10 - Markov Chain Monte Carlo

Bayesian Statistics

Spring 2022-2023

Josep Fortiana

Matemàtiques - Informàtica UB

Monday, May 08, 2023

10 - MCMC

Basic Metropolis algorithm

Metropolis-Hastings algorithm

Gibbs sampling

Hamiltonian Monte Carlo (HMC)

10 - MCMC

Setting

Model with parameter $\theta \in \Theta \subset \mathbb{R}^p$,

Observations: $x = (x_1, \ldots, x_n)$.

$$\begin{aligned} &\textit{Likelihood:} && f(x \mid \theta) = \textstyle\prod_{i=1}^{n} f(x_i \mid \theta), \\ &\textit{Prior pdf:} && h(\theta), \\ &\textit{Joint } (\theta, x) \textit{ pdf:} && f(x \mid \theta) \cdot h(\theta). \end{aligned}$$

Target density

Posterior pdf:

$$h_x(\theta) = h(\theta \mid x) = \frac{f(x, \theta)}{Z_x},$$

Z_x is the *normalization constant,* marginal pdf of x,

a.k.a. *prior predictive pdf,* evaluated at the observed x.

Z_x is the integral of $f(x, \theta)$ with respect to θ.

Metropolis algorithm

Construct a sequence $\{\theta^{(t)}\}_{t\in\mathbb{N}}$ of points in Θ:

A trajectory of a *Markov chain* whose limit pdf is $h_x(\theta)$.

A step in the chain $\theta^{(t)} \longrightarrow \theta^{(t+1)}$ is as follows:

Metropolis algorithm

Each step in the chain has:

1. Change proposal - A random jump.

   Sample from an auxiliary candidate generation pdf

2. Acceptance-rejection: Do we accept the proposal?

   Depends on the target pdf value at the candidate point.

[1] Candidate generation

Sample from a *candidate generation* pdf g, e.g., Gaussian, or Student's t, or Unif$(-\delta, \delta)$, $g(-\theta) = g(\theta)$,.

A dispersion parameter (δ in the uniform)

Candidate: $\theta' = \theta + u$, where $u \sim g$.

Transition kernel (sort of "matrix"):

$$k(\theta' \mid \theta) = g(\theta' - \theta).$$

Since g is a pdf, $\int_{\theta'} k(\theta' \mid \theta)\, d\theta' = 1$.

Remark on notation

Some books write:

$$k(\theta, \theta') \quad \text{(equivalent to our)} \quad k(\theta' \mid \theta).$$

I prefer to keep analogy both with the finite case and conditional density notation.

$$k_{\theta\theta'} = \mathrm{P}(\text{transition to state } \theta' \mid \text{current state is } \theta).$$

(θ'-th entry in row θ)

Metropolis - [2] Acceptance-rejection step

Generate a random indicator $I \sim \text{Ber}(p)$ with:

$$p = \min \left\{ 1, \frac{h_x(\theta')}{h_x(\theta)} \right\},$$

- If $I = 1$, accept the update: $\theta^{(t+1)} = \theta'$,
- If $I = 0$, keep the current state: $\theta^{(t+1)} = \theta$.

Intuitively

Wander randomly about the state space Θ.

We want to go more often, and stay longer, where the probability $h_x(\theta)$ is higher.

A jump to θ' is proposed. Then:

- If h_x is higher at θ', go there unconditionally.
- If it is lower we go there only conditionally, with a probability proportional to this smaller value.

No denominators

Target pdf appears only in the quotient:

$$\frac{h_x(\theta')}{h_x(\theta)},$$

Z_x is NOT required. Only the joint pdf $f(x, \theta)$:

$$p = \min\left\{1, \frac{f(x, \theta')}{f(x, \theta)}\right\}.$$

Scale in candidate generation

Dispersion parameter(s) δ in candidate generation pdf g.

Tradeoff between:

- Small δ High acceptance probability, slow displacement in Θ,
- Large δ Swift displacement, small acceptance probability.

Resulting Markov chain

State space (continuous): Θ.

Transition kernel ("matrix"):

$$P(\theta' \mid \theta) = k(\theta' \mid \theta) \cdot \min \left\{ 1, \frac{h_x(\theta')}{h_x(\theta)} \right\}$$

Check the detailed balance condition

Multiplying by $h_x(\theta)$,

$$h_x(\theta) \cdot P(\theta' \mid \theta) = k(\theta' \mid \theta) \cdot \min\left\{h_x(\theta), h_x(\theta')\right\},$$

By the symmetry of $k(\theta'|\theta)$ this is equal to:

$$k(\theta \mid \theta') \cdot \min\left\{h_x(\theta'), h_x(\theta)\right\} = h_x(\theta') \cdot P(\theta \mid \theta').$$

Hence it is a *time-reversible* Markov chain.

The target pdf $h_X(\theta)$ is the limit probability

Indeed (it satisfies the eigen equation):

$$\int_{\theta\in\Theta} h_X(\theta)\cdot P(\theta'\mid\theta)\,d\theta = \int_{\theta\in\Theta} h_X(\theta')\cdot P(\theta\mid\theta')\,d\theta$$

$$= \; h_X(\theta')\cdot\int_{\theta\in\Theta} P(\theta\mid\theta')\,d\theta = h_X(\theta').$$

10 - MCMC

Basic Metropolis algorithm

Metropolis-Hastings algorithm

Gibbs sampling

Hamiltonian Monte Carlo (HMC)

References

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953), *Equation of State Calculations by Fast Computing Machines*, J. Chemical Physics, Vol. 21, pp. 1087–1092.

Wilfred Keith Hastings (1970), *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika 57, 97-109.

Description; generalize Metropolis

Non-symmetric candidate proposal kernel $k(\cdot \mid \cdot)$.

Acceptation rule is modified to compensate.

Accept $\theta^{(m+1)} = \theta'$ with probability:

$$\min \left\{ 1, \frac{h_x(\theta') \cdot k(\theta \mid \theta')}{h_x(\theta) \cdot k(\theta' \mid \theta)} \right\},$$

Metropolis Hastings transition kernel

Transition kernel:

$$P(\theta' \mid \theta) = k(\theta' \mid \theta) \cdot \min \left\{ 1, \frac{h_x(\theta') \cdot k(\theta \mid \theta')}{h_x(\theta) \cdot k(\theta' \mid \theta)} \right\}.$$

Detailed balance condition

Multiplying by $h_x(\theta)$,

$$\begin{aligned} h_x(\theta) \cdot P(\theta' \mid \theta) &= k(\theta' \mid \theta) \cdot \min\left\{ h_x(\theta), \frac{h_x(\theta') \cdot k(\theta \mid \theta')}{k(\theta' \mid \theta)} \right\} \\ &= \min\left\{ h_x(\theta) \cdot k(\theta' \mid \theta),\ h_x(\theta') \cdot k(\theta \mid \theta') \right\} \\ &= h_x(\theta') \cdot P(\theta \mid \theta'), \end{aligned}$$

hence the chain is reversible with respect to $h_x(\theta)$.

Construction of $k(\cdot\,,\,\cdot)$

Based on a random walk.

From x, the *proposed* y is equal to x plus a random $z = y - x$, generated from a pdf g.

$$k(x, y) = g(z) = g(y - x).$$

When g is a symmetric pdf, we recover the Metropolis algorithm, where $k(\cdot\,,\,\cdot)$ is a symmetric kernel.

10 - MCMC

Basic Metropolis algorithm

Metropolis-Hastings algorithm

Gibbs sampling

Hamiltonian Monte Carlo (HMC)

General idea

Gibbs sampling is a *relaxation* algorithm.

Meaning:

Approximate a multivariate simulation by a sequence, where variables are updated one at a time.

Setting (two-stage)

Two r.v.'s X and Y, with joint pdf $f(x, y)$,

Marginal densities $f_X(x)$, $f_Y(y)$,

Conditional densities:

$$f_{Y|X=x} = \frac{f(x,y)}{f_X(x)}, \qquad f_{X|Y=y} = \frac{f(x,y)}{f_Y(y)},$$

Sequence of pairs

Generate a double sequence:

$$(x, y) = \{(x_t, y_t) : \ t \in \mathbb{Z}, \ t \geq 1\},$$

defined by:

$$\begin{cases} y_t & \sim & f_{Y|X=x_{t-1}} \\ x_t & \sim & f_{X|Y=y_t} \end{cases}$$

Start: $x_0 \in Suport(X)$

Properties of the sequence

The sequence is a trajectory of a Markov chain.

The chain converges to the joint distribution

$$\sim f(x, y).$$

(Each component to the corresponding marginal)

Gibbs sampling (multivariate)

Simulate a random vector:

$$\boldsymbol{X} = (X_1, \ldots, X_k).$$

with unknown or complicated joint pdf $f(\boldsymbol{x})$.

But for each i, the conditional pdf:

$$X_i \mid (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_k)$$

is known and easy to simulate.

Gibbs sampling: first cycle, first step

Set an initial value: $x_0 = (x_{01}, \ldots, x_{0k})$.

Update $\boxed{x_{01} \to x_{11}}$, by simulation of:

$$X_1 \Big| (X_2 = x_{02}, \ldots, X_k = x_{0k}) \ ,$$

This is called a *full conditional* pdf.

New running value: $x = (x_{11}, x_{02}, \ldots, x_{0k})$.

Gibbs sampling: first cycle, second step

Update $\boxed{x_{02} \longrightarrow x_{12}}$, by simulation of:

$$X_2 \Big|_{(X_1 = x_{11}, X_3 = x_{03}, \ldots, X_k = x_{0k})} ,$$

Continue in this fashion until updating $\boxed{x_{0k} \longrightarrow x_{1k}}$, from:

$$X_k \Big|_{(X_1 = x_{11}, X_2 = x_{12}, X_3 = x_{13}, \ldots, X_{k-1} = x_{1,k-1})} ,$$

which ends the first cycle.

Gibbs sampling: second and following cycles

In the second cycle $x_1 = (x_{11}, \ldots, x_{1k})$ is updated to $x_2 = (x_{21}, \ldots, x_{2k})$. Continue in the same manner.

The result is a sequence $\{X_n\}_{n \in \mathbb{N} \cup \{0\}}$, which under fairly general conditions* converges:

$$X_n \xrightarrow[n \to \infty]{\mathscr{L}} X.$$

*Not always!

10 - MCMC

Basic Metropolis algorithm

Metropolis-Hastings algorithm

Gibbs sampling

Hamiltonian Monte Carlo (HMC)

Simulating a pdf

Variable: $\theta \in \Theta \subset \mathbb{R}^p$, parameter in a Bayesian model.

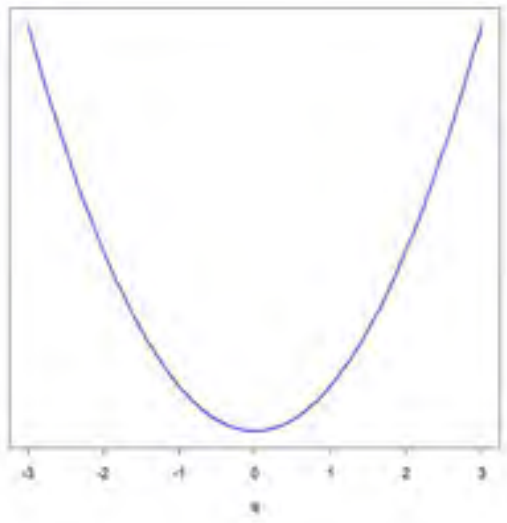
Function: $g(\theta)$, a posterior pdf:

MCMC: A moving point takes a random walk (Metropolis) or moves along a coordinate (Gibbs).

Analogy: Newton dynamics

Variable: coordinate q of a point mass m.

Function: $V(q)$, a potential well:



Newton's Law: $m\,\ddot{q} = F = -m \cdot \dfrac{\partial V}{\partial q}$.

Actually, more than an analogy

Metropolis algorithm originates in Statistical Physics.

Maxwell-Boltzmann. Pmf on a discrete set of states:

$$\mathrm{P}_j = \frac{1}{Z} \cdot \exp\left\{-\frac{E_j}{kT}\right\}.$$

P_j is the probability of state $\{j\}$ and E_j is the energy.

Similarly for the pdf for a continuous state space.

Metropolis algorithm

The acceptance-rejection rule intends to:

Send the particle to high probability (low energy) regions as soon as possible and stay there as long as possible.

Hindered by the random walk wandering.

Random walk drawbacks

At a given time, to move further away is more likely

than getting closer to high probability regions.

Small steps:

High acceptance rate, long chain and computation time.

Alternatives

1. Independent (MH) candidate proposal.

Candidate proposal pdf must be close to the target pdf.

(otherwise acceptance rates are small)

2. Gibbs sampling

A sequence of low dimensional simulations

still converges to the right target.

Break *the curse of dimensionality*.

3. HMC: deterministic complement to RW

Candidate states far from the current one,

that will be accepted with a high probability.

Likely the new state will be

in a higher probability region.

The force does precisely this

Define a *potential well* as minus the log-target pdf:

$$V = -\log(g).$$

The force $F = -\dfrac{\partial V}{\partial q}$ sends the straying particle back down more strongly the further away it went.

(Also the rationale of *gradient descent*)

Hamilton dynamics

Kinetic energy of a point mass: $T = \frac{1}{2} m \dot{q}^2$.

In terms of the *momentum* $p = m \dot{q}$,

$$T = \frac{p^2}{2m},$$

Newton's (second) law becomes:

$$m \ddot{q} = \dot{p} = \frac{dp}{dt} = F = -\frac{\partial V}{\partial q}.$$

Hamilton mechanics

(William Rowan Hamilton, 1833)

Total energy of a dynamical system:

$$H(p, q) = T(p) + V(q).$$

The momentum p is an additional, auxiliary variable.

Hamilton equations

A 2^{nd} order differential equation -Newton's 2^{nd} law- is replaced by two 1^{st} order equations:

$$m\,\ddot{q} = \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\frac{\partial V}{\partial q},$$

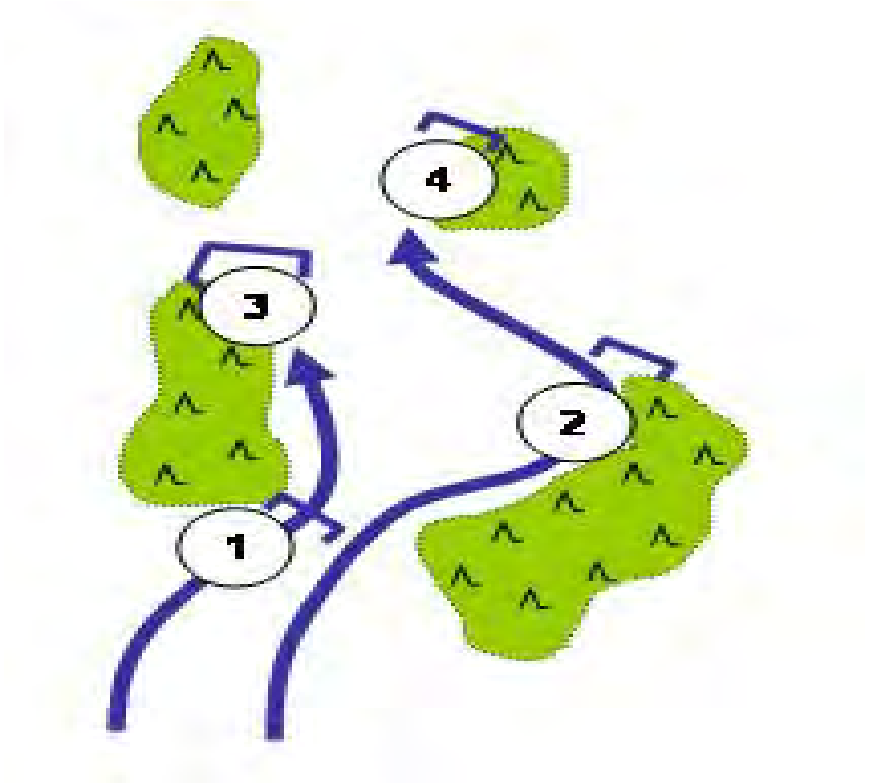
$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = \frac{\partial T}{\partial p} = \frac{p}{m}.$$

Euler integration

$$p(t+\epsilon) = p(t) + \epsilon \cdot \frac{dp}{dt}(t) = p(t) - \epsilon \cdot \frac{\partial V}{\partial q}(t),$$

$$q(t+\epsilon) = q(t) + \epsilon \cdot \frac{dq}{dt}(t) = q(t) + \epsilon \cdot \frac{\partial K}{\partial p}(t).$$

Leapfrogging



Leapfrog integration

$$p(t+\tfrac{\epsilon}{2}) = p(t) - \left(\tfrac{\epsilon}{2}\right) \cdot \frac{\partial V}{\partial q}(t),$$

$$q(t+\epsilon) = q(t) + \epsilon \cdot \frac{\partial K}{\partial p}(t+\tfrac{\epsilon}{2}),$$

$$p(t+\epsilon) = p(t) - \left(\tfrac{\epsilon}{2}\right) \cdot \frac{\partial V}{\partial q}(t+\epsilon).$$

Candidate generation

From an initial state (q, p),

generate a new $\widetilde{p}$ from a Normal$(0, 1)$.

From $(q, \widetilde{p})$, a Hamiltonian trajectory,

L leapfrog steps to a new (q', p').

Acceptance probability

The transition :

$$(q, p) \to (q', p')$$

is accepted with probability:

$$\min\left\{1, \exp(H(q, p) - H(q', p'))\right\}$$

Adjustable parameters

- Step length ϵ
- Number L of leapfrog steps in each "jump".

NUTS: No-U-Turn Sampler

Hoffman, Matthew D. and Gelman, Andrew (2014),

"The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo."

Richard McElreath

Markov Chains: Why Walk When You Can Flow?

Dustin Stansbury

MCMC: Hamiltonian Monte Carlo

(a.k.a. Hybrid Monte Carlo)

Chi Feng

The Markov-chain Monte Carlo Interactive Gallery

Utkarsh Gupta

MCMC: Hamiltonian Monte Carlo and No-U-Turn Sampler

Alex Rogozhnikov

Hamiltonian Monte Carlo explained