Randomized-acceptance MCMC and the separation of chains

Colin Fox fox@physics.otago.ac.nz
Geoff Nicholls, Alexis Muir Watt

Outline

How to compute approximately, and get exactly the right answer ...

- Prelude
  - MCMC for IP
  - One history of inexact MCMC
- MCMC with randomized (estimated) acceptance probability
  - approximate evaluation of the forward map
  - standard- randomized- naïve- exact- approximate-MCMC
- (1/) Separation time as a 'distance' between Markov chains
  - $\tau_{\text{sep}} \gg \tau_{\text{IACT}} \Rightarrow$ approximation is 'exact'
- Scaling of separation time for some (pairs of) algorithms
  - plugging in estimates in r-MCMC is best

Bayesian analysis of inverse problems is simple!

Measurement process: (θ deterministic, $v \sim \pi_{\mathsf{n}}$ random)

$$\begin{aligned} d &= G(\theta, v) \\ &\simeq A(\theta) + v \end{aligned}$$

when A is known forward map

Explore posterior

$$\pi(\theta|d) \propto \pi_{\mathsf{n}}(d - A(\theta))\, \pi_{\mathsf{pr}}(\theta)$$

Write posterior as

$$\pi(\theta) = \exp\{-V(\theta)\}$$

where typically 'potential' $V(\theta) = \chi(A(\theta) - d) + \rho(\theta)$ for simple functions χ and ρ

Can evaluate $V(\theta)$ hence $\pi(\theta)$ for any θ – requires evaluation of A, i.e. simulate physics, etc – hence can implement MCMC

That can be expensive to do *exactly*. How inexact can we get away with?

standard MCMC (s-MCMC)

Let $\theta \sim \pi(d\theta)$ be a target variable with density $\pi(\theta)$. Let $Q(\theta, d\theta')$ be a Hastings proposal distribution with density $q(\theta, \theta')$, satisfying $q(\theta, \theta') > 0 \Leftrightarrow q(\theta', \theta) > 0$. Then

$$h(\theta, \theta') = \frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')} = \mathbf{e}^{\{V(\theta) - V(\theta')\}} \frac{q(\theta', \theta)}{q(\theta, \theta')} \qquad \alpha(\theta, \theta') = \min\left\{1, h(\theta, \theta')\right\}$$

gives the standard Metropolis Hastings acceptance probability

Algorithm 1 (s-MCMC) *At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:*

1. *Simulate $\theta' \sim q(\theta, \cdot)$.*
2. *With probability $\alpha(\theta, \theta')$ set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.*

Targets π when the chain is aperiodic (minorization condition) and irreducible, since the simulated transition kernel is in detailed balance with π, i.e.,

$$\pi(\theta)q(\theta, \theta')\alpha(\theta, \theta') = \pi(\theta')q(\theta', \theta)\alpha(\theta', \theta)$$

We call this an *exact* MCMC because it provably targets π.

Observation: must be robust to error in detailed balance

Approximate MCMC

If we have an estimate $V^*(\theta')$ or $V^*_\theta(\theta')$ of $V(\theta')$ (Monte Carlo, numerical evaluation, linearization, etc) simply plug that in

$$h^*(\theta, \theta') = \mathbf{e}^{\{V^*(\theta) - V^*(\theta')\}} \frac{q(\theta', \theta)}{q(\theta, \theta')} \qquad \alpha^*(\theta, \theta') = \min\left\{1, h^*(\theta, \theta')\right\}$$

and proceed as with s-MCMC.

In general does not target π, may not even have an equilibrium distribution.

We call this a *naïve* algorithm because we have simply plugged estimates into an exact algorithm.

Delayed acceptance uses one step of this inexact algorithm as a *proposal* in a s-MCMC to give an exact algorithm. That requires (exact) evaluation of $V(\theta)$ so we gain a speedup, but not a better scaling.

F Nicholls 1997, Christen F 2005

Penalty method (Ceperley and Dewing 1999)

(from now on wlog take proposal as symmetric)

Suppose we have a normal estimator (e.g. in QMC)

$$\hat{D}_{\theta,\theta'} \sim N(V(\theta') - V(\theta), \sigma^2(\theta,\theta'))$$

with *known* $\sigma^2 \neq 0$.

The naïve acceptance probability

$$\alpha_{\mathsf{N}}(\theta,\theta') = \min\left\{1, \mathrm{e}^{\hat{D}}\right\}$$

provably does not target π.

However, the corrected acceptance probability

$$\alpha_P(\theta,\theta') = \min\left\{1, \mathbf{e}^{\hat{D}-\sigma^2/2}\right\}$$

gives a kernel that is in detailed balance with π, so is exact (!)

Ceperley and Dewing 1999 solved an integral equation to show this. We will prove this more simply, and generalize to a wider class of (randomized) algorithms.

Randomized-acceptance MCMC (r-MCMC)

Target $\pi(\theta)$, proposal $q(\theta, \theta')$ as before. Let X be a scalar random variable with probability density $\xi(x; \theta, \theta')$ with support W independent of θ and θ', and $f : W \to W$ an *involution*, i.e. f satisfies $f(f(x)) = x$, that has a derivative at ξ-a.e. $x \in W$.

The involution f can be thought of as pairing points $(x, f(x))$ in W.

$$h_\xi(\theta, \theta'; x) = h(\theta, \theta') \frac{\xi(f(x); \theta', \theta)}{\xi(x; \theta, \theta')} |f'(x)| \qquad \alpha_\xi(\theta, \theta'; x) = \min\left\{1, h_\xi(\theta, \theta'; x)\right\}$$

is a randomized acceptance probability, that depends on the value of X.

Then the r-algorithm

Algorithm 2 (r-MCMC) *At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:*

1. *Simulate $\theta' \sim q(\theta, \cdot)$ and $x \sim \xi(\cdot; \theta, \theta')$.*
2. *With probability $\alpha_\xi(\theta, \theta'; x)$ set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.*

targets π.

r-MCMC satisfies detailed balance on average

Acceptance probability is $\alpha_\xi(\theta,\theta') = \int_W \xi(x;\theta,\theta')\alpha_\xi(\theta,\theta';x)dx$ and we wish to establish detailed balance, i.e.,

$$\pi(\theta)q(\theta,\theta')\alpha_\xi(\theta,\theta') = \pi(\theta')q(\theta',\theta)\alpha_\xi(\theta',\theta).$$

Multiply expression for α_ξ by $\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta')$

$$\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta')\alpha_\xi(\theta,\theta';x) = \min\{\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta'), \pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)|f'(x)|\}$$

similarly,

$$\pi(\theta')q(\theta',\theta)\xi(y;\theta',\theta)\alpha_\xi(\theta',\theta;y) = \min\{\pi(\theta')q(\theta',\theta)\xi(y;\theta',\theta), \pi(\theta)q(\theta,\theta')\xi(f(y);\theta,\theta')|f'(y)|\}$$

and set $y = f(x)$ so $x = f(y)$ and $f'(y) = 1/f'(x)$ to get

$$\pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)\alpha_\xi(\theta',\theta;f(x))|f'(x)| = \min\{\pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)|f'(x)|, \pi(\theta)q(\theta,\theta')\xi(x;$$

which has the RHS equal to the RHS of first eqn.
Hence 'very detailed balance'

$$\pi(\theta)q(\theta,\theta')\alpha_\xi(\theta,\theta';x)\xi(x;\theta,\theta') = \pi(\theta')q(\theta',\theta)\alpha_\xi(\theta',\theta;f(x))\xi(f(x);\theta',\theta)|f'(x)|$$

Integrating over all x in W establishes detailed balance.

Further properties of r-MCMC

- Under weak conditions, the r-chain to inherits π-irreducibility and minorization from the corresponding s-chain. Can be used to establish ergodicity in particular cases
- Results hold under the generalization from scalar to multivariate X. That is, if $X = (X_1, ..., X_K)$ has multivariate density $\xi(x; \theta, \theta')$ with support W in $\Re^K$, and f is an involution of W having Jacobian $f'(x)$ with determinant $|f'(x)|$, then the r-algorithm targets π.
- r-MCMC is, in general, less statistically efficient than the s-MCMC from which it is derived because the r-chain below the s-chain in Peskin ordering. Hence r-chain estimators have greater asymptotic variance than corresponding s-chain estimators.

A simple example

An s-algorithm targeting $\pi(\theta)$ with symmetric proposal $q(\theta, \theta') = q(\theta', \theta)$ has acceptance probability

$$\alpha(\theta, \theta') = \min\left\{1, \frac{\pi(\theta')}{\pi(\theta)}\right\}.$$

Consider the (randomized) estimate X of $\log(\pi(\theta')/\pi(\theta))$ with normal density $\xi(x; \theta, \theta') = N(x; \log(\pi(\theta')/\pi(\theta)), 1)$, and the identity involution $f(x) = x$, to get the following r-algorithm:

Algorithm 3 (toy r-MCMC) *At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:*

1. *Simulate $\theta' \sim q(\theta, \cdot)$ and $X \sim N(\log(\pi(\theta')/\pi(\theta)), 1)$.*
2. *With probability*

$$\alpha_\xi(\theta, \theta'; x) = \min\left\{1, \left(\frac{\pi(\theta')}{\pi(\theta)}\right)^{1-2X}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

As shown above, this algorithm satisfies detailed balance with respect to π.

The penalty method is a r-MCMC

The acceptance probability in the s-algorithm is $\min(1, \exp(D(\theta, \theta')))$. Let X have a normal density, $\xi(x; \theta, \theta') = N(x; 0, \sigma^2)$ and take for f the involution $f(x) = \sigma^2 - x$. It follows that

$$\frac{\xi(f(x); \theta', \theta)}{\xi(x; \theta', \theta)} |f'(x)| = \mathrm{e}^{x - \sigma^2/2}$$

and hence the acceptance probability in the r-algorithm is

$$\alpha_\xi(\theta, \theta'; x) \min\left\{1, \mathrm{e}^{D(\theta,\theta') + x - \sigma^2/2}\right\}$$

which is the penalty method.

Other examples of r-algorithms:

- 'universal rule' of Ball et al 2003 (uses two randomizations)
- single variable exchange of Murray and MacKay 2006 (identity involution)

Penalty estimate method (Ceperley and Dewing 1999)

As before, suppose we have a normal estimator

$$\hat{D}_{\theta,\theta'} \sim N(V(\theta') - V(\theta), \sigma^2(\theta,\theta'))$$

but *we don't know* $\sigma^2 \neq 0$. Instead we plug in an estimate s^2 of σ^2.

Algorithm 4 (penalty estimate method) *At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:*

1. *Simulate $\theta' \sim q(\theta, \cdot)$ and $\hat{D}_{\theta,\theta'} \sim N(V(\theta') - V(\theta), \sigma^2(\theta,\theta'))$. Form an independent unbiased variance estimate s^2 of σ^2*
2. *With probability*

$$\alpha_{\hat{P}}(\theta,\theta') = \min\left\{1, \mathbf{e}^{\hat{D}-s^2/2}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

This algorithm is inexact, so can't be a r-algorithm. However it is practical and probably useful. How useful? I claim it is better than a naïve s-MCMC.

Separation times and approximate-target MCMC

We analyze a coupling algorithm to show that the naive algorithm gives *exactly the same* MCMC samples as the exact penalty method, out to $O(m)$ steps, where m is the sample size used in D-estimation.

$\hat{D}_{\theta,\theta'}$ is an estimator for $D(\theta,\theta') = V(\theta) - V(\theta')$ with cdf $G_m(\cdot;\theta,\theta')$, that need not be unbiased or normal. We do assume it satisfies a CLT, so that

$$G_m(x) = \Phi\left(\frac{x-D}{\sigma/\sqrt{m}}\right) + O(m^{-1/2}).$$

For example, if $\hat{D}_{\theta,\theta'}$ is computed from a realization of a geometrically ergodic Markov chain $W = \{W_i\}_{i=0}^{\infty}$ and

$$\hat{D}_{\theta,\theta'} = \frac{1}{m}\sum_{i=1}^{m} W_i,$$

then the CLT holds, subject to mild additional conditions specified in Kontoyiannis and Meyn (2003).

Coupling algorithm

The algorithm simulates an exact method and also an indicator variable $B_t \in \{0, 1\}, t = 1, 2, ...$ marking the times at which the approximate chain separates from the exact chain.

Algorithm 5 (Coupling algorithm: exact and arbitrary algorithm) *At state $\Theta_t = \theta$, simulate B_t and Θ_{t+1} as follows:*

1. *Simulate $\theta' \sim q(\theta, \cdot)$ and other quantities δ as needed.*
2. *Simulate $U_t \sim U(0, 1)$.*
   Evaluate exact-chain acceptance probability $\alpha_E(\theta, \theta')$.
   If $V_t \leq \alpha_P$ then set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.
3. *Evaluate arbitrary-chain acceptance probability $\alpha_A(\theta, \theta')$.*
   If
   $$\min(\alpha_A, \alpha_E) < V_t \leq \max(\alpha_A, \alpha_E)$$
   then set $B_t = 1$ and otherwise set $B_t = 0$.

The Θ_t chain targets π exactly, while the B_t marks separations.

Mean time to separation

We now give a bound on the mean time to separation, assuming that the chains start in equilibrium.

Suppose $\Theta_0 \sim \pi$, so that the B_t process is stationary. Let $T = \min\{t > 0; B_t = 1\}$ be the first separation time, and assume $\Pr(T < \infty) = 1$. We call $T|B_0 = 1$ the separation return-time. Let $\rho = \mathbb{E}(T|B_0 = 1)$ be the mean separation return-time. By Kac's Recurrence Theorem $\rho = 1/\Pr(B_0 = 1)$. Let

$$\mathbb{E}|\alpha_E - \alpha_A| = \int_{E^2 \times R} |\alpha_E(\theta, \theta'; \delta) - \alpha_A(\theta, \theta'; \delta)| \, g_m(\delta) d\delta Q(\theta, d\theta') \pi(d\theta).$$

so

$$\rho = \frac{1}{\mathbb{E}|\alpha_E - \alpha_A|}.$$

The separation time from the initialization $\Theta_0 = \theta_0$ is $\tau(\theta_0) = \mathbb{E}(T|\Theta_0 = \theta_0)$. Let $\tau = \int_E \tau(\theta)\pi(d\theta)$ give the mean separation time starting in equilibrium. The return time bounds the separation time by $2\tau \geq \rho$ as the separation time around a fixed time is length-biased.

Scaling of separation time

If the CLT holds for $\hat{D}_{\theta,\theta'}$, then it can be coupled to a normal estimator.

$$\begin{aligned}\hat{D}_{\theta,\theta'} &= D + \frac{\sigma}{\sqrt{m}}\frac{\hat{D}-D}{\sigma/\sqrt{m}} = D + \frac{\sigma}{\sqrt{m}}\Phi^{-1}(G_m(\hat{D}) + O(m^{-1/2})) \\ &= D + \frac{\sigma}{\sqrt{m}}\Phi^{-1}(G_m(\hat{D})) + O(1/m),\end{aligned}$$

where $\Phi^{-1}(G_m(\hat{D}))$ is a standard normal random variable.

Coupling the penalty method (exact) and naive algorithm shows that the mean separation time grows at least linearly with increasing m, since $|\alpha_P - \alpha_N| = O(1/m)$.

If we couple the naive algorithm to the standard algorithm, s-MCMC, with acceptance probability α_S, we find $\mathbb{E}|\alpha_S - \alpha_N| = O(1/\sqrt{m})$. The naive algorithm is therefore closer to the exact penalty method, in the 'distance' $\mathbb{E}|\alpha - \alpha_{\mathsf{N}}|$ than it is to the exact standard algorithm.

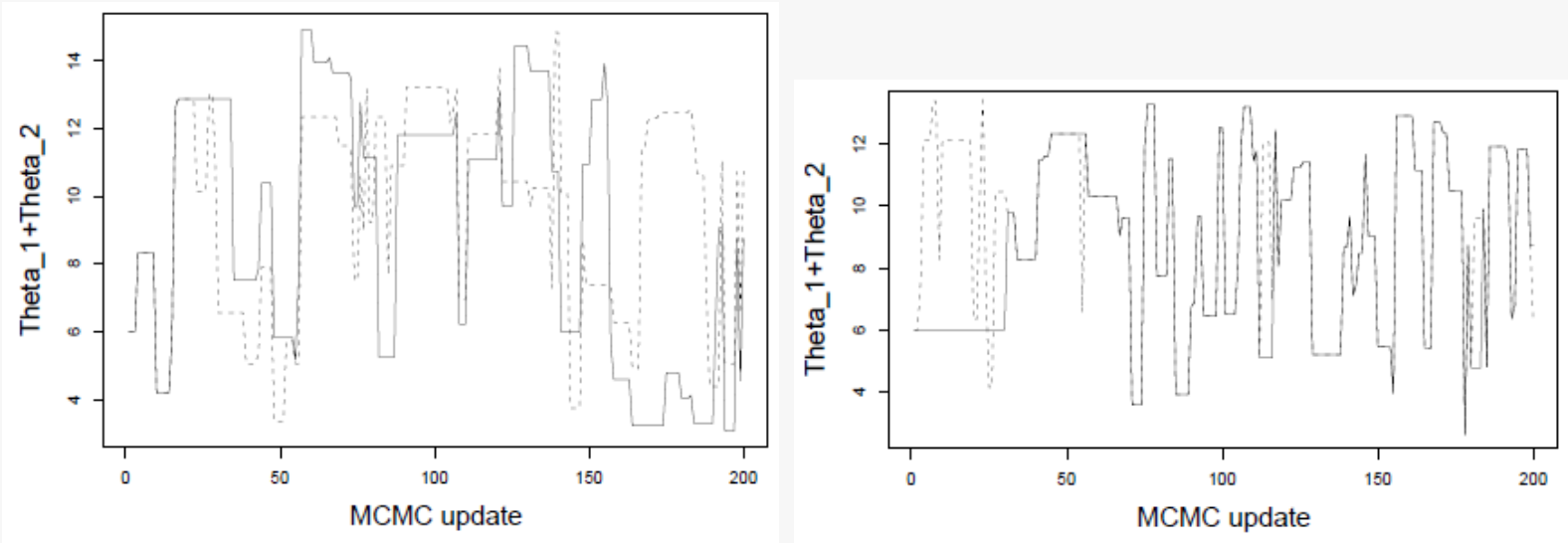
The penalty estimate method achieves separation times of $O(m^{3/2})$ at the price of stronger conditions on th eestimator $\hat{D}$.

Example

A very simple example for which we can compute $\Phi^{-1}(G_m(x))$

$$\pi(\theta) = p\, MVN(\theta; \mu_1, \Sigma_1) + (1-p) MVN(\theta; \mu_2, \Sigma_2),$$

$\mu_1 = (3,3)^T$, $\mu_2 = (6,6)^T$, $[\Sigma_a]_{i,i} = 1$ for $a, i = 1, 2$, and $[\Sigma_1]_{1,2} = 1/2$ and $[\Sigma_2]_{1,2} = -0.5$.



Simulations of the MCMC coupling-separation algorithm $\Theta_1 + \Theta_2$ in the penalty method (solid lines) and naive algorithm (dashed lines), (left) with Random-Walk Metropolis updates and (right) with independence-sampler updates. Target density is a mixture of bivariate normals, D-estimator using $m = 8$ samples at each update.

Separation time

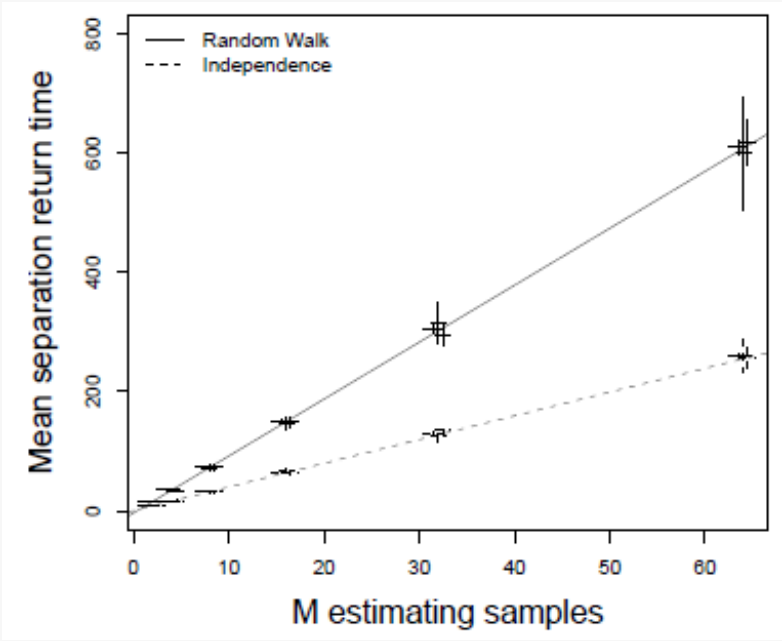
Two estimates of ρ are computed:

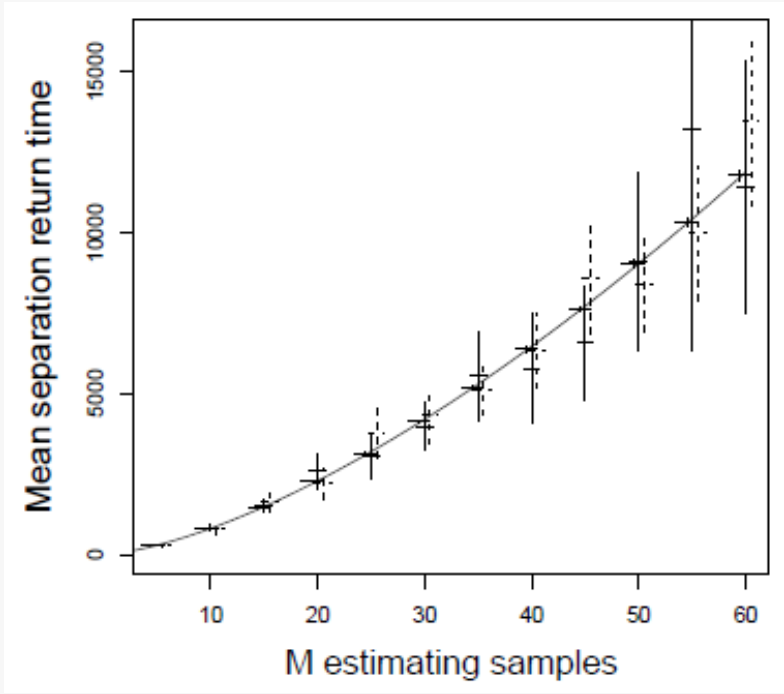
$$\hat{\rho}_1 = \frac{1}{K^{-1}\sum_{t=1}^{K} |\alpha_P(\theta_t, \theta_t'; y_t) - \alpha_N(\theta_t, \theta_t'; x_t)|}$$

and

$$\hat{\rho}_2 = S^{-1}\sum_{i=1}^{S}(T_i - T_{i-1})$$

where $T_i = \min\{t > T_{i-1}; B_t = 1\}$ with $T_0 = 0$ re the separation times in the coupling algorithm. The $\hat{\rho}_1$ estimator has lower variance than the $\hat{\rho}_2$ estimator. The τ-estimator is the mean of 1000 realizations of T_1.





(left) Estimated separation times ρ and τ between the exact Penalty Method and the approximate Naive Algorithm, as a function of estimator sample size m, for Random-Walk (solid lines) and Independence sampler (dashed lines) updates. Two estimates of ρ, $\hat{\rho}_1$ (left error bar in each group of three) and $\hat{\rho}_2$ (central error bar) and $\hat{\tau}$ (right error bar) are plotted for each sample and each m with a linear regression of the $\hat{\rho}_2$ estimates. (rght) Estimated separation times, as above, between Penalty Method and approximate Penalty Estimate chains regressed with $\hat{\rho}_1 = cm^{3/2}$.

Conclusions

phew!