Reversible Jump MCMC in Gravitational Wave Astronomy

Richard Stiskalek

Summer intern at AEI Hannover
University of Glasgow

richard.stiskalek@protonmail.com

Supervised by Dr Collin Capano

September 7, 2020

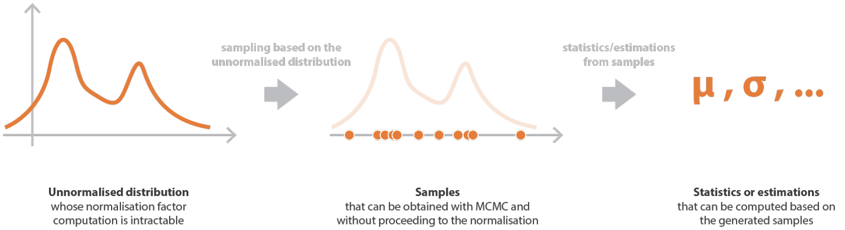
Introduction

1. Markov Chain Monte Carlo
    - Brief overview of MCMC methods
    - Metropolis-Hastings algorithm
    - Adaptive proposal for the Metropolis-Hastings algorithm
    - Tests on the Rosenbrock function
    - Applicability of adaptive proposals in GW inference
2. Reversible Jump MCMC (RJMCMC)
    - Short introduction to RJMCMC
    - Nested models
    - Birth-death RJMCMC
    - Black hole ringdown

Markov chain Monte Carlo methods

- Direct sampling of N-dimensional posteriors often intractable
- MCMC sequentially samples the posterior and constructs Markov chains whose distribution is proportional to the desired equilibrium distribution
- Bayes' theorem:

$$p(\boldsymbol{x} \mid \mathbf{d}, I) = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} = \frac{p(\mathbf{d} \mid \boldsymbol{x}, I)\, p(\boldsymbol{x} \mid I)}{\int p(\mathbf{d} \mid \boldsymbol{x}, I)\, p(\boldsymbol{x} \mid I)\, \mathrm{d}^n \boldsymbol{x}}$$



[https://towardsdatascience.com/bayesian-inference-problem-mcmc-and-variational-inference-25a8aa9bce29]

Metropolis-Hastings Algorithm

At iteration $i+1$ and current position $\boldsymbol{x}_i$:

1. Propose the next position $\boldsymbol{y}_{i+1} \sim q\left(\boldsymbol{y}_{i+1} \mid \boldsymbol{x}_i\right)$ and sample $u \sim U(0,1)$
2. Evaluate the acceptance ratio $\mathcal{A}$:

$$\mathcal{A}\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}_i\right) = \min\left[1, \frac{p\left(\boldsymbol{y}_{i+1} \mid \mathrm{d}, I\right)}{p\left(\boldsymbol{x}_i \mid \mathrm{d}, I\right)} \frac{q\left(\boldsymbol{x}_i \mid \boldsymbol{y}_{i+1}\right)}{q\left(\boldsymbol{y}_{i+1} \mid \boldsymbol{x}_i\right)}\right]$$

3. If $\mathcal{A}\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}_i\right) > u$ set $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1}$, else set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i$.

Proposal distribution

- The proposal distribution is paramount for the MH algorithm
- With an unsuitable proposal chains will have trouble burning in (potentially never), samples will be strongly correlated
- Can become very problem–specific!
- Common choice is a Gaussian, centred at the current position:

$$q\left(\boldsymbol{y}_{i+1} \mid \boldsymbol{x}_i\right) = \mathcal{N}\left(\boldsymbol{y}_{i+1} \mid \boldsymbol{x}_i, \boldsymbol{\Sigma}\right)$$

- What is the optimal proposal covariance then?

Adaptive proposal

- The covariance can either be set constant or adapted to the posterior
- The optimal covariance for a Gaussian proposal should ensure:
  1. The chain's distribution is representative of the posterior
  2. The chain burns in in a relatively short amount of time
  3. The chain's autocorrelation time is relatively small
- One solution to this are adaptive proposals: gather information about posterior during the burn-in phase to suggest optimal jumps
- Implemented an algorithm that trains the proposal covariance matrix to propose optimal jumps

- Burn-in: initial sampler positions are not drawn from the posterior, takes some number of iterations before samples correspond to the posterior

Adaptive Metropolis Algorithm (Andrieu & Thoms)

At iteration $i+1$, given $\boldsymbol{x}_i$, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, λ_i

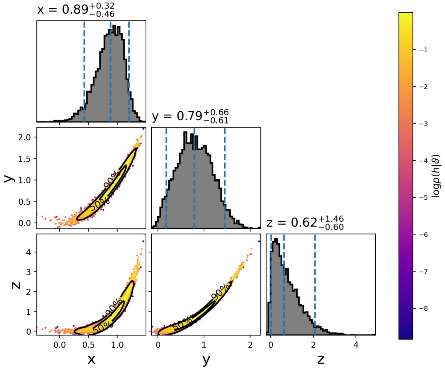
1. Propose the next position $\boldsymbol{y}_{i+1} \sim \mathcal{N}\left(\boldsymbol{y}_{i+1} \mid \boldsymbol{x}_i, \lambda_i \boldsymbol{\Sigma}_i\right)$ and $u \sim U(0,1)$
2. If $\mathcal{A}\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}_i\right) > u$ set $\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1}$, else set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i$.
3. Update:

$$\begin{aligned}
\log \lambda_{i+1} &= \log \lambda_i + \gamma_{i+1}\left[\mathcal{A}\left(\boldsymbol{y}_{i+1}, \boldsymbol{x}_i\right) - \widehat{\mathcal{A}}\right] \\
\boldsymbol{\mu}_{i+1} &= \boldsymbol{\mu}_i + \gamma_{i+1}\left(\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_i\right) \\
\boldsymbol{\Sigma}_{i+1} &= \boldsymbol{\Sigma}_i + \gamma_{i+1}\left[\left(\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_i\right)\left(\boldsymbol{x}_{i+1} - \boldsymbol{\mu}_i\right)^{\intercal} - \boldsymbol{\Sigma}_i\right],
\end{aligned}$$

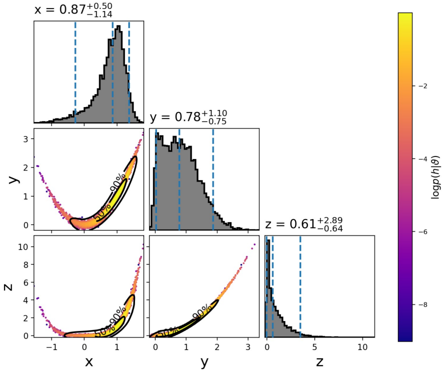
where $\gamma_i \propto i^{-0.6}$ ensures vanishing adaptation during burn-in

Adaptive Metropolis: test on 3D Rosenbrock function

Adaptive Metropolis Algorithm:



emcee_pt (16 temperatures)



- Comparable ACTs for both samplers
- Unlike emcee_pt, Adaptive Metropolis did not employ parallel tempering in this test (standard emcee never converges on this distribution)

Adaptive Metropolis: GW inference

- Prevalent sampler in PyCBC is emcee_pt:
  - created for sampling multi-modal distributions
  - may struggle with "valley-like" distributions
- Can Adaptive Metropolis be successfully used in GW inference?
  - sky location
  - distance and inclination
- Currently we are testing the proposal on real GW events

Reversible Jump MCMC

- Used to solve "trans-dimensional" problems, where the number of "unknowns" is unknown
- Can be generically formulated as joint Bayesian inference about a model indicator k and a parameter vector $\boldsymbol{x}_k$:

$$p\left(k, \boldsymbol{x}_k \mid \mathbf{d}\right) = p\left(k \mid \mathbf{d}\right) p\left(\boldsymbol{x}_k \mid k, \mathbf{d}\right)$$

- Relative support for a model k:

$$p\left(k \mid \mathbf{d}\right) = \int p\left(k \mid \mathbf{d}\right) p\left(\boldsymbol{x}_k \mid k, \mathbf{d}\right) \mathrm{d}^N \boldsymbol{x}_k$$

Gravitational-wave applications

- Ringdown analysis (*this work*)
- Resolution of LISA overlapping binaries ▸ Littenberg et al.
- Un-modelled searches ▸ BayesWave

Reversible Jump MCMC

- Comparing spaces on which the Markov chains are constructed:

$$\mathcal{X}_{\mathrm{MCMC}} = \mathcal{R}^{n}, \quad \mathcal{X}_{\mathrm{RJMCMC}} = \bigcup_{k \in \mathcal{K}} \left(\{k\} \times \mathcal{R}^{n_k}\right) \tag{1}$$

- The dimension $\mathcal{R}^{n_k}$ of each model can be generally different
- Finding relative support for several distributions given a set of data with a single Markov chain on $\mathcal{X}_{\mathrm{RJMCMC}}$
- The acceptance ratio for a trans-dimensional move $x_k = \{k, \boldsymbol{x}_k\}$ to $x_j = \{j, \boldsymbol{x}_j'\}$ is

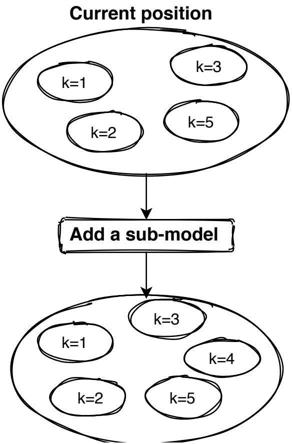
$$\mathcal{A}\left(x_j, x_k\right) = \min\left[1, \boxed{\frac{\pi\left(x_j\right)}{\pi\left(x_k\right)}} \frac{j_m\left(x_j\right)}{j_m\left(x_k\right)} \frac{g_m'\left(u'\right)}{g_m\left(u\right)} \boxed{\left|\frac{\partial g_{k \to j}\left(x_j, u'\right)}{\partial\left(x_k, u\right)}\right|}\right],$$

$|\ldots|$ is the determinant of the Jacobian of the mapping between the two models (because the mapping is specified in terms of x_j and u')

Nested trans-dimensional models

- Calculating the Jacobian term becomes, again, generally very problem–specific
- Nested trans-dimensional models: for $k < j$ and a move $k \to j$ the k model is nested within j, so only only have to add some new model components
- For nested models the Jacobian term is just the ratio of prior volumes:

$$\mathcal{A}\left(x_j, x_k\right) = \min\left[1, \frac{\pi\left(x_j\right)}{\pi\left(x_k\right)} \frac{q\left(x_k \mid x_j\right)}{q\left(x_j \mid x_k\right)}\right]$$

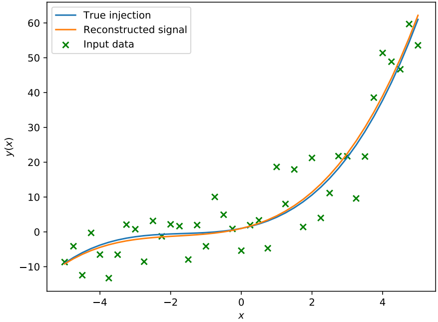


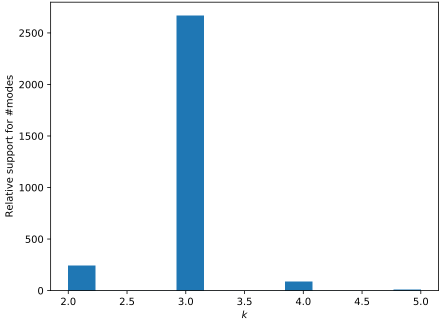
Birth-death RJMCMC

- Birth-death is a specific implementation of trans-dimensional moves for nested models
- The model index k serves as a counter of active models, use a proposal to suggest jumps for k
- Available moves for $k \to j$:
  - Birth: $k < j$ addition of a model(s)
  - Death: $k > j$ removal of a model(s)
  - Update: $k = j$ no addition/removal
- Birth: randomly activate a previously inactive model(s). Sample its position from a “birth” distribution
- Death: randomly remove a previously active model
- Update: MCMC move for all active models

Birth-death RJMCMC: Toy model

- Polynomial regression $y(x) = a_0 + \sum_{j=1}^{N} a_j x^j$
- a_0 is a free parameter shared by all trans-dimensional models
- $a_1, \ldots a_N$ are the possible constituent trans-dimensional models

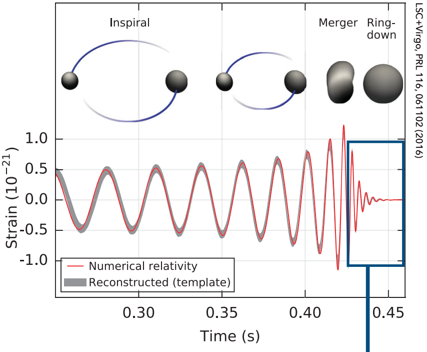




- Identifies the true number of active models in the injection ($k = 3$)
- Finds marginal support for $k = 2, 4$ due to the noisy data

Black hole ringdown

- The remnant BH radiates GWs while settling down to a Kerr BH
- Perturbation theory: ringdown consists of a superposition of exponentially damped sinusoids (quasinormal modes)
- Quasinormal modes labeled by l, m, n
- Each mode characterised by frequency, decay time, amplitude and phase



$$h_{\mathrm{RD}}(t) = \sum_{\ell mn} A_{\ell mn} e^{-t/\tau_{\ell mn}} \cos\left(2\pi f_{\ell mn} t + \phi_{\ell mn}\right)$$

No-hair theorem

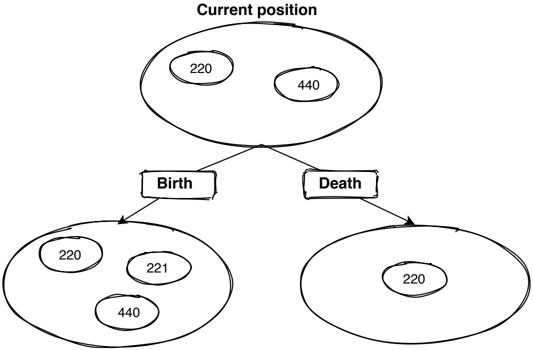
- No–hair theorem: stationary astrophysical BHs fully characterised by mass and spin (charge is negligible)
- Quasinormal modes uniquely determined by the BH's mass and spin in standard GR

  $\Rightarrow$ Tests of GR

- Finding evidence for the presence of subdominant quasinormal modes in data typically via nested sampling algorithms
- Can RJMCM be used to explore several combinations of quasinormal modes, while providing accurate model selection estimates?

Ringdown analysis with RJMCMC

- In principle can be done similarly as the polynomial birth model
- Add or remove exponentially damped sinusoids: modes only differ by frequency and decay time
- About to investigate the convergence of our algorithm on injections



Future plans & Conclusion

- Adaptive MCMC proposal:
  - Demonstrated suitability of the Adaptive Metropolis proposal on test distributions and implemented this in Epsie
  - Add support to PyCBC inference for a proposal that is well-suited for "banana"-shaped, strongly correlated posteriors, relax the need for parallel tempering
  - Potentially add a proposal dedicated to sampling the sky localisation directly on the surface on a sphere, not its projection
- RJMCMC ringdown analysis
  - Demonstrated a working RJMCMC algorithm on simple problems
  - Add support to PyCBC inference for a reversible jump MCMC capable of efficiently recovering the quasinormal modes

Thank you!

Birth distribution

- Efficiency of any birth-death algorithm heavily dependant on the birth distribution
- In case of a “bad” birth distribution all births might be rejected
- Currently have support for uniform distribution - can be easily extended to a Gaussian, etc.
- Setting the birth distribution to any particular probability distribution requires some prior knowledge
- Ongoing investigation:can the optimal birth distribution be learnt during burn-in (KDE, ...)?