12: Computationally Efficient Markov chain Simulation

November 11, 2019

Introduction

We mention:

1. an example where adding auxiliary variables increases computational efficiency
2. a few tuning tips for Random-Walk Metropolis-Hastings
3. Metropolis-adjusted Langevin Algorithm (MALA)
4. Hamiltonian Monte Carlo (HMC)
5. Pseudo-Marginal Metropolis-Hastings (PMMH).

Example: Data Augmentation

- $y_1, \ldots, y_n \mid \mu, \sigma^2 \overset{\text{iid}}{\sim} t_\nu(\mu, \sigma^2)$
- ν is assumed known
- $p(y_i \mid \mu, \sigma^2) \propto \left(1 + \frac{1}{\nu}\left(\frac{y_i - \mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$
- $p(\mu) \propto 1$
- $p(\sigma^2) \propto (\sigma^2)^{-1}$ (uniform for $\log \sigma$)

Example: Data Augmentation

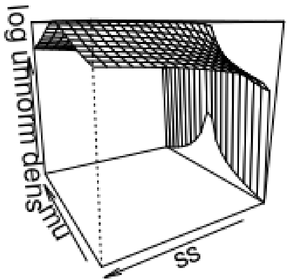
- $y_1, \ldots, y_n \mid \mu, \sigma^2 \overset{\text{iid}}{\sim} t_\nu(\mu, \sigma^2)$
- ν is assumed known
- $p(y_i \mid \mu, \sigma^2) \propto \left(1 + \frac{1}{\nu}\left(\frac{y_i - \mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$
- $p(\mu) \propto 1$
- $p(\sigma^2) \propto (\sigma^2)^{-1}$ (uniform for $\log \sigma$)

Normally we would do

$$p(\mu, \sigma \mid y) \propto p(y \mid \mu, \sigma^2) p(\mu) p(\sigma^2)$$

Example: Data Augmentation

Gibbs sampler not available :(



Data Augmentation: auxiliary variables

Instead, we introduce V_i (hidden/latent/unobserved data):

- $p(y_i \mid V_i, \mu, \sigma^2) \sim N(\mu, V_i)$
- $p(V_i \mid \sigma^2) \sim \text{Inv-}\chi^2(\nu, \sigma^2)$
- ν is assumed known still
- $p(\mu) \propto 1$ still
- $p(\sigma^2) \propto (\sigma^2)^{-1}$ (uniform for $\log \sigma$) still

$p(y_i \mid \mu, \sigma^2)$ is the same as before.

Data Augmentation: auxiliary variables

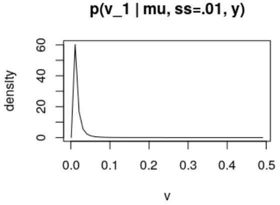
We can show that

1. $V_i \mid \mu, \sigma^2, y \sim \text{Inv-}\chi^2\left(\nu + 1, \frac{\nu\sigma^2 + (y_i - \mu)^2}{\nu + 1}\right)$
2. $\mu \mid \sigma^2, V_{1:n}, y \sim \text{Normal}\left(\frac{\sum_i \frac{1}{V_i} y_i}{\sum_i \frac{1}{V_i}}, \frac{1}{\sum_i \frac{1}{V_i}}\right)$
3. $\sigma^2 \mid \mu, V_{1:n}, y \sim \text{Gamma}\left(\frac{n\nu}{2}, \frac{\nu}{2} \sum_i \frac{1}{V_i}\right)$

Data Augmentation: auxiliary variables

Note:

$$V_i \mid \mu, \sigma^2, y \sim \text{Inv-}\chi^2\left(\nu+1, \frac{\nu\sigma^2 + (y_i-\mu)^2}{\nu+1}\right)$$
$$= \text{Inv-Gamma}\left(\frac{\nu+1}{2}, \frac{\nu\sigma^2 + (y_i-\mu)^2}{2}\right)$$



(see `t_visualization.r`)

Near-zero values of V_is lead to σ^2 being near zero, too.

Data Augmentation: parameter expansion

Add another parameter: $\alpha > 0$

Rename a few things:

$$\tau^2 = \sigma^2/\alpha^2 \tag{1}$$

$$U_i = V_i/\alpha^2 \tag{2}$$

Assume a noninformative prior for α:

$$p(\alpha^2) \propto (\alpha^2)^{-1}$$

Data Augmentation: parameter expansion

- $p(y_i \mid V_i, \mu, \sigma^2) \sim N(\mu, \alpha^2 U_i)$
- $p(U_i \mid \tau^2) \sim \text{Inv-}\chi^2(\nu, \tau^2)$
- ν is assumed known
- $p(\mu) \propto 1$
- $p(\tau^2) \propto (\tau^2)^{-1}$ (uniform for $\log \tau$)

Prove that the model is not identifiable in the full parameter space!
However, the inference about μ, $\alpha\tau$, and $\alpha^2 U_i$ is still valid.

Data Augmentation: parameter expansion

Posterior conditional distributions are similar:

1. $U_i \mid \alpha, \mu, \tau^2, y \sim \text{Inv-}\chi^2\left(\nu + 1, \frac{\nu\tau^2 + ((y_i - \mu)/\alpha)^2}{\nu+1}\right)$
2. $\mu \mid \alpha, \tau^2, U_{1:n}, y \sim \text{Normal}\left(\frac{\sum_i \frac{1}{\alpha^2 U_i} y_i}{\sum_i \frac{1}{\alpha^2 U_i}}, \frac{1}{\sum_i \frac{1}{\alpha^2 U_i}}\right)$
3. $\tau^2 \mid \mu, U_{1:n}, y \sim \text{Gamma}\left(\frac{n\nu}{2}, \frac{\nu}{2} \sum_i \frac{1}{\alpha^2 U_i}\right)$
4. $\alpha \mid \mu, \tau^2, U_{1:n}, y \sim \text{Inv-}\chi^2\left(n, \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \mu_i)^2}{U_i}\right)$

α breaks the dependence between $V_i = \alpha^2 U_i$ and τ^2.

Random Walk M-H: Some Tricks

Last chapter, when we were using the Metropolis-Hastings algorithm, we need to specify the proposal's covariance matrix:
If θ is roughly normal,

$$q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \Sigma).$$

The book recommends setting

$$\Sigma \approx \frac{2.4^2}{d} \text{Var}(\theta \mid y).$$

Here d is the dimension of θ. A rough approximation of the posterior covariance matrix is required, e.g., Hession matrix at the posterior mode.

Random Walk M-H: Some Tricks

Why set the proposal covariance matrix this way?

It is all about the efficiency of the posterior samples.

Specifically, our goal is to increase the **rate** that a new independent θ being generated.

It can be shown that under the suggested proposal, $q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \frac{2.4^2}{d}\text{Var}(\theta \mid y))$, the efficiency is $0.3/d$, meaning that, on average, every $d/0.3$ iterations a new independent θ is drawn.

The efficiency is low when the dimension d is large!!!

Adaptive Metropolis-Hasting algorithm

Adaptive MH is aimed to improve the acceptance rate:

Initial stage:

- Start simulations with a fixed MH algorithm using proposals like $q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \frac{2.4^2}{d}\text{Var}(\theta \mid y))$

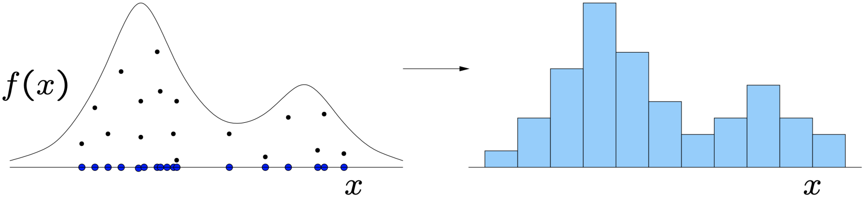
Adaptive stage:

- Update the jumping rule as $q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \Sigma)$, where Σ is estimated from the simulation in initial stage.
- (Optional, needs parallel simulations) Adjust the scale of the jumping distribution until an acceptance rate of 0.44 in one dimension or 0.23 when parameters are updated as a vector.

Note: if Σ is the same as the target distribution, the jumping rule, $q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \frac{2.4^2}{d}\Sigma)$, has acceptance rate 0.44 in one dimension or 0.23 when parameters are updated as a vector.

Slice sampling

To sample from a distribution, simply sample uniformly from the region under the density function and consider only the horizontal coordinates.



Slice sampling

One way to do this is

- Introduce latent (auxiliary) variables
- Use Gibbs sampling on the area beneath the density

Suppose we wish to sample from $f(x)$, it is equivalent to:

- $y \mid x \sim \text{Uniform}(0, f(x))$, then $f(x, y)$ is constant over $\{(x, y) : 0 \leq y \leq f(x)\}$.
- $x \mid y \propto f(x, y) \sim \text{Uniform}(S(y))$, where $S(y) = \{x : y \leq f(x)\}$

Slice sampling

This leads to an iterative algorithm:

- $y_i \mid x_{i-1} \sim \text{Uniform}(0, f(x_{i-1}))$
- $x_i \mid y_i \sim \text{Uniform}(S(y_i))$

No need to specify proposal distributions as needed by MH or rejection sampling.

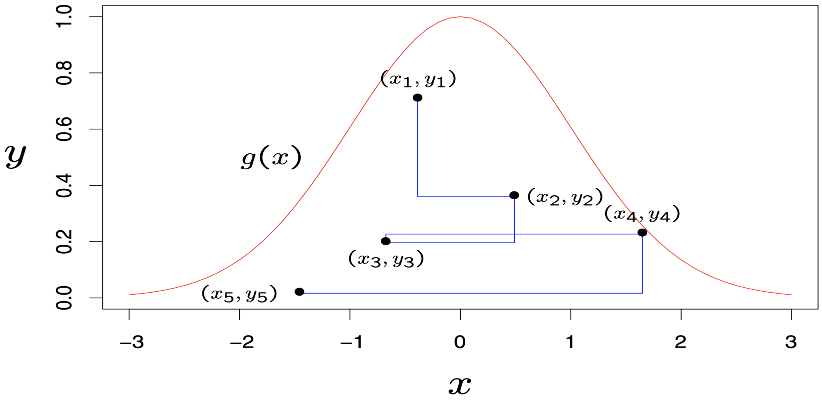
Determining the slice $S(y)$ can be tricky!

Slice sampling: example for normal distribution

Suppose $x \sim \text{Normal}(0, 1)$, so $f(x) \propto g(x) = \exp(-x^2/2)$, then the slice through the density is

$$S(y) = \{x : -\sqrt{-2\log(y)} \leq x \leq \sqrt{-2\log(y)}\}.$$

First five iterations of the slice sampler for the standard normal example:



Sampling from multi-modal distribution: simulated tempering

Let $p(\theta)$ be the target (unnormalized) density. Consider

$$q_k(\theta) \propto p(\theta)^{1/T_k}$$

for a set of "temperature" parameters $T_k > 0$, $k = 0, 1, \ldots, K$, where $T_0 = 1$ and $q_0(\theta) = p(\theta)$.

For T_k large (high temperature), the density q_k will be more flat than q_0. Simulated tempering constructs a Markov chain with augmented state (θ^t, s^t) at time t with s^t an integer indicating the current temperature.

Sampling from multi-modal distribution: simulated tempering

The algorithm is an Metropolis-Hasting algorithm leading to a composite Markov chain:

- Propose a new state for θ^t: θ^t is then generated using Markov chain simulation corresponding to stationary distribution $q_{s^{t-1}}$.
- An MH step for s^t:
  proposing s^t according to $P(s^t = k) \propto J_{s^{t-1},k}$
  accept with probability $\min(r, 1)$, where

$$r = \frac{c_k q_k(\theta^t) J_{ks^{t-1}}}{c_{s^{t-1}} q_{s^{t-1}}(\theta^t) J_{s^{t-1}k}}$$

  c_k is the normalizing constant for q_k.

Once the composite Markov chain is simulated for $t = 1, \ldots, T$, only θ^t corresponding to $s^t = 0$ are kept for inference about q_0.