ for Example 2 across 20 runs of the experiment, dimensions 4 and 10. We consider the results for adaptive parallel tempering `APT` and three `MultiMCMC` methods: deterministic, Gaussian and t-distributed jumps.

dimension 10. The starting points were sampled uniformly on $[-2, 12]^d$. The initial covariance matrix estimation was running for 10,000 iterations per mode (split into 4 rounds). For the parallel tempering we used 2,100,000 iterations and 4 temperature levels with a burn-in period of length 600,000. Analogously to Example 1, we applied thinning and based our analysis on every third MCMC sample. Each of the 20 chains was initialised at a different mode. The full details of the settings of the experiments may be found in Table 3.1.

The mode-finding turned out to be much more problematic here than in Example 1, in particular, the domains of attraction of the banana-shaped modes were difficult to get to. For dimension 4 our method managed to find all the modes, but for dimension 10 all the modes were identified 10 times and remaining 10 runs missed mode number 4. Even in such case our method outperformed adaptive parallel tempering, as shown in the boxplots of Figure 3.6. In fact, the largest RMSE/$\sqrt{d}$ for dimension 10 across all the three `MultiMCMC` methods was still smaller than the smallest RMSE/$\sqrt{d}$ for `APT` (with a larger computational budget). Even in dimension 4 the results for `APT` were considerably less stable than for `MultiMCMC`, as illustrated by the scatterplots in Figure 3.7.

	deterministic		Gaussian		t-distributed	
	Lowest	Highest	Lowest	Highest	Lowest	Highest
$d = 4$	0.65	0.72	0.53	0.58	0.61	0.67
$d = 10$	0.65	0.74	0.42	0.49	0.54	0.63
$d = 50$	0.50	0.53	0.08	0.10	0.12	0.13

Table 3.3: The lowest and the highest value (across 20 runs of the experiment) of the minimal acceptance rates of jump moves between any two modes for Example 2.

The example studied here motivated asking a question how well our algorithm would deal with jumping between irregularly-shaped modes in high dimensions. To answer this question, we considered a similar example in dimension $d = 50$ (the centres of the modes were the original ones repeated 25 times) constructed as follows:

- modes 1-5 followed the banana-shaped distributions with $b = 0.03$ scaled by $\sqrt{d}/50$ with the formula in the second coordinate of (3.4) assigned to coordinates 2, 4, 6, 8, 10, respectively;
- modes 6-10 followed the multivariate t distribution with 7 degrees of freedom and the covariance matrix equal to $0.01\sqrt{d}I_d$;
- the remaining modes followed the normal distribution with the covariance matrix $0.01\sqrt{d}I_d$;
- all the weights of the mixture were equal to 0.05.

The mode-finding would be extremely difficult here so we assumed the locations of the modes were known. The left panel of Figure 3.8 shows that jumping between the modes is still efficient, especially in case of the deterministic jumps, and the empirical mean approximates well the true one. Analogously to the previous example, we measured the lowest acceptance rate of jumps between the modes. Not surprisingly, the jumps from and to the banana-shaped modes tended to exhibit lower acceptance rates than jumps between regularly-shaped modes. The results may be found in Table 3.3. The deterministic jumps again proved to ensure the best efficiency measured by the lowest acceptance rate between any two modes. The fact that even the jumps between banana-shaped modes in dimension 50 were accepted with the acceptance rate at least 0.5 shows that the deterministic jumps method indeed provides very good mixing properties, even for irregularly shaped modes. The right panel of Figure 3.8 presents the distribution of the weights of modes across the 20 simulations for different types of modes. For the deterministic jumps the weights corresponding to modes of all types were tightly concentrated around the true value 0.05, which again confirms the superiority of the deterministic jumps method.

This example shows that even though our algorithm may seem to be designed to work well on mixtures of Gaussians or t distributions, its good performance goes far beyond the class of target densities that are well-approximated by mixtures of spherical distributions. The bottleneck may be the mode-finding, but once the modes are known, the high dimension or the irregular shape of the mode is not a major obstacle.

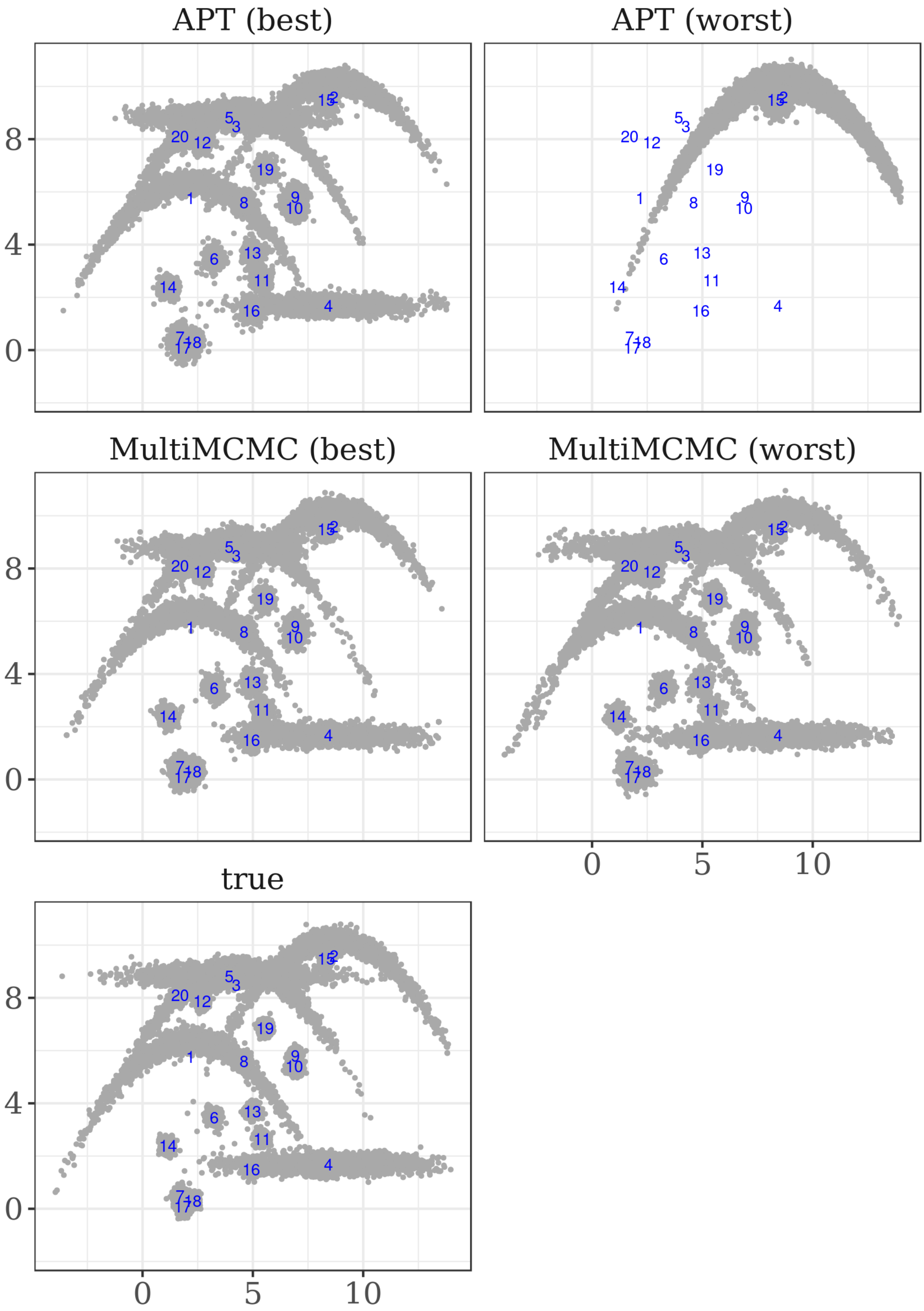


Figure 3.7: Scatterplots of the first and second coordinate (Example 2, dimension 4). As before, "best" and "worst" correspond to the lowest and highest value of RMSE/$\sqrt{d}$ across the 20 runs. For the `MultiMCMC` we present the results for the t-distributed jumps.

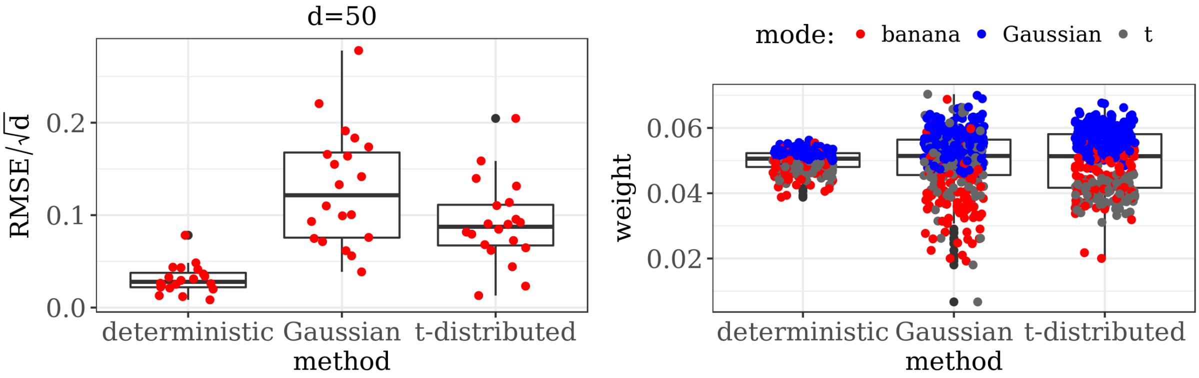


Figure 3.8: Left panel: a boxplot of the values of RMSE/$\sqrt{d}$ for Example 2 across 20 runs of the experiment, dimension 50 for the three methods implemented in `MultiMCMC` (deterministic, Gaussian and t-distributed jumps). Right panel: a boxplot of the weights of the modes across 20 runs of the experiments, for different types of modes.

Chapter 4

Conclusions and future work

The approach we proposed here is based on three fundamental ideas. Firstly, we split the task into mode finding and sampling from the target distribution. Secondly, we base our algorithm on local moves responsible for mixing within the same mode and jumps that facilitate crossing the low probability barriers between the modes. Finally, we account for inhomogeneity between the modes by using different proposal distributions for local moves at each mode and adapting their parameters separately. This is possible thanks to the auxiliary variable approach which enables assigning each MCMC sample to one of the modes and ensuring that it is unlikely to escape to another mode via local moves. The popular tempering-based approaches do not have the mechanism of controlling the mode at each step and therefore the adaptive parallel tempering ([Miasojedow et al., 2013]) algorithm learns the global covariance matrix rather than the local ones. This seems highly inefficient if the shapes of the modes are very distinct. The optimisation-based approaches are naturally well-suited for the task of collecting the MCMC samples separately for each mode and learning the covariance matrix on this basis. However, the approaches known in the literature tend to be either very costly ([Zhou, 2011]) or to ignore the issue of the possibility of moving between the modes via local steps ([Ahn et al., 2013]). Moreover, some of the other fundamental issues of optimisation-based methods have not been systematically addressed by the researchers so far. These include an efficient design of the mode-finding phase and distinguishing between newly discovered modes and replicated ones.

To develop a methodological approach and prove ergodic results for our algorithm, we introduced the Auxiliary Variable Adaptive MCMC class. As discussed briefly in Chapter 2, there are other adaptive algorithms falling in this category, so our theoretical results may potentially be useful beyond the scope of Adaptive MCMC for Multimodal Distributions. Furthermore, an important advantage of our method from the point of view of the modern compute resources is that a large part of the algorithm is parallelisable.

As shown in Chapter 3, our method is particularly useful when the user knows the locations of the modes, or at least knows them roughly and can narrow down the optimisation searches to some specific regions. In such cases the algorithm proves to work very well, even when the dimension of the problem is high or when the shape of the mode is irregular, e.g. banana shaped. Note that tempering-based methods, on the other hand, do not allow the user to benefit from the knowledge they have of the locations of the modes. The first example in Chapter 3 shows that the performance of adaptive parallel tempering was poor and unstable even when we ran the algorithm from the more narrow (and hence, more difficult to detect) mode.

The future directions of our work will focus primarily on enhancing the computational side of the method. Firstly, we are planning to implement efficient communication between the main algorithm and the burn-in algorithm, as shown in Figure 3.1. Additionally, we would like to implement the "learn from the neighbour" version of the algorithm (in the spirit of [Craiu et al., 2009]). Namely, apart from the burn-in algorithm running on several cores, we would have the main algorithm running in parallel on a number of cores. The parameters of the algorithm would be then updated jointly, every certain number of iterations, based on the samples collected in all the chains. This approach may be particularly useful in high dimensions when a very large number of samples is needed to estimate correctly the covariance matrices used as parameters of the proposal distributions.

What is more, we would like to improve the way we choose the starting points for the optimisation procedures to make mode-finding more efficient. One path we would like to explore is to run the Wang-Landau algorithm in order to detect the areas of high probability mass, which would give us a hint as for where to place the starting points. This idea, however, is quite vague at the moment and it is not clear how efficient it would be in practice and if we would manage to split this task across different cores.

Another possible improvement of our method could involve automation of the choice of the number of rounds in the burn-in algorithm. Recall that the between-mode mixing is strongly linked with the acceptance rate of jumps, which in turn depends heavily on how close the local empirical covariance matrices are to the true ones. In particular, for each of the three methods for jumps considered in this work, a term $\sqrt{\det \Sigma_k}/\sqrt{\det \Sigma_i}$ appears in the formula for the acceptance probability. One idea to explore would be to measure how much the value of $\det \Sigma_k$ for each mode changes between rounds of the burn-in algorithm and initialise the main algorithm when the modifications are no longer significant.

Finally, we would like to test how successful our method is when applied to real problems with multimodal posterior distributions. Initially our plan was to work on variable selection in Bayesian linear or logistic regression. Posterior distributions in these models are often multimodal when there is a strong correlation between covariates and when the number of observations is fairly small compared with the number of predictors. However, we found out that the modern interesting problems of this type usually have at least a few hundreds of predictors and MCMC-based methods are not best-suited to deal with such dimensions. An exception to this rule is randomised control trials as usually both the number of predictors and the number of observations is fairly small. One of the examples we are planning to look at involves Bayesian logistic regression applied to randomised control trial data on malaria.

Another example that we are planning to consider comes from a hierarchical media mix model. It is essentially a regression model used to quantify the influence of different marketing activities on key performance indicators. The particular model we are considering is standard hierarchical linear regression with Gaussian noise, where some of the covariates x_i are transformed using the Hill function

$$\mathrm{Hill}(x; K, S) = \frac{1}{1 + (x/K)^{-S}}$$

with unknown parameters K and S. In some settings the posterior distribution for $(S_1, \dots S_M, K_1, \dots, K_M)$ is multimodal, where M is the number of marketing activities (media) involved in the model.

Multimodal distributions are also common in astrophysics. One such example was used in [Tak et al., 2017]. Others appeared as posterior distributions in the Evidence Challenge, which was part of the Third Workshop on Extremely Precise Radial Velocities (see [Nelson et al., 2018]). One of our next steps will be to test our method on these examples.

Appendix A

Technical results

A.1 Proof of the detailed balance for the deterministic jumps

Fix $i, k \in \{1, \ldots, N\}$ such that $i \neq k$. Let f_{ik} be a function such that

$$f_{ik}(x, i) = (y, k) \quad \text{for} \quad y := \mu_k + \Lambda_k \Lambda_i^{-1}(x - \mu_i)$$

where

$$\Sigma_i = \Lambda_i \Lambda_i^T \quad \text{and} \quad \Sigma_k = \Lambda_k \Lambda_k^T.$$

Analogously,

$$f_{ki}(y, k) := \mu_i + \Lambda_i \Lambda_k^{-1}(y - \mu_k).$$

Note that $f_{ik}^{-1}(y, k) = f_{ki}(y, k) = (x, i)$. Observe additionally that

$$\det(Df_{ki})(y, k) = \det \Lambda_i \Lambda_k^{-1} = \frac{\det \Lambda_i}{\det \Lambda_k} = \frac{\sqrt{\det \Sigma_k}}{\sqrt{\det \Sigma_i}}. \tag{A.1}$$

We can now define formally the proposal kernel of the deterministic jump from mode i to mode k as

$$Q_{ik}\left(y|x\right) = \delta_{f_{ik}(x,i)}(y, k),$$

where δ stand for Dirac's δ. Recall that the acceptance probability of the jump from (x, i) to (y, k) is given by

$$\alpha_J\left((x, i) \to (y, k)\right) = \min\left[1, \frac{\tilde{\pi}(y, k)}{\tilde{\pi}(x, i)} \frac{a_i \sqrt{\det \Sigma_k}}{a_k \sqrt{\det \Sigma_i}}\right].$$

To prove that the detailed balance condition holds for the deterministic jump, it is sufficient to show that

$$
\begin{aligned}
&\int_{(y,k)\in B\times\{k\}}\int_{(x,i)\in A\times\{i\}} \tilde{\pi}(x,i)a_k\alpha_J\left((x,i)\to(y,k)\right)dxdQ_{ik}(y|x)\\
=&\int_{(y,k)\in B\times\{k\}}\int_{(x,i)\in A\times\{i\}} \tilde{\pi}(y,k)a_i\alpha_J\left((y,k)\to(x,i)\right)dydQ_{ki}(x|y)
\end{aligned}
$$

for A and B such that $f_{ik}(A\times\{i\})=B$ and $f_{ki}^{-1}(B\times\{k\})=A\times\{i\}$. Observe that using integration by substitution and formula A.1 (in the third and fourth equality below, respectively), we obtain

$$
\begin{aligned}
&\int_{(y,k)\in(B\times\{k\})}\int_{(x,i)\in(A\times\{i\})} \tilde{\pi}(x,i)a_k\alpha_J\left((x,i)\to(y,k)\right)dxdQ_{ik}(y|x)\\
=&\int_{(x,i)\in A\times\{i\}} \tilde{\pi}(x,i)a_k\alpha_J\left((x,i)\to f_{ik}(x,i)\right)dx\\
=&\int_{(x,i)\in f_{ki}^{-1}(B\times\{k\})} \tilde{\pi}(x,i)a_k\min\left[1,\frac{\tilde{\pi}\left(f_{ik}(x,i)\right)}{\tilde{\pi}(x,i)}\frac{a_i\sqrt{\det\Sigma_k}}{a_k\sqrt{\det\Sigma_i}}\right]dx\\
=&\int_{(y,k)\in(B\times\{k\})} \tilde{\pi}\left(f_{ki}(y,k)\right)a_k\min\left[1,\frac{\tilde{\pi}\left(f_{ik}(f_{ki}(y,k))\right)}{\tilde{\pi}\left(f_{ki}(y,k)\right)}\frac{a_i\sqrt{\det\Sigma_k}}{a_k\sqrt{\det\Sigma_i}}\right]\left|\det(Df_{ki})(y,k)\right|dy\\
=&\int_{(y,k)\in(B\times\{k\})} \min\left[\tilde{\pi}\left(f_{ki}(y,k)\right)a_k\frac{\sqrt{\det\Sigma_i}}{\sqrt{\det\Sigma_k}},\tilde{\pi}(y,k)a_i\right]dy\\
=&\int_{(y,k)\in(B\times\{k\})} \tilde{\pi}(y,k)a_i\min\left[1,\frac{\tilde{\pi}\left(f_{ki}(y,k)\right)}{\tilde{\pi}(y,k)}\frac{a_k\sqrt{\det\Sigma_i}}{a_i\sqrt{\det\Sigma_k}}\right]dy\\
=&\int_{(y,k)\in(B\times\{k\})} \tilde{\pi}(y,k)a_i\alpha_J\left((y,k)\to f_{ki}(y,k)\right)dy\\
=&\int_{(y,k)\in(B\times\{k\})}\int_{(x,i)\in(A\times\{i\})} \tilde{\pi}(y,k)a_i\alpha_J\left((y,k)\to(x,i)\right)dydQ_{ki}(x|y),
\end{aligned}
$$

which ends the proof.

A.2 Technical results for the multivariate t distribution

Assume that $Q_i(\mu_i,\Sigma_{\gamma,i})$ and $Q_k(\mu_k,\Sigma_{\gamma,i})$ follow the d-dimensional multivariate t distribution with ν degrees of freedom, centred at μ_i and μ_k, respectively and with covariance matrices $\frac{\nu}{\nu-2}\Sigma_{\gamma,i}$ and $\frac{\nu}{\nu-2}\Sigma_{\gamma,k}$. Observe that

$$
\begin{aligned}
\limsup_{|x|\to\infty}\sup_{\gamma\in\mathcal{Y}}\frac{Q_i(\mu_i,\Sigma_{\gamma,i})(x)}{Q_k(\mu_k,\Sigma_{\gamma,k})(x)} &= \limsup_{|x|\to\infty}\sup_{\gamma\in\mathcal{Y}}\frac{\left(\det\Sigma_{\gamma,i}\right)^{-1/2}}{\left(\det\Sigma_{\gamma,k}\right)^{-1/2}}\left(\frac{1+\frac{1}{\nu}(x-\mu_i)^T\Sigma_{\gamma,i}^{-1}(x-\mu_i)}{1+\frac{1}{\nu}(x-\mu_k)^T\Sigma_{\gamma,k}^{-1}(x-\mu_k)}\right)^{-(\nu+d)/2}\\
&\leq \limsup_{|x|\to\infty}\left(\sup_{\gamma\in\mathcal{Y}}\frac{\left(\det\Sigma_{\gamma,i}\right)^{-1/2}}{\left(\det\Sigma_{\gamma,k}\right)^{-1/2}}\right)\left(\frac{1+\frac{1}{\nu}(x-\mu_i)^T M^{-1}I_d(x-\mu_i)}{1+\frac{1}{\nu}(x-\mu_k)^T m^{-1}I_d(x-\mu_k)}\right)^{-(\nu+d)/2}\\
&= \left(\sup_{\gamma\in\mathcal{Y}}\frac{\left(\det\Sigma_{\gamma,i}\right)^{-1/2}}{\left(\det\Sigma_{\gamma,k}\right)^{-1/2}}\right)\left(\frac{M}{m}\right)^{(\nu+d)/2}<\infty,
\end{aligned}
\tag{A.2}
$$

since the supremum of the determinants is bounded from above and the infimum is greater than 0. Similarly,

$$\liminf_{|x|\to\infty} \inf_{\gamma} \frac{Q_i(\mu_i, \Sigma_{\gamma,i})(x)}{Q_k(\mu_k, \Sigma_{\gamma,k})(x)} > 0.$$

Therefore, combining (A.2) with (2.29) implies that if $Q_j(\mu_j, \Sigma_{\gamma,j})$ for $j \in \mathcal{I}$ follow the t distribution with ν degrees of freedom, then

$$\frac{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)}{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)} \leq \tilde{K} \quad \text{for all } x \in \mathcal{X} \text{ and } \gamma \in \mathcal{Y} \tag{A.3}$$

for some positive constant $\tilde{K}$, and consequently

$$\frac{w_{\gamma,i} Q_i(\mu_i, \Sigma_{\gamma,i})(x)}{\sum_{j\in\mathcal{I}} w_{\gamma,j} Q_j(\mu_j, \Sigma_{\gamma,j})(x)} \geq \frac{1}{\tilde{K}} \quad \text{for all } x \in \mathcal{X} \text{ and } \gamma \in \mathcal{Y}. \tag{A.4}$$

Bibliography

[Ahn et al., 2013] Ahn, S., Chen, Y., and Welling, M. (2013). Distributed and adaptive darting Monte Carlo through regenerations. In *Artificial Intelligence and Statistics*, pages 108–116.

[Andricioaei et al., 2001] Andricioaei, I., Straub, J. E., and Voter, A. F. (2001). Smart darting Monte Carlo. *The Journal of Chemical Physics*, 114(16):6994–7000.

[Atchadé et al., 2011] Atchadé, Y. F., Roberts, G. O., and Rosenthal, J. S. (2011). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Stat. Comput.*, 21(4):555–568.

[Bornn et al., 2013] Bornn, L., Jacob, P. E., Del Moral, P., and Doucet, A. (2013). An adaptive interacting Wang–Landau algorithm for automatic density exploration. *Journal of Computational and Graphical Statistics*, 22(3):749–773.

[Chimisov et al., 2018] Chimisov, C., Łatuszyński, K., and Roberts, G. (2018). Air markov chain monte carlo. *arXiv preprint arXiv:1801.09309*.

[Chopin et al., 2012] Chopin, N., Lelièvre, T., and Stoltz, G. (2012). Free energy methods for Bayesian inference: efficient exploration of univariate Gaussian mixture posteriors. *Statistics and Computing*, 22(4):897–916.

[Craiu et al., 2009] Craiu, R., Rosenthal, J., and Yang, C. (2009). Learn from thy neighbor: parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association*, 104(488):1454–1466.

[Duane et al., 1987] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222.

[Fok et al., 2017] Fok, R., An, A., and Wang, X. (2017). Optimization assisted MCMC.

[Geyer, 1991] Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood.

[Girolami and Calderhead, 2011] Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

[Haario et al., 1999a] Haario, H., Saksman, E., and Tamminen, J. (1999a). Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14(3):375–396.

[Haario et al., 1999b] Haario, H., Saksman, E., and Tamminen, J. (1999b). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14(3):375–396.

[Jarner and Hansen, 2000] Jarner, S. and Hansen, E. (2000). Geometric ergodicity of Metropolis algorithms. *Stochastic Processes and Their Applications*, 85(2):341–361.

[Jasra et al., 2015] Jasra, A., Paulin, D., and Thiery, A. H. (2015). Error Bounds for Sequential Monte Carlo Samplers for Multimodal Distributions. *arXiv preprint arXiv:1509.08775*.

[Kone and Kofke, 2005] Kone, A. and Kofke, D. A. (2005). Selection of temperature intervals for parallel-tempering simulations. *The Journal of chemical physics*, 122(20):206101.

[Kou et al., 2006] Kou, S., Zhou, Q., and Wong, W. H. (2006). Discussion paper equi-energy sampler with applications in statistical inference and statistical mechanics. *The annals of Statistics*, pages 1581–1619.

[Lan et al., 2014] Lan, S., Streets, J., and Shahbaba, B. (2014). Wormhole Hamiltonian Monte Carlo. In *AAAI*, pages 1953–1959.

[Liang and Wong, 2000] Liang, F. and Wong, W. H. (2000). Evolutionary Monte Carlo: Applications to C p model sampling and change point problem. *Statistica sinica*, pages 317–342.

[Luengo and Martino, 2013] Luengo, D. and Martino, L. (2013). Fully adaptive Gaussian mixture Metropolis-Hastings algorithm. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6148–6152. IEEE.

[Maire et al., 2016] Maire, F., Friel, N., Mira, A., and Raftery, A. (2016). Adaptive Incremental Mixture Markov chain Monte Carlo. *arXiv preprint arXiv:1604.08016*.

[Marinari and Parisi, 1992] Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *EPL (Europhysics Letters)*, 19(6):451.

[Miasojedow et al., 2013] Miasojedow, B., Moulines, E., and Vihola, M. (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664.

[Mykland et al., 1995] Mykland, P., Tierney, L., and Yu, B. (1995). Regeneration in Markov Chain Samplers. *Journal of the American Statistical Association*, 90(429):233–241.

[Neal, 1996] Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6(4):353–366.

[Nelson et al., 2018] Nelson, B. E., Ford, E. B., Buchner, J., Cloutier, R., Díaz, R. F., Faria, J. P., Rajpaul, V. M., and Rukdee, S. (2018). Quantifying the Evidence for a Planet in Radial Velocity Data. *arXiv preprint arXiv:1806.04683*.

[Nemeth et al., 2017] Nemeth, C., Lindsten, F., Filippone, M., and Hensman, J. (2017). Pseudo-extended Markov chain Monte Carlo. *arXiv preprint arXiv:1708.05239*.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). Numerical optimization: Springer series in operations research and financial engineering. *Springer, New York.*

[Peskun, 1973] Peskun, P. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612.

[R Core Team, 2014] R Core Team (2014). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

[Roberts and Rosenthal, 1997] Roberts, G. and Rosenthal, J. (1997). Geometric ergodicity and hybrid Markov chains. *Electron. Comm. Probab*, 2(2):13–25.

[Roberts and Rosenthal, 2004] Roberts, G. and Rosenthal, J. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71.

[Roberts and Rosenthal, 2007] Roberts, G. and Rosenthal, J. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2):458.

[Roberts and Rosenthal, 2009] Roberts, G. and Rosenthal, J. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367.

[Roberts and Rosenthal, 2013] Roberts, G. O. and Rosenthal, J. S. (2013). A note on formal constructions of sequential conditional couplings. *Statistics & Probability Letters*, 83(9):2073–2076.

[Schreck et al., 2013] Schreck, A., Fort, G., and Moulines, E. (2013). Adaptive equi-energy sampler: convergence and illustration. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(1):5.

[Sminchisescu and Welling, 2011] Sminchisescu, C. and Welling, M. (2011). Generalized darting monte carlo. *Pattern Recognition*, 44(10):2738–2748.

[Tak et al., 2017] Tak, H., Meng, X.-L., and van Dyk, D. A. (2017). A Repelling-Attracting Metropolis Algorithm for Multimodality. *Journal of Computational and Graphical Statistics.*

[Tjelmeland and Hegstad, 2001] Tjelmeland, H. and Hegstad, B. (2001). Mode jumping proposals in mcmc. *Scandinavian journal of statistics*, 28(1):205–223.

[Vihola, 2011] Vihola, M. (2011). On the stability and ergodicity of adaptive scaling Metropolis algorithms. *Stochastic Processes and their Applications*, 121(12):2839–2860.

[Wang and Landau, 2001a] Wang, F. and Landau, D. (2001a). Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E*, 64(5):056101.

[Wang and Landau, 2001b] Wang, F. and Landau, D. (2001b). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050.

[Woodard et al., 2009a] Woodard, D., Schmidler, S., and Huber, M. (2009a). Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electronic Journal of Probability*, 14:780–804.

[Woodard et al., 2009b] Woodard, D. B., Schmidler, S. C., and Huber, M. (2009b). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability*, 19(2):617–640.

[Zhou, 2011] Zhou, Q. (2011). Multi-Domain Sampling With Applications to Structural Inference of Bayesian Networks. *Journal of the American Statistical Association*, 106(496):1317–1330.