JAE. Bayes in AI
2. Intro Markov chain Monte Carlo

David Ríos Insua and Roi Naveiro

david.rios@icmat.es roi.naveiro@icmat.es

Brief description

The intro presented key methods in Bayesian inference and basic models and faced ‘severe’ computational problems quite rapidly

We introduce core computational strategies to deal with those problems.

Here intro to Markov chain Monte Carlo (MCMC) strategies

- Gibbs sampling
- Metropolis-Hastings
- Hamiltonian Monte Carlo

Sources

French and Rios Insua (2000) Ch 7

Rios Insua et al (2010) Ch4

BDA3 (2015) Ch11, 12

Hoff (2009) Ch 6,7

Computational problems in Bayesian analysis

Computing the posterior

Computing the predictive

Finding the optimal alternative

$$f(\theta|x) = \frac{f(x|\theta)\, f(\theta)}{f(x)} = \frac{f(x|\theta)\, f(\theta)}{\int f(x|\theta)\, f(\theta)\, d\theta} \propto f(x|\theta)\, f(\theta)$$

$$f(y|x) = \int f(y|\theta)\, f(\theta|x)\, d\theta$$

$$\max_a \int u(a,\theta)\, f(\theta|x)\, d\theta$$

$$\max_a \int u(a,y)\, f(y|x)\, dy$$

Strategies so far

- Conjugate models
- (Posterior asymptotics to normality)
- (Laplace integration)

Insufficient for modern stats and machine learning!!

Numerical and MC integration

Numerical integration. Brief recall

Problem

$$I_s = \int_{[0,1]^s} f(u)\, du$$

s-dimensional trapezium rule

$$I_s \simeq \sum_{n_1=0}^{m} \cdots \sum_{n_s=0}^{m} w_{n_1} \cdots w_{n_s}\, f\left(\frac{n_1}{m}, \ldots, \frac{n_s}{m}\right)$$

$$w_0 = w_m = 1/2m \qquad w_n = 1/m \quad 1 \le n \le m-1$$

error analysis

$$\frac{\partial^2 f}{\partial u_i^2} \text{ CONT IN } [0,1]^s \longrightarrow O(m^{-2})$$

$$N = (m+1)^s \qquad \longrightarrow O(N^{-2/s})$$

$$s = 5,\ \varepsilon \le 10^{-2} \Longrightarrow N = 10^5 \text{ !!! } \ldots$$

Dependence of error bound on dimension is typical!!!!

Monte Carlo integration. Brief recall

Problem

$$I_s = \int_{[0,1]^s} f(u)\, du$$

Deterministic problem recast as stochatic (Monte Carlo)

$$I_s = E(f)$$

Monte Carlo integration. Brief recall

Suggested strategy

Sample $u_1, \dots, u_N \sim \mathcal{U}[0,1]^s$

Do $\hat{I}_s = \frac{1}{N} \sum_{i=1}^{N} f(u_i)$

Monte Carlo integration. Brief recall

Analysis. SLLN

$$\hat{I}_s = \frac{1}{N}\sum_{i=1}^{N} f(u_i) \xrightarrow{a.s.} E(f) = \boxed{I_s}$$

Error bounds

$$\int_{[0,1]^s} (\hat{I}_s - I_s)^2 \, du = \frac{\sigma^2(f)}{N} = Var(\hat{I}_s)$$

$$\sigma^2(f) = \int_{[0,1]^s} (f - E(f))^2 \, du \qquad \boxed{\int_{[0,1]^s} f^2(u)\, du}$$

CLT prob. error bounds

$$Pr\left(\frac{c_1 \sigma(f)}{\sqrt{N}} \le \hat{I}_s - I_s \le \frac{c_2 \sigma(f)}{\sqrt{N}}\right) \xrightarrow[N]{} \Phi(c_2) - \Phi(c_1)$$

SE

$$EE(\hat{I}_s) = \frac{1}{\sqrt{N}} \sqrt{\frac{\sum (f(u_i) - \hat{I}_s)^2}{N-1}}$$

MC vs trapezium

$$O\left(N^{-\frac{1}{2}}\right) \quad \text{vs} \quad O\left(N^{-\frac{2}{s}}\right)$$

This is general. As dimension grows, numerical gets less efficient… but MC's efficiency is dimension independent!!!

Markov chain Monte Carlo intro

MC. Generalization

Problem

$$I_g = \int f(u)\, g(u)\, du = E_g(f)$$

Strategy

Sample $u_1, \dots, u_N \sim g$

Do $\hat{I}_g = \frac{1}{N} \sum_{i=1}^{N} f(u_i)$

General idea

Objective

$$I_g = \int f(x)\, g(x)\, dx = E_g(f)$$

Difficult or inefficient to sample from g

General idea

Markov chain X_n with same state space and convergent to target distribution g

$$X_n \xrightarrow{d} g$$

Strategy

Initialise x_0, $n=1$

Until convergence, Generate $X_n \mid X_{n-1}=x_{n-1}$, $n=n+1$, n^*

Until n^*+m, Generate and collect $X_n \mid X_{n-1}=x_{n-1}$, $n=n+1$

$$\hat{I}_g \approx \frac{1}{m} \sum_{n^*}^{n^*+m} f(x_i)$$

Problem

So how do we ‘invent’ such Markov chains?

Motivating Gibbs sampler

(X,Y) Bernoulli variables with joint distribution

X	Y	$P(X,Y)$
0	0	p_1
1	0	p_2
0	1	p_3
1	1	p_4

Compute the marginals of X and Y

Compute the conditionals

Motivating Gibbs sampler

The conditionals are characterised by

$$A_{yx} = \begin{pmatrix} P(Y=0|X=0) & P(Y=1|X=0) \\ P(Y=0|X=1) & P(Y=1|X=1) \end{pmatrix} = \begin{pmatrix} \frac{p_1}{p_1+p_3} & \frac{p_3}{p_1+p_3} \\ \frac{p_2}{p_2+p_4} & \frac{p_4}{p_2+p_4} \end{pmatrix}$$

$$A_{xy} = \begin{pmatrix} \frac{p_1}{p_1+p_2} & \frac{p_2}{p_1+p_2} \\ \frac{p_3}{p_3+p_4} & \frac{p_4}{p_3+p_4} \end{pmatrix}$$

Motivating Gibbs sampler

Consider the sampling scheme

Initialise $Y_0 = y_0$, $i = 1$

Do

  Sample $X_i \sim X \mid Y = y_{i-1}$

  Sample $Y_i \sim Y \mid X = x_i$

  $i = i + 1$

Motivating Gibbs sampler

X_n a Markov chain with transition matrix

$$A = A_{yx} A_{xy}$$

Motivating Gibbs sampler

Convergence of X_n

$$X_n \xrightarrow{d} X$$

$$(p_1+p_3 \quad p_2+p_4) = (p_1+p_3 \quad p_2+p_4)\,A$$

Similarly,

$$Y_n \xrightarrow{d} Y \qquad (X_n, Y_n) \xrightarrow{d} (X, Y)$$

Recall

1. Choose initial values $(\theta_2^0, \ldots, \theta_k^0)$. $i = 1$

2. Until convergence is detected, iterate through

. Generate $\theta_1^i \sim \theta_1 | \theta_2^{i-1}, .., \theta_k^{i-1}$

. Generate $\theta_2^i \sim \theta_2 | \theta_1^i, \theta_3^{i-1}, ..., \theta_k^{i-1}$

. ...

. Generate $\theta_k^i \sim \theta_k | \theta_1^i, ..., \theta_{k-1}^i$.

. $i = i + 1$

Convergence

Kernel $$p_G(\theta^n, \theta^{n+1}) = \prod_{i=1}^{k} p(\theta_i^{n+1} \mid \theta_j^n, j > i; \theta_j^{n+1}, j < i).$$

Proposition 43 *Suppose that* $D = \{\theta : p(\theta) > 0\}$ *is a product set,* $D = \prod_{i=1}^{k} D_i$. *Then:*

1. $p_{\theta_i}(\theta_i | \theta_j, j \neq i)$ *and* p_G *are well-defined for* $\theta \in D$.
2. p_G *is* p*-irreducible and aperiodic.*
3. p *is invariant with respect to* p_G.
4. $\theta^n \xrightarrow{w} \theta$

Motivating Gibbs sampler

Example

$$\pi(x_1, x_2) = \frac{1}{\pi} e^{-x_1(1+x_2^2)} \qquad (x_1, x_2) \in (0, \infty) \times (-\infty, \infty)$$

Motivating Gibbs sampler

Example

$$\pi(x_1, x_2) = \frac{1}{\pi} e^{-x_1(1+x_2^2)} \qquad (x_1, x_2) \in (0, \infty) \times (-\infty, \infty)$$

$$\pi(x_1|x_2) = \frac{\pi(x_1, x_2)}{\pi(x_2)} \propto \pi(x_1, x_2) \propto e^{-x_1(1+x_2^2)} \qquad X_1|X_2 = x_2 \sim \mathcal{E}xp(1 + x_2^2)$$

$$\pi(x_2|x_1) \propto \pi(x_1, x_2) \propto e^{-x_1 x_2^2}, \qquad X_2|X_1 = x_1 \sim \mathcal{N}\left(0, \sigma^2 = \frac{1}{2x_1}\right)$$

Motivating Gibbs sampler

Example

Choose initial x_2^0, $i = 1$

Until convergence

Sample $x_1^i \sim Exp(1 + (x_2^{i-1})^2)$

Sample $x_2^i \sim N\left(0, \frac{1}{2x_1^i}\right)$

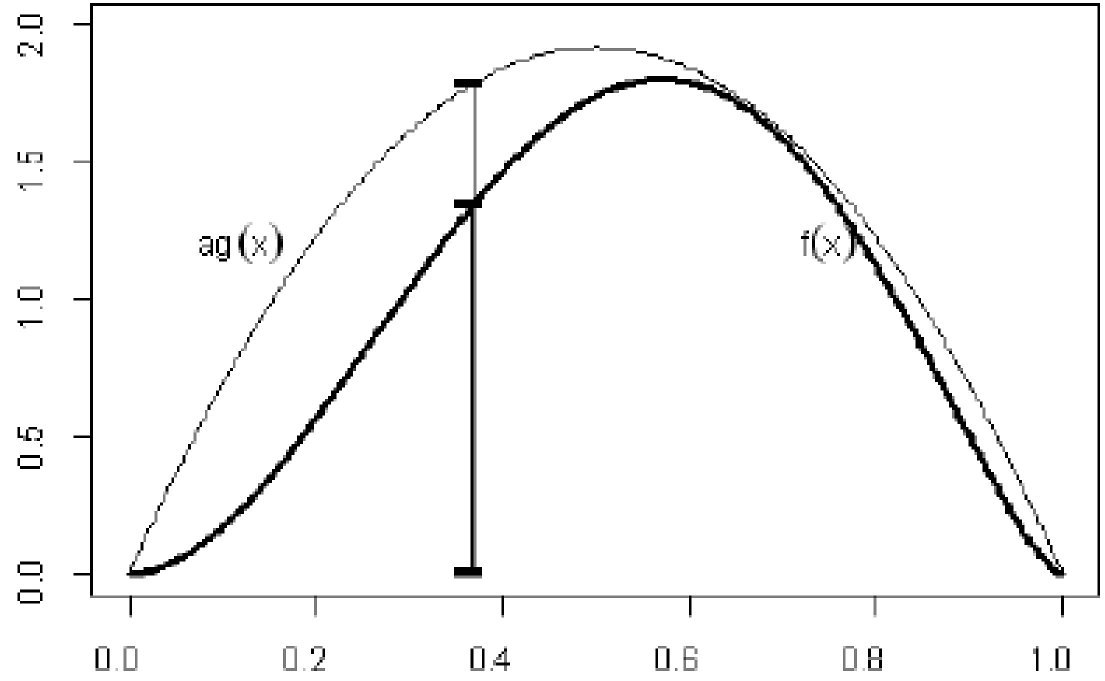
$i = i + 1$

Metropolis-Hastings algorithm

Recall: Acceptance-rejection sampling

TARGET DENSITY $\pi(x) = f(x)/K$, K POSSIBLY UNKNOWN

SAMPLING DENSITY $g(x)$: $f(x) \leq a g(x)$, $\forall x$



WHILE $U > f(x) / (a g(x))$

GENERATE $X \sim g$, $U \sim \mathcal{U}[0,1]$

OUTPUT X

$P(X \leq x \mid X \text{ ACCEPTED}) = F(x)$ (F cdf of X)

Metropolis-Hastings rationale I

Transition kernel of Markov chain $P(x,A)$

Invariant distribution $\pi^*(dy) = \int P(x,dy)\,\pi(x)\,dx$

n-th iterate

$$P^{(n)}(x,A) = \int P^{(n-1)}(x,dy)\,P(y,A)$$

Metropolis-Hastings rationale II

Invariant distribution and reversibility. Suppose that for p kernel (x,y)

$$P(x, dy) = p(x,y)\, dy + r(x)\, \delta_x(dy)$$

$$p(x,x) = 0 \qquad \delta_x(dy) = \begin{cases} 1, & \text{if } x \in dy \\ 0, & \text{OTHERWISE} \end{cases}$$

$$r(x) = 1 - \int p(x,y)\, dy$$

$$\pi(x)\, p(x,y) = \pi(y)\, p(y,x) \Rightarrow \pi \text{ INVARIANT FOR } P(y, \cdot)$$

Metropolis-Hastings rationale III

Adjusting a candidate generating distribution

$q(x,y)$ CANDIDATE GENERATING DENSITY

SUPPOSE

$$\pi(x)\, q(x,y) > \pi(y)\, q(y,x)$$

INTRODUCE 'CORRECTION' PROBABILITY OF MOVE

$$\alpha(x,y) < 1$$

$$p_{MH}(x,y) \equiv q(x,y)\, \alpha(x,y) \qquad x \neq y$$

$$\pi(x)\, q(x,y)\, \alpha(x,y) = \pi(y)\, q(y,x)\, \alpha(y,x)$$

$$= \pi(y)\, q(y,x)$$

$$\Longrightarrow \alpha(x,y) = \frac{\pi(y)\, q(y,x)}{\pi(x)\, q(x,y)}$$

Metropolis-Hastings rationale IV

Balance condition

$$\alpha(x,y) = \begin{cases} \min\left(\frac{\pi(y)\, q(y,x)}{\pi(x)\, q(x,y)}, 1\right), & \text{IF } \pi(x)\, q(x,y) > 0 \\ 1, & \text{OTHERWISE} \end{cases}$$

Observations

- Normalising constant not required
- If q symmetric, Metropolis

$$\alpha(x,y) = \min\left(\frac{\pi(y)}{\pi(x)}, 1\right)$$

Metropolis-Hastings algo

1. Choose initial values θ^0. $i = 0$

2. Until convergence is detected, iterate through

   . Generate a candidate $\theta^* \sim q(\theta|\theta^i)$.

   . If $p_\theta(\theta^i) q(\theta^i \mid \theta^*) > 0$, $\alpha(\theta^i, \theta^*) = \min\left(\frac{p_\theta(\theta^*) q(\theta^* | \theta^i)}{p_\theta(\theta^i) q(\theta^i | \theta^*)}, 1\right)$;

   . else, $\alpha(\theta^i, \theta^*) = 1$.

   . Do

$$\theta^{i+1} = \begin{cases} \theta^* & \text{with prob } \alpha(\theta^i, \theta^*), \\ \theta^i & \text{with prob } 1 - \alpha(\theta^i, \theta^*) \end{cases}$$

   . $i = i + 1$.

Metropolis-Hastings variants I

Random walk chain

Metropolis algorithm

$$q(x,y) = q_1(x-y)$$

$$y = x + z, \quad z \sim q_1 \quad \to \text{NORMAL} \quad \to t$$

$$\alpha(x,y) = \min\left(\frac{\pi(y)}{\pi(x)}, 1\right)$$

Independence chain

$$q(x,y) = q_2(y) \quad \to \text{NORMAL} \quad \to t$$

Convergence

Kernel $p_{MH}(\theta^n, \theta^{n+1}) = \; q(\theta^n, \theta^{n+1})\alpha(\theta^n, \theta^{n+1})$, if $\theta^n \neq \theta^{n+1}$, and $1 - \int q(\theta^n, z)\alpha(\theta^n, z)dz$, otherwise.

Proposition 44 *The following hold:*

1. *If q is aperiodic, p_{MH} is aperiodic.*
2. *If q is p_{MH}-irreducible and $q(\theta^n, \theta^{n+1}) = 0$ iff $q(\theta^{n+1}, \theta^n) = 0$, p_{MH} is p_θ irreducible.*

Hamiltonian MC. Basics

HMC. Pros and cons

- Improved computational efficiency over MH et al (specially in high dimensional complex problems)

- Difficulties in implementation…. But Stan is available now: automates tuning of HMC parameters (and can be called from R and Python)
- But Stan a bit of a black box

- Still insufficient for Bayesian analysis of deep learning models… Lect 3

The drawback of MH

Balance condition in MH and M algos. Currentx, proposed y

$$\alpha(x,y) = \begin{cases} \min\left(\frac{\pi(y)\,q(y,x)}{\pi(x)\,q(x,y)}, 1\right), & \text{IF } \pi(x)\,q(x,y)>0 \\ 1, & \text{OTHERWISE} \end{cases}$$

$$\alpha(x,y) = \min\left(\frac{\pi(y)}{\pi(x)}, 1\right)$$

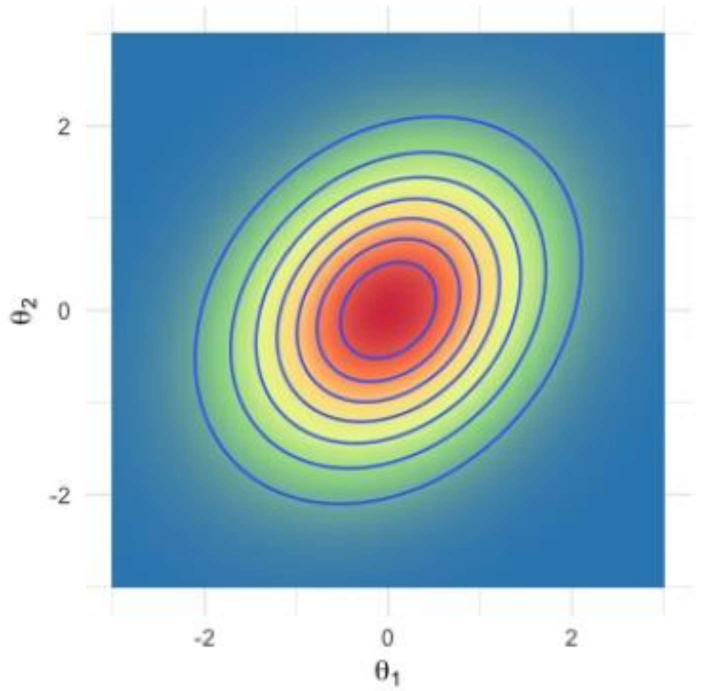
Frequents regions of higher posterior density. Sample from the right region

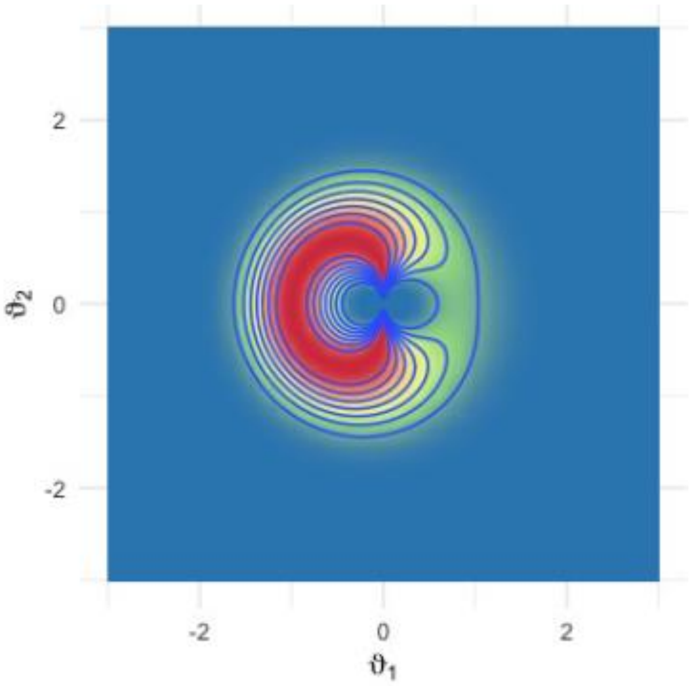
Ocasionally visits low density regions. Fully explore the sample space

As proposals are random, may take quite some time to get in HPD regions

May get stuck

The drawback of MH





HMC. Qualitative description

- A guided proposal generation scheme

- Uses the gradient of log posterior to direct MC towards HPD regions:

A well-tuned HMC accepts proposals at much higher rate than MH

- But still samples the tails properly

HMC. Idea

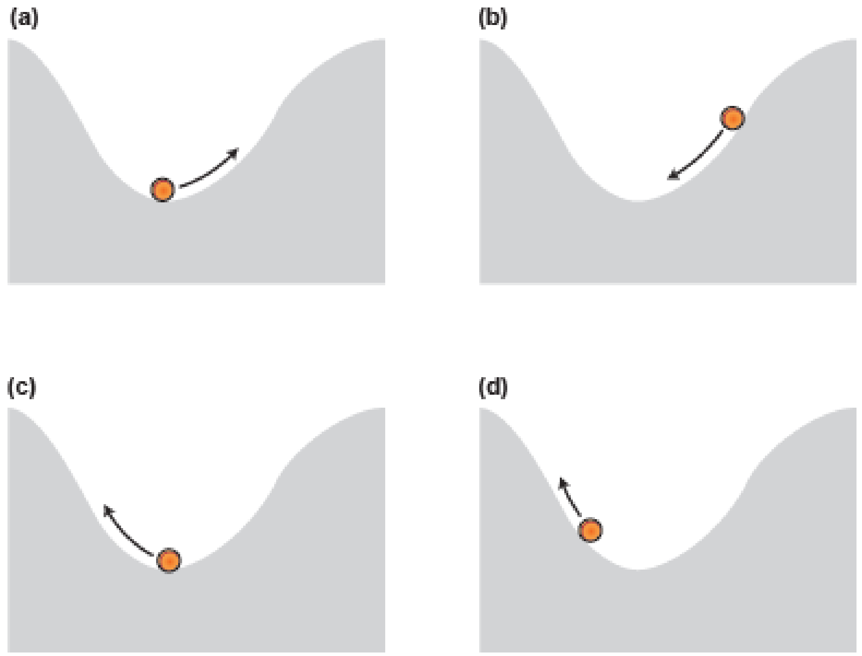
f is the posterior

-log (f) inverse bell-shaped

lower values reached guided by its gradient

In classical mechanics, exchanges between kinetic and potential energy dictate location through hamiltonian equations

(θ, p) horizontal and vertical positions. p is a momentum (auxiliary variable to actually simulate from θ) mass x velocity



Hamiltonian equations and MCMC

Target is posterior. Auxiliary momentum (same dimension)

$\theta \sim f(\theta) \qquad \dim(\theta) = \dim(p)$

Hamiltonian as potential + kinetic

$U(\theta) = -\log f(\theta) \qquad p \sim N(0, M)$

$H(\theta, p) = U(\theta) + K(p)$

$H(\theta, p) = -\log f(\theta) + \frac{1}{2} p^T M^{-1} p$

Hamiltonian equations

$$\left.\begin{aligned} \frac{dp}{dt} &= -\frac{\partial H(\theta, p)}{\partial \theta} = \nabla_\theta \log(f(\theta)) \\ \frac{d\theta}{dt} &= \frac{\partial H(\theta, p)}{\partial p} = M^{-1} p \end{aligned}\right] (\theta, p)$$

Hamiltonian equations through leapfrog

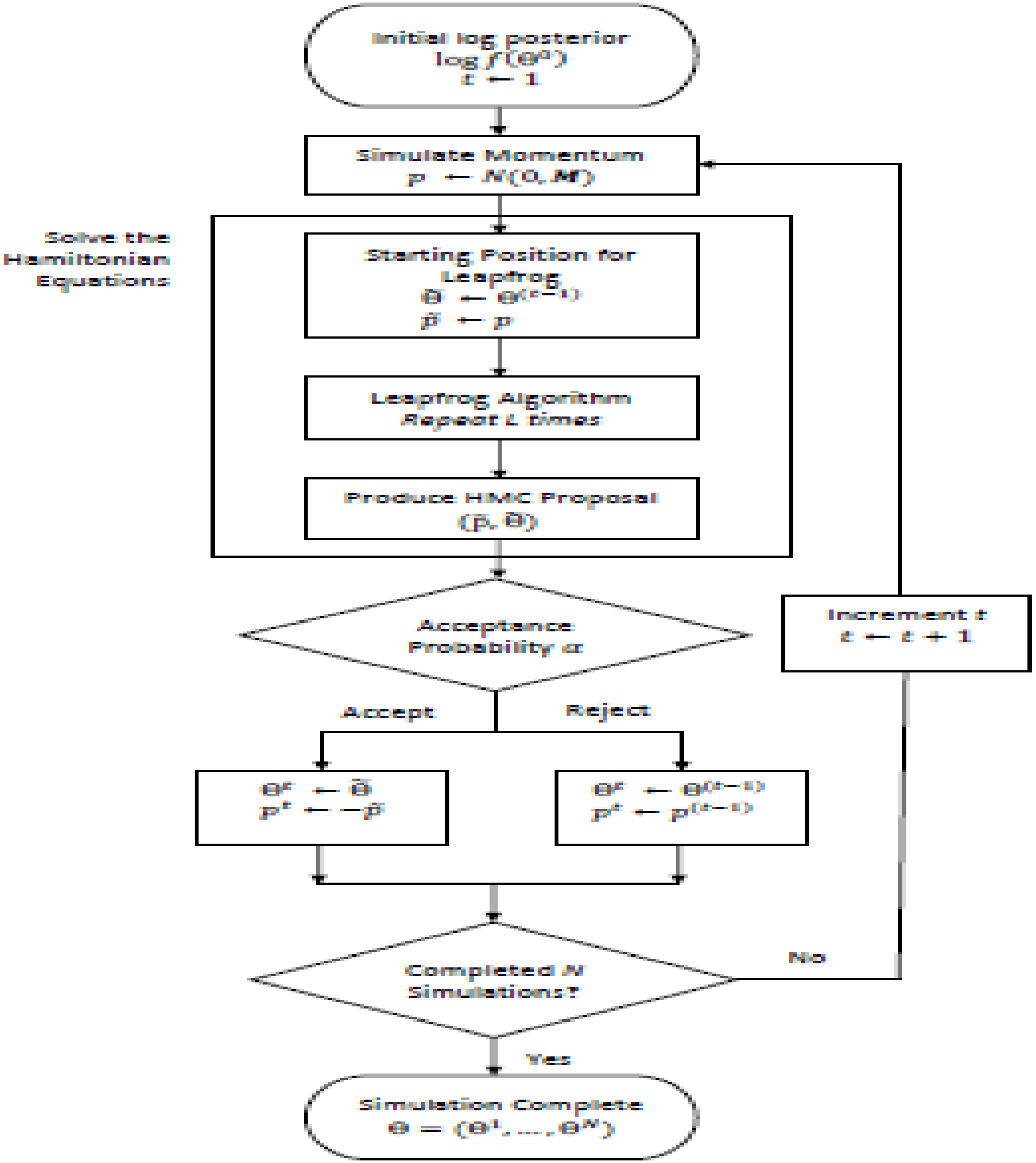
$$\left.\begin{aligned} \frac{dp}{dt} &= -\frac{\partial H(\theta,p)}{\partial \theta} = \nabla_\theta \log(f(\theta)) \\ \frac{d\theta}{dt} &= \frac{\partial H(\theta,p)}{\partial p} = M^{-1} p \end{aligned}\right] (\theta,p)$$

$$\mathbf{p}(t+\epsilon/2) = \mathbf{p}(t) + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}(t)),$$

$$\boldsymbol{\theta}(t+\epsilon) = \boldsymbol{\theta}(t) + \epsilon \mathbf{M}^{-1}\mathbf{p}(t+\epsilon/2),$$

$$\mathbf{p}(t+\epsilon) = \mathbf{p}(t+\epsilon/2) + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}(t+\epsilon)).$$

HMC. Algo I


Initial log posterior
$\log f(\Theta^0)$
$t \leftarrow 1$
Simulate Momentum
$p \leftarrow N(0, M)$
Solve the Hamiltonian Equations
Starting Position for Leapfrog
$\tilde{\Theta} \leftarrow \Theta^{(t-1)}$
$\tilde{p} \leftarrow p$
Leapfrog Algorithm
Repeat L times
Produce HMC Proposal
$(\tilde{p}, \tilde{\Theta})$
Acceptance Probability α
Accept
Reject
$\Theta^t \leftarrow \tilde{\Theta}$
$p^t \leftarrow -\tilde{p}$
$\Theta^t \leftarrow \Theta^{(t-1)}$
$p^t \leftarrow p^{(t-1)}$
Increment t
$t \leftarrow t + 1$
Completed N Simulations?
No
Yes
Simulation Complete
$\Theta = (\Theta^1, \ldots, \Theta^N)$

HMC. Algo II

procedure HMC($\boldsymbol{\theta}^{(0)}, \log f(\boldsymbol{\theta}), \mathrm{M}, N, \epsilon, L$)

  Calculate $\log f(\boldsymbol{\theta}^{(0)})$

  for $t = 1, ..., N$ **do**

    $\mathrm{p} \leftarrow N(0, \mathrm{M})$

    $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}, \bar{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(t-1)}, \bar{\mathrm{p}} \leftarrow \mathrm{p}$

    for $i = 1, ..., L$ **do**

      $\bar{\boldsymbol{\theta}}, \bar{\mathrm{p}} \leftarrow \text{Leapfrog}(\bar{\boldsymbol{\theta}}, \bar{\mathrm{p}}, \epsilon, \mathrm{M})$

    end for

    $\alpha \leftarrow \min\left(1, \frac{\exp(\log f(\bar{\boldsymbol{\theta}}) - \frac{1}{2}\bar{\mathrm{p}}^T \mathrm{M}^{-1} \bar{\mathrm{p}})}{\exp(\log f(\boldsymbol{\theta}^{(t-1)}) - \frac{1}{2}\mathrm{p}^T \mathrm{M}^{-1} \mathrm{p})}\right)$

    With probability α, $\boldsymbol{\theta}^{(t)} \leftarrow \bar{\boldsymbol{\theta}}$ and $\mathrm{p}^{(t)} \leftarrow -\bar{\mathrm{p}}$

  end for

  return $\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(N)}$

  function LEAPFROG($\boldsymbol{\theta}^*, \mathrm{p}^*, \epsilon, \mathrm{M}$)

    $\bar{\mathrm{p}} \leftarrow \mathrm{p}^* + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}^*)$

    $\bar{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \epsilon \mathrm{M}^{-1} \bar{\mathrm{p}}$

    $\bar{\mathrm{p}} \leftarrow \bar{\mathrm{p}} + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\bar{\boldsymbol{\theta}})$

    return $\bar{\boldsymbol{\theta}}, \bar{\mathrm{p}}$

  end function

end procedure

HMC Tuning

Step size. Small relative to parameter of interest

x Number of leapfrog steps. Large L.

Jointly acceptance rate of 65%

Examine for correlations

Adaptively select L as in No U-turn Sampler (NUTS) . To be seen with Stan

Covariance matrix M

Example

Sampling the bi-variate normal

Simple example to recall approach

Model. Bivariate normal with unknown means. Variances 1. Known correlation ϱ

1 observation

Prior. Uniform

Use. Expected value and variance of parameters, Expected cross product, Probability that parameter belongs to a set

Sampling the bi-variate normal. Model and posterior

$$y = (y_1, y_2) \sim N\left(\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}, \Sigma\begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right) \qquad \pi(\theta) \propto K$$

$$\pi(\theta_1,\theta_2|y) \propto \pi(\theta)\,\pi(y|\theta) \propto \pi(y|\theta)$$

$$\propto \exp\left(-\frac{1}{2}(y-\theta)'\Sigma^{-1}(y-\theta)\right) = \exp\left(-\frac{1}{2}(\theta-y)'\Sigma^{-1}(\theta-y)\right)$$

$$\theta|y \sim N(y, \Sigma)$$

Sampling the bi-variate normal. Gibbs sampler

$$\theta_1 \mid \theta_2, y \sim N\left(y_1 + \rho\,(\theta_2 - y_2),\ 1 - \rho^2\right)$$

$$\theta_2 \mid \theta_1, y \sim N\left(y_2 + \rho\,(\theta_1 - y_1),\ 1 - \rho^2\right)$$

θ_2^0 ARBITRARY, $i = 1$

UNTIL CONVERGENCE

$$\theta_1^i \sim N\left(y_1 + \rho\,(\theta_2^{i-1} - y_2),\ 1 - \rho^2\right)$$

$$\theta_2^i \sim N\left(y_2 + \rho\,(\theta_1^i - y_1),\ 1 - \rho^2\right)$$

$$i = i + 1$$

Sampling the bi-variate normal. Metropolis Hastings

$$\alpha(\theta, z) = \min\left\{ \frac{\exp\left(-\frac{1}{2}(z-y)'\,\Sigma^{-1}(z-y)\right)}{\exp\left(-\frac{1}{2}(\theta-y)'\,\Sigma^{-1}(\theta-y)\right)}, 1 \right\}$$

$$= \min\left\{ \frac{\exp\left(-\frac{1}{2}\left((z'\Sigma^{-1}z) - 2z\,\Sigma^{-1}y\right)\right)}{\exp\left(-\frac{1}{2}\left((\theta'\Sigma^{-1}\theta) - 2\theta\,\Sigma^{-1}y\right)\right)}, 1 \right\}$$

$$z = \theta + u \qquad u \sim N(0, D)$$

Sampling the bi-variate normal. Metropolis Hastings

CHOOSE θ^0, $i=0$

UNTIL CONVERGENCE DETECTED

GENERATE $u \sim N(0, D)$

$z = \theta^i + u$

COMPUTE $\alpha(\theta^i, z)$

DO $\theta^{i+1} = \begin{cases} z, & \text{WITH PROB } \alpha(\theta^i, z) \\ \theta^i, & \text{OTHERWISE} \end{cases}$

$i = i+1$

Sampling the bi-variate normal. Hamiltonian MC

procedure HMC($\boldsymbol{\theta}^{(0)}, \log f(\boldsymbol{\theta}), \mathbf{M}, N, \epsilon, L$)

  Calculate $\log f(\boldsymbol{\theta}^{(0)})$

  for $t = 1, ..., N$ **do**

    $\mathbf{p} \leftarrow N(0, \mathbf{M})$

    $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}, \tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(t-1)}, \tilde{\mathbf{p}} \leftarrow \mathbf{p}$

    for $i = 1, ..., L$ **do**

      $\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{p}} \leftarrow \text{Leapfrog}(\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{p}}, \epsilon, \mathbf{M})$

    end for

    $\alpha \leftarrow \min\left(1, \frac{\exp(\log f(\tilde{\boldsymbol{\theta}}) - \frac{1}{2}\tilde{\mathbf{p}}^T \mathbf{M}^{-1} \tilde{\mathbf{p}})}{\exp(\log f(\boldsymbol{\theta}^{(t-1)}) - \frac{1}{2}\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p})}\right)$

    With probability α, $\boldsymbol{\theta}^{(t)} \leftarrow \tilde{\boldsymbol{\theta}}$ and $\mathbf{p}^{(t)} \leftarrow -\tilde{\mathbf{p}}$

  end for

  return $\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(N)}$

  function LEAPFROG($\boldsymbol{\theta}^*, \mathbf{p}^*, \epsilon, \mathbf{M}$)

    $\tilde{\mathbf{p}} \leftarrow \mathbf{p}^* + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}^*)$

    $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \epsilon \mathbf{M}^{-1} \tilde{\mathbf{p}}$

    $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}} + (\epsilon/2)\nabla_{\boldsymbol{\theta}} \log f(\tilde{\boldsymbol{\theta}})$

    return $\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{p}}$

  end function

end procedure

$\theta | y \sim N(y, \Sigma)$

$\log(f(\theta)) \propto -\frac{1}{2}(\theta - y)' \Sigma^{-1} (\theta - y)$

$-\log(f(\theta)) \propto \frac{1}{2}(\theta - y)' \Sigma^{-1} (\theta - y)$

$-\nabla_\theta (\log(f(\theta)) \propto (\theta - y)' \Sigma^{-1}$

Sampling the bi-variate normal. Hamiltonian MC. Bonus

$$H(\theta,p) = \frac{1}{2}(\theta-y)^t\Sigma^{-1}(\theta-y) + \frac{1}{2}p^tM^{-1}p$$

$$\frac{dp}{dt} = -\frac{\partial H(\theta,p)}{\partial\theta} = (\theta-y)'\Sigma^{-1}$$

$$\frac{d\theta}{dt} = \frac{\partial H(\theta,p)}{\partial p} = M^{-1}p$$

$$p(t+\tfrac{\varepsilon}{2}) = p(t) + \frac{\varepsilon}{2}\left(y-\theta(t)\right)'\Sigma^{-1}$$

$$\theta(t+\varepsilon) = \theta(t) + \varepsilon M^{-1}p(t+\varepsilon/2)$$

$$p(t+\varepsilon) = p(t+\tfrac{\varepsilon}{2}) + \frac{\varepsilon}{2}\left(y-\theta(t+\tfrac{\varepsilon}{2})\right)'\Sigma^{-1}$$

Sampling the bi-variate normal. Answers (Whatever the method used)

$$\hat{E}(\theta_1|y) = \frac{1}{K}\sum_{i=1}^{K} \theta_1^{M+i}$$

$$\hat{E}(\theta_1\theta_2|y) = \frac{1}{K}\sum_{i=1}^{K} \left(\theta_1^{M+i}\,\theta_2^{M+i}\right)$$

$$Pr(\theta_1^i \in A|y) = \frac{\#\{\theta_1^{M+i} \in A\}}{K}$$

Further variants

Reversible jump

In many complex problems we need to do trans-dimensional Markov chain simulation

- Mixtures with unknown number of components
- Shallow neural nets with unknown number of hidden nodes
- Model averaging
- Bartmachine

Parameters=(indicator of model, parameters of such model)

Within the model, a ‘standard’ Markov chain (Gibbs, MH, HMC, etc…)

Between models, Metropolis Hastings with reversible moves (jumps, collapsing and splitting models…)

See classic paper by Peter Green in VC

Particle filtering

For nonlinear sequential problems, MCMC gets complex

Generate initial sample at time t=0

Let them evolve (and learn) according to nonlinear sequential model

Introduce rules to avoid collapse of particles

Augmented probability simulation (I)

Expected utility when probabilities depend on alternative

$$\psi(a) = \int u(a,\theta)\, p(\theta \mid x, a)\, d\theta$$

If utility positive and integrable, define augmented probability distribution

$$h(a,\theta) \propto u(a,\theta)\, p(\theta \mid x, a)$$

Mode of marginal of AP is the optimal alternative

$$\int h(a,\theta)\, d\theta \propto \int u(a,\theta)\, p(\theta \mid x, a)\, d\theta = \psi(a)$$

Augmented probability simulation (II)

Proposed scheme

1. Generate a sample $((\theta^1, a^1), ..., (\theta^m, a^m))$ from density $h(a, \theta)$.
2. Convert it to a sample $(a^1, ..., a^m)$ from the marginal $h(a)$.
3. Find the sample mode.

For 1, MCMC technology

For 2, cluster analysis, density estimation,….

Inference and assessing convergence

Inference

Once convergence detected, collect samples from posterior and perform inference (point estimates, intervals, hypothesis tests, predictions and expected utility computations) via Monte Carlo (recall uncertainty associated, they are stochastic algos!!!)

Difficulties

If iterations have not proceeded long enough, target is not approximated well, samples are unrepresentative of target!!!

Early iterations may bias results

Autocorrelation impacts precision of estimates and the effective number of samples may be smaller than the one actually drawn (as if we'd be using a smaller number of samples)

Solutions

Runs to allow for effective monitoring of convergence (based on multiple chains, recall labs)

Monitor convergence by comparing variation within and between simulated sequences (until within and between variation are similar)

Modifying the algorithm by reparameterising or learning good parameterisations, if efficiency is very low (algo too slow)

Discarding initial values

Thinning

Take into account AC when estimating precisions

Final comments

Final comments

Gibbs. Lots of hard prior work. Not always implementable. If so may work well.

Metropolis Hastings. Less prior work. Quite general. May work slowly.

Hamiltonian. Even less work. Quite general. Works more efficiently…. Yet suffers in large scale problems

Computational problem in Bayesian analysis

Computing the posterior

$$p(\theta|x) = \frac{p(x|\theta)\, p(\theta)}{p(x)} = \frac{p(x|\theta)\, p(\theta)}{\int p(x|\theta)\, p(\theta)\, d\theta} \propto p(x|\theta)\, p(\theta)$$

Large scale problems. The modern statistical paradigm (but not always and not for the whole problem)

What if the amount of data is large? (Big data problems)
What if the amount of parameters is large? (e.g. neural nets)