Multicore adaptive MCMC for multimodal distributions

Emilia Pompe\textsuperscript{1}, joint work with Chris Holmes\textsuperscript{1} and Krzysztof Łatuszyński\textsuperscript{2}

\textsuperscript{1}University of Oxford, Department of Statistics \hspace{1em} \textsuperscript{2}University of Warwick, Department of Statistics

Description of the algorithm

1. Let \( \pi \) be the target distribution on \( X = \mathbb{R}^d \) and let \( X = \{x_1, \ldots, x_N\} \) be the set of its modes. We define a new target distribution \( \tilde{\pi} \) on the augmented state space \( X \times \mathcal{I} \)

\[ \tilde{\pi}(x, i) = \pi(x) \prod_{j \in \mathcal{I}} w_j Q_j(p_j, \Sigma_j)(x) \]

where \( w_j \) are weights and \( Q_j(p_j, \Sigma_j) \) is an elliptical distribution centred at \( p_j \) with the covariance matrix \( \Sigma_j \), e.g. \( Q_j \) is the multivariate normal or multivariate \( t \) with the marginal distribution of \( i \) with respect to the \( X \)-coordinate.

2. An optimisation algorithm running in the background finds the locations of the modes \( x_1, \ldots, x_N \) 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 \( \Sigma_j \) so that the mixture \( \sum_{i \in X} \pi_i Q_i(p_i, \Sigma_i)(x) \) provides a good estimate of \( \pi(x) \).

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

Making use of multicore implementation.

- 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.
- After a new mode has been identified, a standard Adaptive MCMC procedure is started from the mode.
- The moves between modes take place via jumps, but it is unlikely to escape to another mode using only local steps.

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 \( B \times X \) for \( B \subseteq X \).
- 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 \( \pi \) 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 \( i \) at point \( y \) belonging to region associated with mode \( k \), it is proposed:

\[
\text{acceptance probability} = \min \left\{ \frac{\tilde{\pi}(y)}{\tilde{\pi}(x)}, \min_{s \in [0, 1]} \left( \frac{\pi_i Q_j(p_j, \Sigma_j)(y)}{\pi_i Q_j(p_j, \Sigma_j)(x)} \right)(1 + s) \right\}
\]

The ratio \( \frac{\pi_i Q_j(p_j, \Sigma_j)(y)}{\pi_i Q_j(p_j, \Sigma_j)(x)} \) 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]

\[
\text{target distribution} = 0.5 \mathcal{N} \left( -1, \sigma_1^2 \right) + 0.5 \mathcal{N} \left( 1, \sigma_2^2 \right),
\]

where \( \mathcal{N} \) is the standard normal distribution and \( \sigma_1^2, \sigma_2^2 \) in this case are \( 1, 2 \). Our algorithm (MultiMCMC) outperformed Parallel Tempering (PT) on this example.

References