June 2015

MURALI HARAN: Some notes on MCMC (in conjunction w/ astrostats tutorial)

Monte Carlo Methods

- Learn about characteristics of probability distr.
  Common but <u>not limited to</u> Bayes

More specific: approximate expected values

Want: $\mu = E_f\{g(x)\} = \int g(x) f(x)\,dx$

$\sum_{\text{all } x} g(x) f(x)$

$g(x)$ is a real-valued function

Monte Carlo idea (basic stats.)

Simulate: $X_1, X_2, \ldots \sim f(x)$

$\hat{\mu}_n = \dfrac{\sum_{i=1}^{n} g(X_i)}{n}$ approximates μ.

[Here: only probability calculations, not statistics]

Need for MCMC / other methods:
Usually ~~ca~~ cannot easily generate $X_1, X_2, \ldots \sim f(x)$
(HARD)

Two methods:

- Importance sampling: sample from a different distr. (EASY) and reweight to approx μ.
- MCMC: construct ~~use~~ Markov chain and use its states to approximate μ.

Imp. sampling idea (basics):

$$\mu = \int g(x) f(x)\,dx$$ if easy distr. $q(x)$ satisfies some conditions s.t.

Then, $Y_1, Y_2, \ldots \sim q(x)$

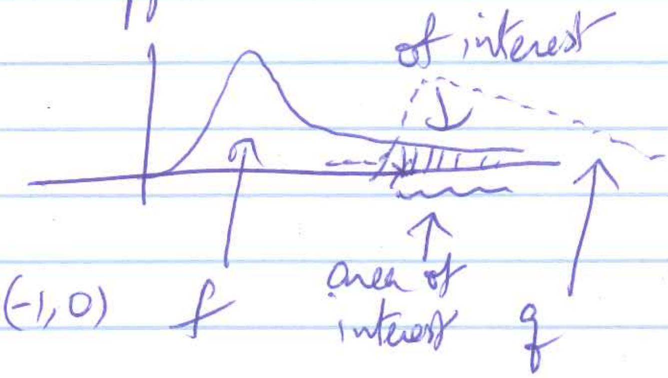
Use the fact that $\int g(x) f(x)\,dx = \int \left[\frac{g(x)}{q(x)} f(x)\right] q(x)\,dx$

$$= E_q\left\{ \frac{g(x)}{q(x)} f(x) \right\}$$

$$\hat{\mu}_{IMP} = \sum_{i=1}^{n} \frac{g(Y_i)}{q(Y_i)} f(Y_i) \Big/ n$$

Choose q s.t. it matches

- Easy to simulate
- support of q includes f's support
- match $f(x)\, g(x)$



E.g. of $g(x)$:

Toy e.g. $P(-1 < X < 0)$

(1) $g(x) = \begin{cases} 1 & \text{if } X \in (-1, 0) \\ 0 & \text{else} \end{cases}$

$$E\, g(x) = P(-1 < X < 0)$$

(2) $g(x) = x$ Simple expected value

MCMC = Use Metropolis-Hastings algorithm (recipe but w/ lots of flexibility/freedom)
[Note: Metropolis alg. Gibbs sampling etc. are special cases.]
to do the following:
Construct a Markov chain w/ stationary/target distr. $f(x)$.
$X_1, X_2, X_3, \ldots$
And then, ~~construct~~ obtain approximation

$$\hat{\mu}_n = \frac{\sum_{i=1}^{n} g(x_i)}{n}$$

(Skipping theory) for large n, X_n looks like a sample from $f(x)$.
More importantly, $\hat{\mu}_n \to \mu$ as $n \to \infty$

Recall: Goal: approximate $\mu = E_f\{g(x)\}$
Problem: HARD, also sampling $X_1, X_2, \ldots \overset{iid}{\sim} f(x)$ is HARD
MCMC provides an alternative.

Bayes example. First, review

Simple version:

Have a probability model for observations, x,

$r(x;\theta)$

~~$X_1, X_2, X_3,$~~

$z_1, z_2, z_3, \ldots, z_N \overset{iid}{\sim} r(z;\theta)$ e.g. Gamma $(z; \underbrace{\alpha,\beta}_{\theta})$

Prior for θ, $p(\theta)$

Bayesian inference is based on posterior distr.,

$$f(\theta \mid z_1,\ldots,z_N) \propto \underbrace{\prod_{i=1}^{N} r(z_i;\theta)}\, p(\theta)$$

if data values are fixed, this is the likelihood function, $\mathcal{L}(\theta; \underset{\sim}{z})$

Goal: study characteristics (expectations) w.r.t. $f(\theta \mid z_1,\ldots,z_N)$

Pick initial state of Markov chain:

$$\begin{pmatrix}\theta^{(1)}\\ \lambda^{(1)}\\ b_1^{(1)}\\ b_2^{(1)}\\ k^{(1)}\end{pmatrix} \xrightarrow[\text{full conditional)}]{\text{Gibbs update (sample from Gamma}} \begin{pmatrix}\theta_1^{(2)}\\ \lambda^{(1)}\\ b_1^{(1)}\\ \vdots\\ k^{(1)}\end{pmatrix} \xrightarrow[\text{update}]{\text{Gibbs}}$$

$$\begin{pmatrix}\theta^{(2)}\\ \lambda^{(2)}\\ b_1^{(1)}\\ b_2^{(1)}\\ k^{(1)}\end{pmatrix} \longrightarrow \begin{pmatrix} \\ \\ \\ \end{pmatrix} \longrightarrow \begin{pmatrix}\theta_1^{(2)}\\ \vdots\\ b_2^{(2)}\\ k^{(1)}\end{pmatrix} \xrightarrow[\text{update}]{\text{Metrop-Hastings}} \begin{pmatrix}\theta^{(2)}\\ \lambda^{(2)}\\ b_1^{(2)}\\ b_2^{(2)}\\ k^{(2)}\end{pmatrix}$$

Note: $k^{(2)}$ may be $= k^{(1)}$ if proposal was rejected.

$$\begin{pmatrix}\theta^{(1)}\\ \vdots\\ k^{(1)}\end{pmatrix} \quad \begin{pmatrix}\theta^{(2)}\\ \vdots\\ k^{(2)}\end{pmatrix} \quad \cdots \quad \begin{pmatrix}\theta^{(n)}\\ \vdots\\ k^{(n)}\end{pmatrix}$$

e.g. Approximate $E_f(\theta)$ by $\sum_{i=1}^{n} \theta^{(i)}/n$

" $E_f(\theta/\lambda)$ by $\dfrac{\sum_{i=1}^{n} \theta^{(i)}/\lambda^{(i)}}{n}$

Things to worry about:

(A) — bias due to poor initial value or "slow mixing" chain

If we could get ~~a~~ initial draw close to a draw from f, no bias!

(B) — variance of $\hat{\mu}_n$: compute MCMC s.error to assess this

↖ e.g. batch means method (see tutorials)

(C) — potential multiple modes in $f(x)$.

2 modes

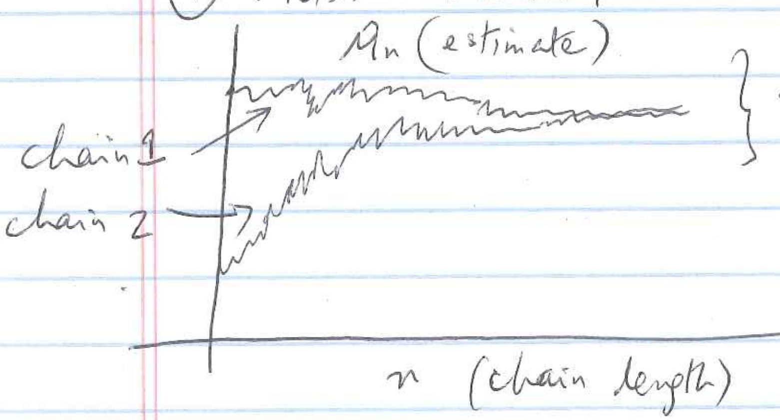
Quick rules:

(1) run multiple (say > 3-5) chains from different starting values: helps (A), (C).

(2) run chains for as long as you can

(3) Compute MCMC s.e: helps (B).

(4) Plots: monitor / occassionally look at:

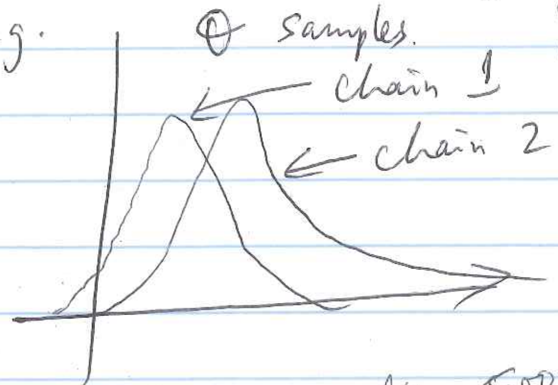


} ← Chains w/ different initial values get similar results ⇒ GOOD!

Different results ⇒ BAD! Fix code or run longer or investigate possible multiple modes

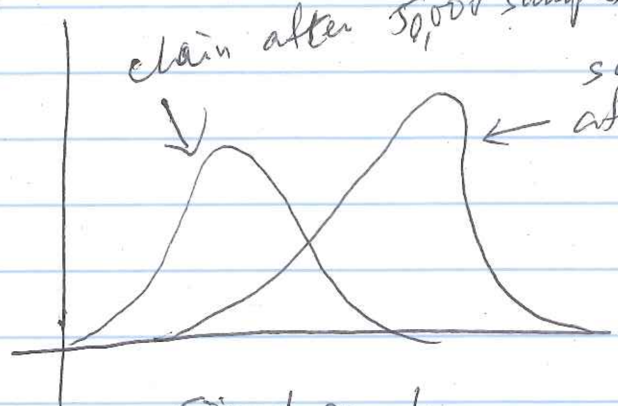
(4) Look at density (smooth histogram) plots

E.g.



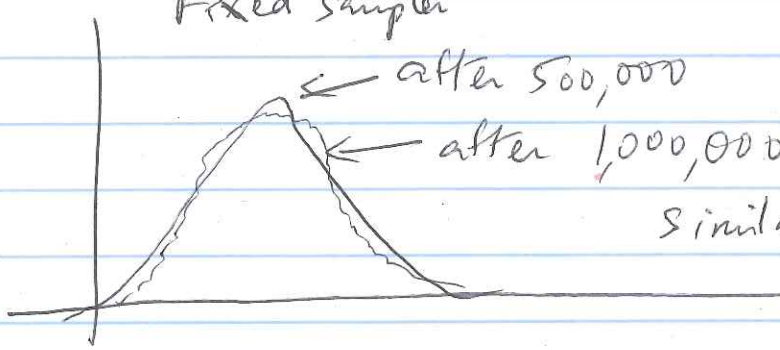
← differential initial values give different plots: BAD!

E.g.



BAD!!
Run longer?
Fix code?
Improve sampler?

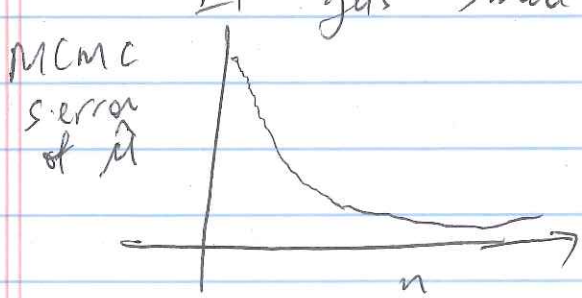
Fixed sampler

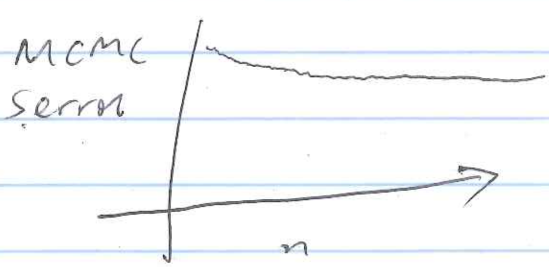


Similar: GOOD!

(5) Always monitor MCMC s.errors, especially for important parameters.

If gets small fast, good Else, BAD!

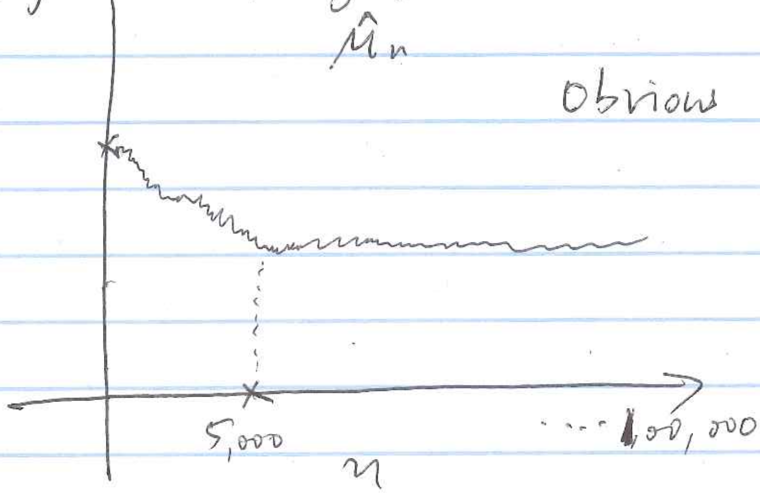




(6) Burn-in is useful, i.e. discarding some initial values is useful, but do it carefully.

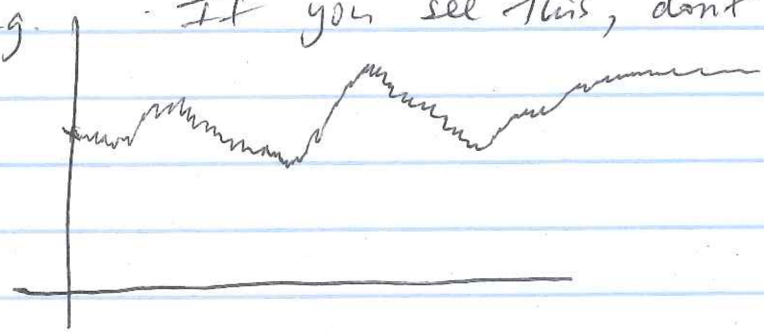
BUT } Remember: it *may* help reduce bias but throwing out samples will invariably inflate variance / s. errors!

E.g. If you see this:



then: discard first 5,000

E.g. If you see this, don't discard anything!



If you want to be really sure:

(1) Write 2 different MCMC algorithms for same problem and compare.

(2) Run chains from a few different starting pts.

(3) Run chains for *very* long times.

Good luck!