FIREFLY MONTE CARLO: EXACT MCMC WITH SUBSETS OF DATA

MCMC

We have a unnormalized distribution from which we want to sample

- Metropolis-Hastings algorithm for MCMC

With probability $p = \min\left[1, \exp\left(\frac{U(\mathbf{x}_n) - U(\mathbf{x}'_n)}{T}\right)\right]$ system accepts new state (jumps to the new state): $\mathbf{x}_{n+1} = \mathbf{x}'_n$ and with probability $1 - p$ new state is rejected: $\mathbf{x}_{n+1} = \mathbf{x}_n$.

- Hamiltonian Monte Carlo

$$E(\mathbf{x}, \mathbf{v}) = U(\mathbf{x}) + K(\mathbf{v}), \qquad K(\mathbf{v}) = \sum_i \frac{m\, v_i^2}{2}$$

Thus, velocities $\mathbf{v}$ and positions $\mathbf{x}$ have **independent** canonical distributions:

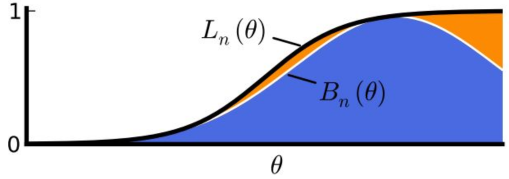
$$p(\mathbf{x}, \mathbf{v}) \propto \exp\left(\frac{-E(\mathbf{x}, \mathbf{v})}{T}\right) = \exp\left(\frac{-U(\mathbf{x})}{T}\right) \exp\left(\frac{-K(\mathbf{v})}{T}\right) \propto p(\mathbf{x})\, p(\mathbf{v}).$$

MCMC problem

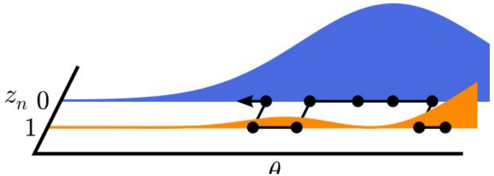
MCMC cannot be practically applied to large data sets because of the prohibitive cost of evaluating every likelihood term at every iteration.

Solution

$$p(z_n \mid x_n, \theta) = \left[\frac{L_n(\theta) - B_n(\theta)}{L_n(\theta)}\right]^{z_n} \left[\frac{B_n(\theta)}{L_n(\theta)}\right]^{1-z_n}.$$



$$\begin{aligned}
&p(x_n \mid \theta)p(z_n \mid x_n, \theta) \\
&= L_n(\theta) \left[\frac{L_n(\theta) - B_n(\theta)}{L_n(\theta)}\right]^{z_n} \left[\frac{B_n(\theta)}{L_n(\theta)}\right]^{1-z_n} \\
&= \begin{cases} L_n(\theta) - B_n(\theta) & \text{if } z_n = 1 \\ B_n(\theta) & \text{if } z_n = 0 \end{cases}.
\end{aligned}$$



For each data point, n, we introduce a binary auxiliary variable, $z_n \in \{0, 1\}$, and a function $B_n(\theta)$ which is a sctrictly positive lower bound on the nth likelihood: $0 < B_n(\theta) \leq L_n(\theta)$. Each z_n has the following Bernoulli distribution conditioned on the parameters:

Algorithm 1 Firefly Monte Carlo *Note: Using simple random-walk MH for clarity.*

$\theta_0 \sim$ InitialDist ▷ Initialize the Markov chain state.
for $i \leftarrow 1 \ldots$ Iters **do** ▷ Iterate the Markov chain.
  for $j \leftarrow 1 \ldots \lceil N \times \text{ResampleFraction} \rceil$ **do**
    $n \sim \text{RandInteger}(1, N)$ ▷ Select a random data point.
    $z_n \sim \text{Bernoulli}(1 - B_n(\theta_{i-1})/L_n(\theta_{i-1}))$ ▷ Biased coin-flip to determine whether n is bright or dark.
  end for
  $\theta' \leftarrow \theta_{i-1} + \eta$ where $\eta \sim \text{Normal}(0, \epsilon^2 \mathbb{I}_D)$ ▷ Make a random walk proposal with step size ϵ.
  $u \sim \text{Uniform}(0, 1)$ ▷ Draw the MH threshold.
  if $\dfrac{\text{JointPosterior}(\theta' ; \{z_n\}_{n=1}^N)}{\text{JointPosterior}(\theta ; \{z_n\}_{n=1}^N)} > u$ **then** ▷ Evaluate MH ratio conditioned on auxiliary variables.
    $\theta_i \leftarrow \theta'$ ▷ Accept proposal.
  else
    $\theta_i \leftarrow \theta_{i-1}$ ▷ Reject proposal and keep current state.
  end if
end for

function JointPosterior$(\theta ; \{z_n\}_{n=1}^N)$ ▷ Modified posterior that conditions on auxiliary variables.
  $P \leftarrow p(\theta) \times \prod_{n=1}^N B_n(\theta)$ ▷ Evaluate prior and bounds. Collapse of bound product not shown.
  for each n **for which** $z_n = 1$ **do** ▷ Loop over bright data only.
    $P \leftarrow P \times (L_n(\theta)/B_n(\theta) - 1)$ ▷ Include bound-corrected factor.
  end for
  return P
end function

In the case of tight bound, explicit sampling doesn't effective. So the author's propose the following solution

Algorithm 2 Implicit z_n sampling

for $n \leftarrow 1 \ldots N$ **do** ▷ Loop over all the auxiliary variables.
  if $z_n = 1$ **then** ▷ If currently bright, propose going dark.
    $u \sim \text{Uniform}(0, 1)$ ▷ Sample the MH threshold.
    if $\frac{q_{d \to b}}{\tilde{L}_n(\theta)} > u$ **then** ▷ Compute MH ratio with $\tilde{L}_n(\theta)$ cached from θ update.
      $z_n \leftarrow 0$ ▷ Flip from bright to dark.
    end if
  else ▷ Already dark, consider proposing to go bright.
    if $v < q_{d \to b}$ **where** $v \sim \text{Uniform}(0, 1)$ **then** ▷ Flip a biased coin with probability $q_{d \to b}$.
      $u \sim \text{Uniform}(0, 1)$ ▷ Sample the MH threshold.
      if $\frac{\tilde{L}_n(\theta)}{q_{d \to b}} < u$ **then** ▷ Compute MH ratio.
        $z_n \leftarrow 1$ ▷ Flip from dark to bright.
      end if
    end if
  end if
end for

Results

		Algorithm	Average Likelihood queries per iteration	Effective Samples per 1000 iterations	Speedup relative to regular MCMC
Data set:	MNIST	Regular MCMC	12,214	3.7	(1)
Model:	Logistic regression	Untuned FlyMC	6,252	1.3	0.7
Updates:	Metropolis-Hastings	MAP-tuned FlyMC	207	1.4	22
Data set:	3-Class CIFAR-10	Regular MCMC	18,000	8.0	(1)
Model:	Softmax classifcation	Untuned FlyMC	8,058	4.2	1.2
Updates:	Langevin	MAP-tuned FlyMC	654	3.3	11
Data set:	OPV	Regular MCMC	18,182,764	1.3	(1)
Model:	Robust regression	Untuned FlyMC	2,753,428	1.1	5.7
Updates:	Slice sampling	MAP-tuned FlyMC	575,528	1.2	29

TABLE 1

Results from empirical evaluations. Three experiments are shown: logistic regression applied to MNIST digit classification, softmax classification for three categories of CIFAR-10, and robust regression for properties of organic photovoltaic molecules, sampled with random-walk Metropolis–Hastings, Langevin-adjusted Metropolis, and slice sampling, respectively. For each of these, the vanilla MCMC operator was compared with both untuned FlyMC and FlyMC where the bound was determined from a MAP estimate of the posterior parameters. We use likelihood evaluations as an implementation-independent measure of computational cost and report the number of such evaluations per iteration, as well as the resulting sample efficiency (computed via R-CODA (Plummer et al., 2006)), and relative speedup.