Multicore adaptive MCMC for multimodal distributions

Emilia Pompe[1], joint work with Chris Holmes[1] and Krzysztof Łatuszyński[2]

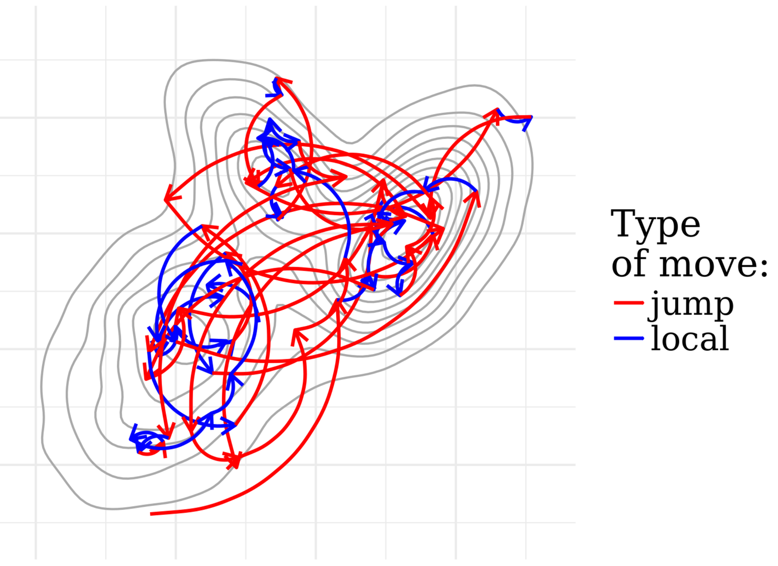
[1]University of Oxford, Department of Statistics [2]University of Warwick, Department of Statistics

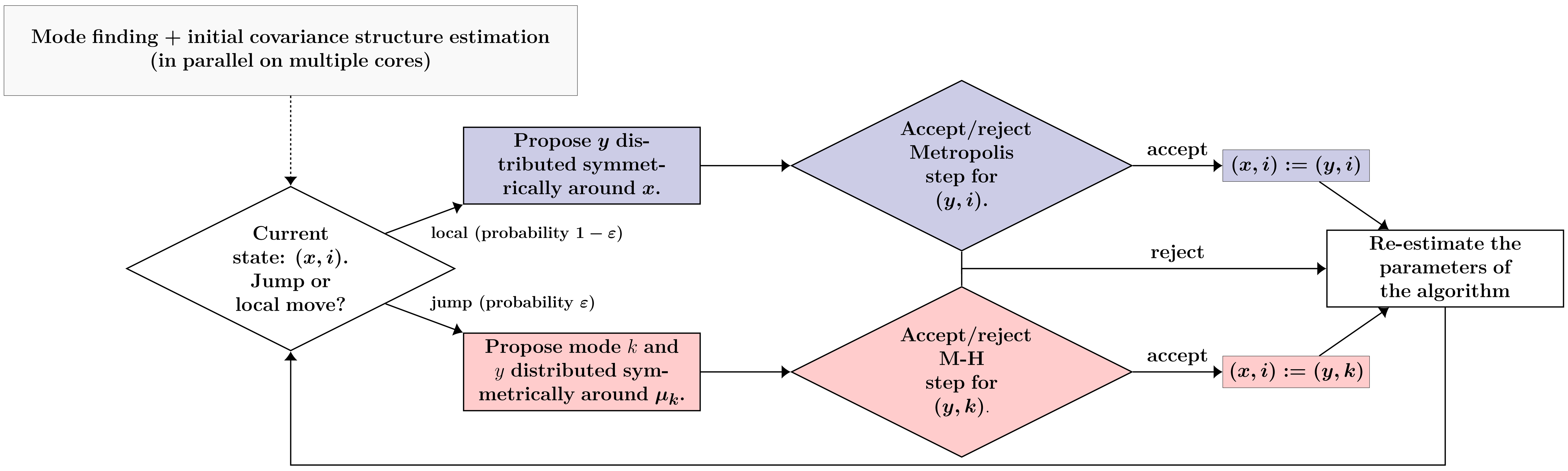
UNIVERSITY OF OXFORD DEPARTMENT OF STATISTICS

OxWaSP

Description of the algorithm

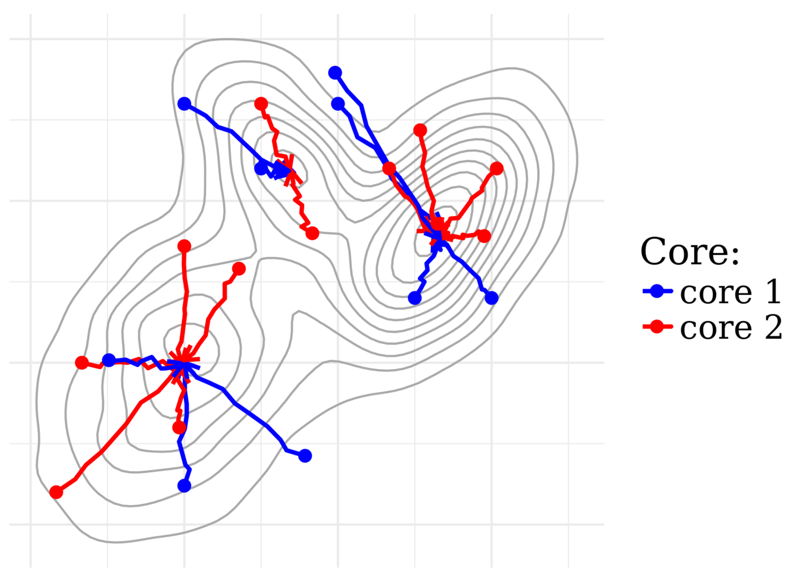
1. Let π be the target distribution on $\mathcal{X} = \mathbb{R}^d$ and let $\mathcal{I} = \{\mu_1, \ldots, \mu_N\}$ be the set of its modes. We define a new target distribution $\tilde{\pi}$ on the **augmented state space** $\mathcal{X} \times \mathcal{I}$
$$\tilde{\pi}(x, i) := \pi(x) \frac{w_i Q_i(\mu_i, \Sigma_i)(x)}{\sum_{j \in \mathcal{I}} w_j Q_j(\mu_j, \Sigma_j)(x)},$$
where w_j are weights and $Q_j(\mu_j, \Sigma_j)$ is an elliptical distribution centred at μ_j with the covariance matrix Σ_j, e.g. Q_i is the **multivariate normal or multivariate** t. π **is the marginal distribution of** $\tilde{\pi}$ with respect to its $\mathcal{X}$-coordinate.
2. An optimisation algorithm running in the background **finds the locations of the modes** $\mu_1, \ldots, \mu_N$ and passes them to the main MCMC sampler.
3. The algorithm **learns its parameters as it runs**: it updates the weights w_j and the matrices Σ_j so that the mixture $\sum_{j \in \mathcal{I}} w_j Q_j(\mu_j, \Sigma_j)(x)$ provides a good estimate of $\pi(x)$.
4. The algorithm explores the state space $\mathcal{X} \times \mathcal{I}$ via **local moves**, preserving the mode, and **jumps** to a region associated with a different mode.
   - **local moves** are steps of the Metropolis algorithm targeting $\tilde{\pi}$;
   - **jumps** to mode k are steps of the Metropolis-Hastings algorithm targeting $\tilde{\pi}$, with independent proposals from a symmetric distribution centred at μ_k.





What properties would an ideal MCMC algorithm for multimodal distributions have?

Making use of multicore implementation. ✓



1. The main MCMC sampler is supported by an **optimisation algorithm running on multiple cores** from different starting points, which enables efficient exploration of the state space.
2. After a new mode has been identified, a standard **Adaptive MCMC procedure** is started from the mode. The samples collected this way give us an **initial estimate of the covariance matrix** for this mode.

Provable ergodicity under mild regularity conditions. ✓

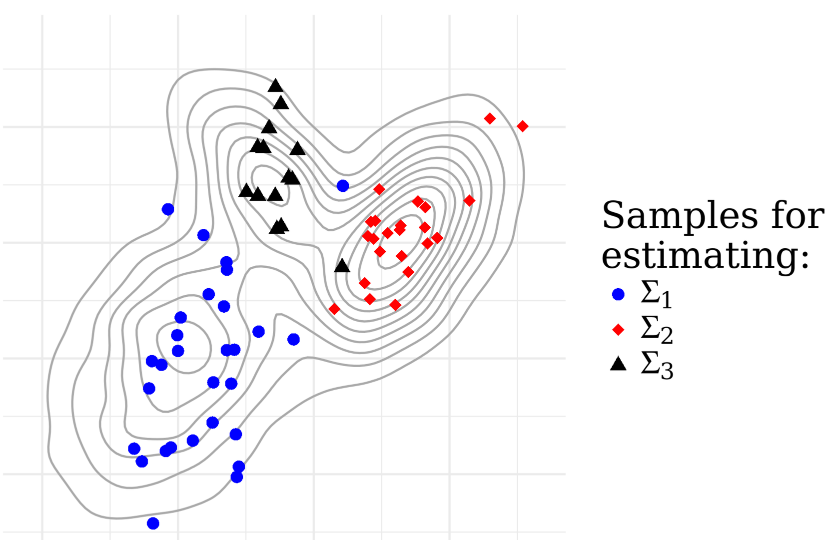
- The target distribution keeps being modified as the algorithm runs, so what would ergodicity mean? We consider ergodicity on sets $B \times \mathcal{I}$ for $B \subseteq \mathcal{X}$.
- The algorithm falls into the category of **Auxiliary Variable Adaptive MCMC** algorithms, for which analogous ergodic results to those of [Roberts and Rosenthal, 2007] can be proved.

Theorem 1. Assume that the mode finding algorithm stops adding new modes at a finite time with probability one. Then under
- **standard curvature conditions** for π and proposal distributions for local moves (see: [Jarner and Hansen, 2000]),
- appropriate **tail conditions** for Q_i and proposal distributions for jumps,

the **multicore adaptive MCMC algorithm for multimodal distributions is ergodic.**

Learning the local covariance structure around each mode on the fly. ✓

- The covariance matrices for each mode are estimated based on samples obtained around this mode so far. This allows the use of **optimal proposal distributions** for local moves.
- The **auxiliary variable** i indicates which element of the mixture the sample was drawn from. This enables the estimation of the local covariance structure, for each mode separately.
- The moves between modes take place via jumps, but it is **unlikely to escape to another mode using only local steps**. Suppose in a local move around mode i a point y, belonging to region associated with mode k, is proposed:



$$\text{acceptance probability} = \min\left[1, \frac{\tilde{\pi}(y, i)}{\tilde{\pi}(x, i)}\right] = \min\left[1, \frac{\pi(y) Q_i(\mu_i, \Sigma_i)(y) \sum_{j \in \mathcal{I}} w_j Q_j(\mu_j, \Sigma_j)(x)}{\pi(x) Q_i(\mu_i, \Sigma_i)(x) \sum_{j \in \mathcal{I}} w_j Q_j(\mu_j, \Sigma_j)(y)}\right].$$

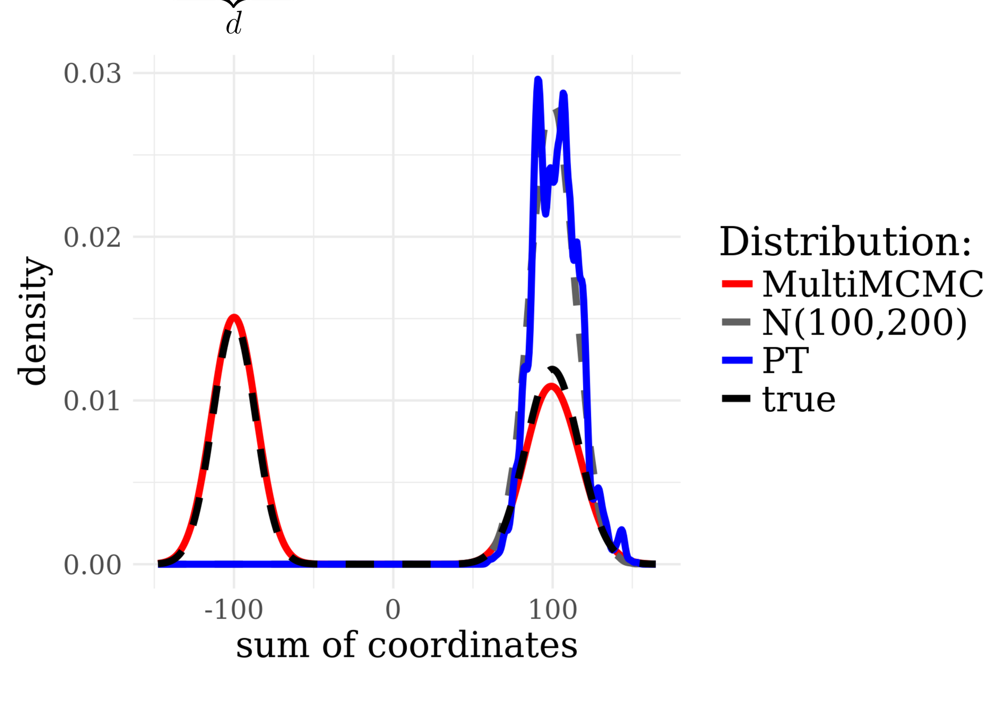
The ratio $\frac{Q_i(\mu_i, \Sigma_i)(y)}{Q_i(\mu_i, \Sigma_i)(x)}$ is typically tiny, so the probability of accepting such a move is very small.

Good mixing in practice on challenging examples. ✓/ ✗

We consider a modified version of the example used in [Woodard et al., 2009].

$$\text{target distribution} = 0.5N\left(-\mathbf{1}, \sigma_1^2 I_d\right) + 0.5N\left(\mathbf{1}, \sigma_2^2 I_d\right),$$

where $\mathbf{1} = \underbrace{(1, \ldots, 1)}_{d}$ and $\sigma_1 \neq \sigma_2$. In this case $d = 100$, $\sigma_1^2 = 1$ and $\sigma_2^2 = 2$.



Our algorithm (`MultiMCMC`) outperformed Parallel Tempering (`PT`) on this example.

Based on 10^5 iterations, with a 30% burn-in period. For the `PT`*, 10 temperatures were used, with the average acceptance rate of the swaps between temperatures equal to 0.34.*

However, main bottleneck: mode finding in high dimensions.

References

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

[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.

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