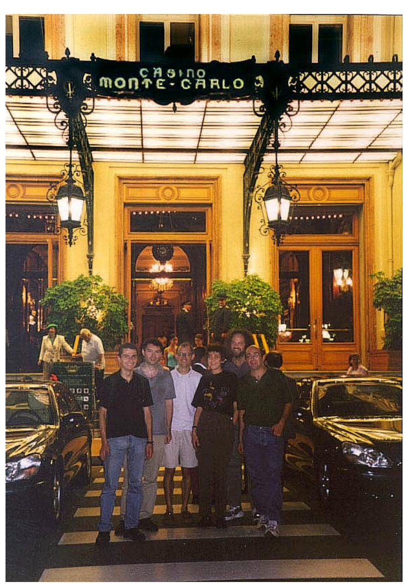
The Magic of Monte Carlo

Jeffrey S. Rosenthal
University of Toronto
www.probability.ca

(Data Sciences Speaker Series at U of T, Oct 18, 2021)

What is Monte Carlo?

Nice Place for a Conference!



But What About Monte Carlo Algorithms?

Monte Carlo in a nutshell: To sample is to know.

e.g. Estimate $\mathbf{E}[Z^4 \cos(Z)]$ where $Z \sim N(0,1)$. How?
Sample $z_1, z_2, \ldots, z_M \sim N(0,1)$, use $\frac{1}{M}\sum_{i=1}^M z_i^4 \cos(z_i)$.

e.g. Compute $\int_0^1 \int_0^1 \sin(x^2 y + y^3)\, dy\, dx$. How?
Sample $x_1, \ldots, x_M, y_1, \ldots, y_M \sim \text{Uniform}[0,1]$, use $\frac{1}{M}\sum_{i=1}^M \sin(x_i^2 y_i + y_i^3)$.

e.g. If π is a posterior density from a Bayesian data analysis, we can use a sample $X_1, X_2, \ldots, X_M \sim \pi$ in order to:

- See a picture of π: histogram, density estimate, ...
- Estimate the mean of π, by $\frac{1}{M}\sum_{i=1}^M X_i$.
- Estimate the mean of any function h of π, by $\mathbf{E}_\pi(h) \approx \frac{1}{M}\sum_{i=1}^M h(X_i)$.
- Estimate the probability of any event A, by $\mathbf{P}_\pi(A) \approx \frac{1}{M}\sum_{i=1}^M \mathbf{1}(X_i \in A)$.

Extremely popular! Widely used for data analysis in: Bayesian Statistics, Medical Research, Statistical Genetics, Chemical Physics, Computer Science, Mathematical Finance, Insurance, Engineering, etc.

- To sample is to know!

But How Can We Sample?

If π is complicated and high-dimensional, we can't easily write a computer program to directly sample from it.

Instead, use Markov Chain Monte Carlo (MCMC)!

e.g. the Metropolis Algorithm (1953):

- Given a previous state X, propose a new state $Y \sim Q(X, \cdot)$.

(Assume that Q is symmetric about X; otherwise "Metropolis-Hastings".)

- Then, if $\pi(Y) > \pi(X)$, accept the new state and move to it.
- If not, then accept it only with probability $\pi(Y) / \pi(X)$, otherwise reject it and stay where you are.
- Then, sit back and watch the magic! [Metropolis]

The empirical distribution (black) converges to the target (blue).

So, MCMC works! Magic! (Or, rather, Markov chain theory: The process is irreducible, and reversible so π is a stationary distribution.)

So, after throwing away some initial bad samples ("burn-in"), can then estimate $\mathbf{E}_\pi(h) \approx \frac{1}{M-B}\sum_{i=B+1}^M h(X_i)$, etc.

Example: Interacting Particle System

Suppose there are n particles in some region, with probability density proportional to e^{-H} (where H is an "energy function").

What is (say) the average rightmost location?

Can we average over all possible configurations? Difficult – infinite.

Can we create random samples, and use Monte Carlo? Yes!

e.g. Suppose the probability of a configuration is proportional to e^{-H}, where

$$H = A \sum_{i<j} \left|(x_i, y_i) - (x_j, y_j)\right| + B \sum_{i<j} \frac{1}{\left|(x_i, y_i) - (x_j, y_j)\right|} + C \sum_i x_i$$

$A = B = C = 0$: independent particles.
$A \gg 0$: particles like to be close together.
$B \gg 0$: particles like to be far apart.
$C \gg 0$: particles like to be towards the left.

Then what is the distribution of the rightmost point $\max_i x_i$ (etc.)?

Cannot directly sample particles with density proportional to e^{-H}.
But we can use a Metropolis algorithm!
Propose to move the particles, one at a time, each with a $N(0, \sigma^2)$ increment.
Then accept/reject those proposals, using the same rule as before.
Does it converge? How quickly? [PointProc]

What other theory is known? Lots!

Optimising MCMC

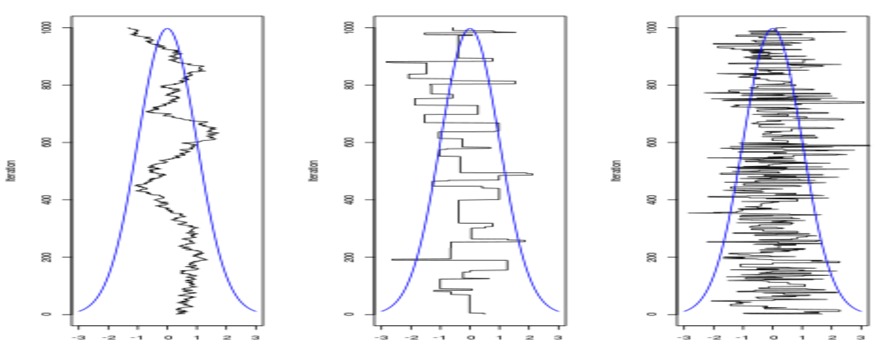
To be useful, MCMC must converge sufficiently quickly.

Ideally: Prove that the black and blue are within 0.01 (say) of each other, after some specific number n_* of iterations.

Some progress (e.g. $n_* = 140$), by "coupling" two different algorithm copies together. [R., JASA 1995, Stat & Comput. 1996] But difficult!

Instead: Which proposal distribution converges the fastest? [Metropolis]

Example: Target $\pi = N(0, 1)$. Suppose we propose from $Q(x, \cdot) = N(x, \sigma^2)$. What is best σ? Trace plots, with "time" moving upwards:



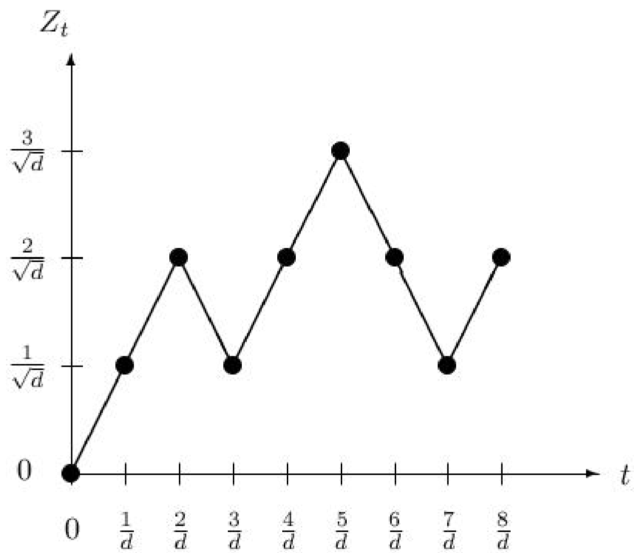
$\sigma = 0.1$	$\sigma = 25$	$\sigma = 2.38$
too small !	too big !	just right
A.R. = 0.962	0.052	0.441

So, want "moderate" σ, and "moderate" acceptance rate (A.R.).

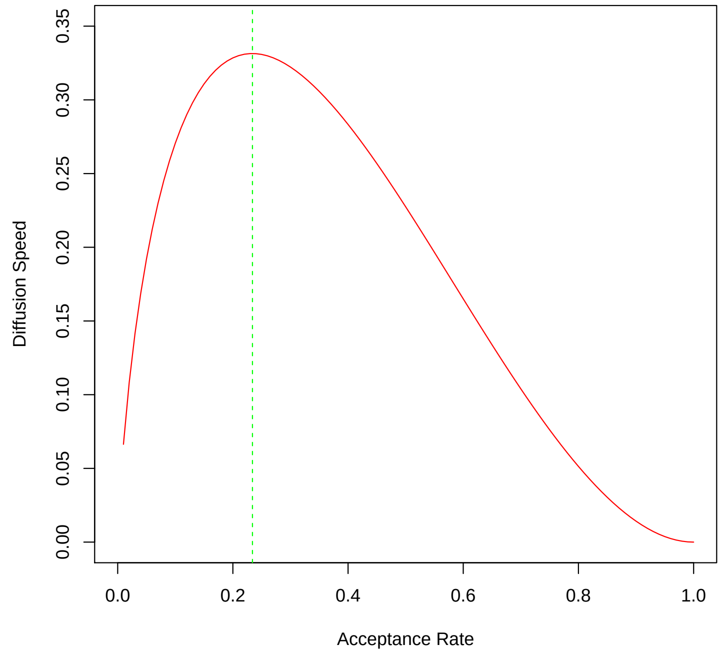
The best proposals are not too big, and not too small, but "just right".

Learning from Diffusion Limits

Recall: if $\{X_n\}$ is simple random walk, and $Z_t = d^{-1/2} X_{dt}$ (i.e., we speed up time, and shrink space), then as $d \to \infty$, the process $\{Z_t\}$ converges to Brownian motion (i.e., a diffusion).



Theorem [Roberts, Gelman, Gilks, AAP 1997]:
Similar limits hold for a Metropolis algorithm, in dimension d, as $d \to \infty$:
A Metropolis algorithm with normal proposals converges (coordinate-wise) under "certain conditions" as $d \to \infty$ to a diffusion with speed proportional to $A\,[\Phi^{-1}(A/2)]^2$ where A is the acceptance rate.



- This speed is maximised when $A \doteq 0.234$, i.e. it is <u>optimal</u> to find a scaling σ^2 which gives an acceptance rate of 0.234. Simple! Good!

- The corresponding optimal proposal covariance is $\Sigma = \frac{(2.38)^2}{d} \Sigma_\pi$, where Σ_π is the covariance of π. [Roberts & R., Stat Sci 2001]

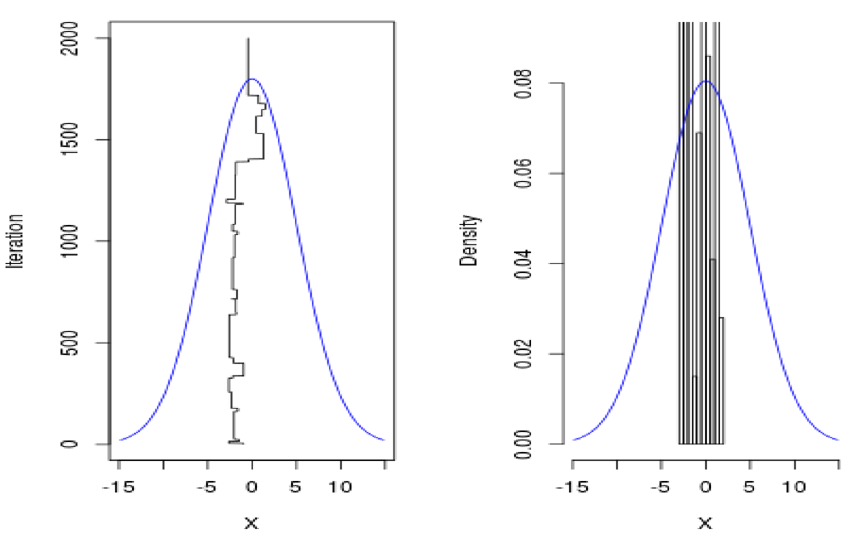
Later generalizations to Langevin diffusions (Roberts & R., JRSSB 1998), and to other targets (Bédard, AAP 2007; Bédard & R., CJS 2008; Yang, Roberts, & R., SPA 2020; ...). Also shows that the computational complexity is $O(d)$. [Roberts & R., J Appl Prob 2016; Yang & R., 2017]

Important? 20-Dimensional Metropolis Example

Case study: Target density π on $\mathbf{R}^{20}$, with Metropolis proposal $N(x, \Sigma)$.

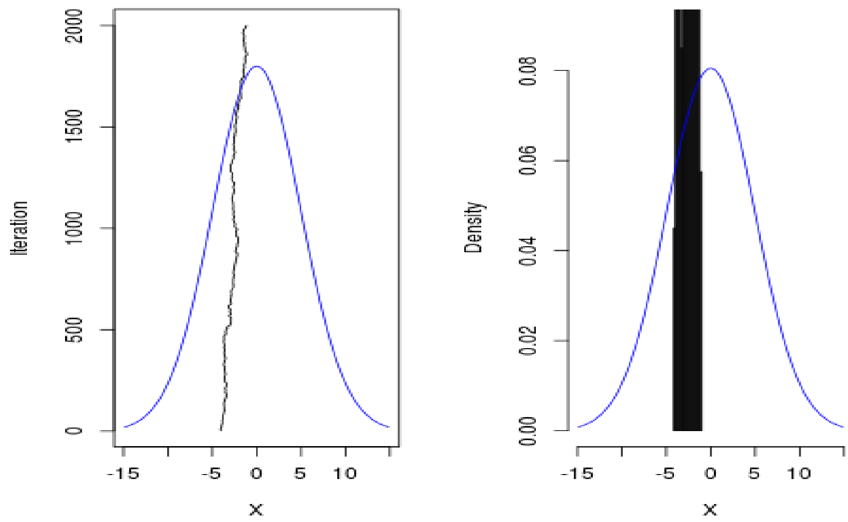
<u>First try:</u> Proposal covariance $\Sigma = I_{20}$?

(Left: trace plot, with "time" moving <u>upwards</u>. Right: histogram.)



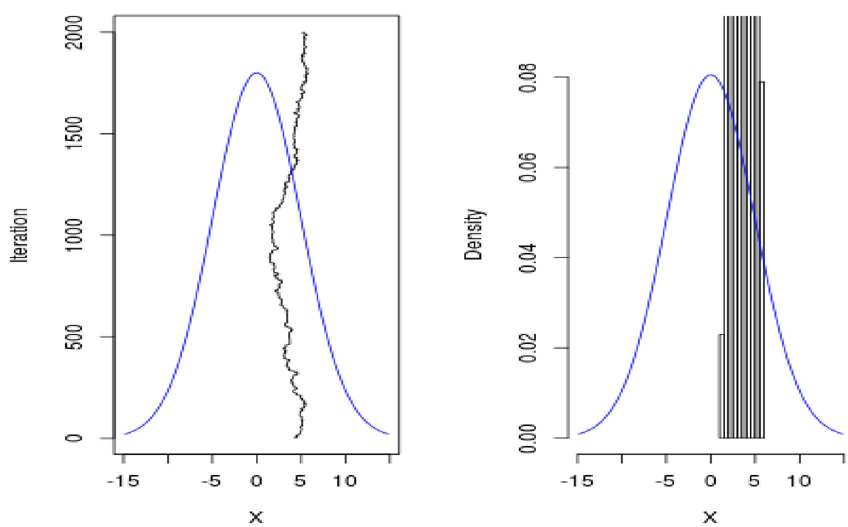
Acceptance rate $= 0.017$. Too low! Need smaller Σ !

Second try: $\Sigma = 0.001 * I_{20}$.



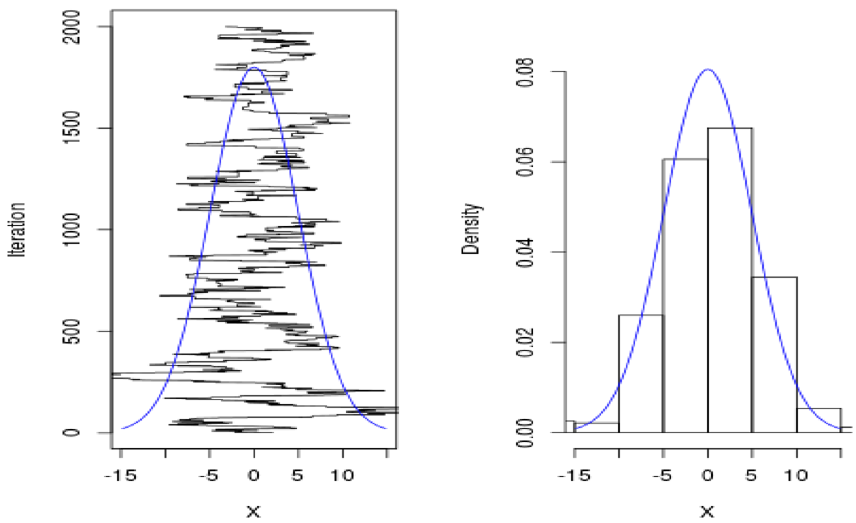
Acceptance rate $= 0.652$. Too high! Need bigger Σ !

Third try: $\Sigma = 0.02 * I_{20}$.



Acceptance rate ≈ 0.234. "Just right".
So, why such poor performance?

Fourth try: $\Sigma = \Sigma_{opt} := \frac{(2.38)^2}{20}\,\Sigma_\pi$, with Σ_π the covariance of π.



Acceptance rate ≈ 0.234 still.
But now the proposal covariance is optimal! Works much better!
Similarly in many other examples, including in hundreds of dimensions.
Optimal proposals make a big difference!

Adaptive MCMC

Recall that (under certain conditions) the optimal proposal covariance is $\Sigma = \frac{(2.38)^2}{d}\,\Sigma_\pi$, with optimal acceptance rate 0.234.

But what if Σ_π is unknown? And what if we don't know what scaling gives 0.234 acceptance rate? How can we make use of this optimality information?

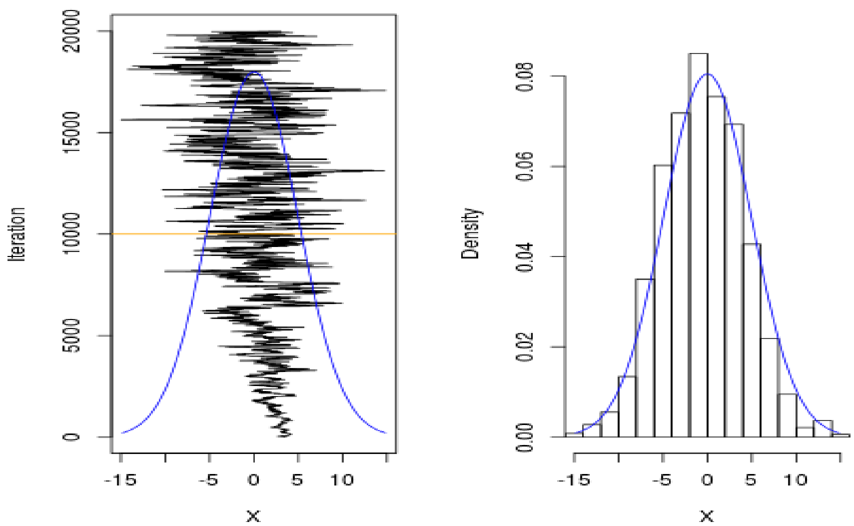
Idea: Replace Σ_π with the empirical estimate Σ_n of the target covariance, based on the run so far. ["Adaptive Metropolis algorithm": Haario et al., 2001; Roberts & R., J Appl Prob 2007, JCGS 2009]

If the run is going well, then Σ_n is a pretty good approximation to Σ_π, so hopefully we still get a nearly-optimal proposal. Good!

But adjusting the run based on the chain's history destroys the Markov property. Bad!

- Does adapting work well in practice? (Yes!)
- Does it still converge eventually to π? (Sometimes!)

Trace plot of first coordinate in a 20-dimensional example:



In 20 dimensions, after about 10,000 iterations, it finds good proposal covariances and starts mixing well. Good.

Similarly good performance in higher dimensions (100, 200, ...), componentwise samplers (dimension 500), variable selection probabilities, etc. [Roberts & R., JCGS 2009; Latuszynski, Roberts, & R., Ann Appl Prob 2013]. Good!

But can we prove that adaptive MCMC still converges to π?
Difficult – no longer Markovian, might fail! [Metropolis]

But still converges under certain assumptions, e.g. "Diminishing Adaptation" (easy) and "Containment" (harder). [Roberts & R., JAP 2007, JCGS 2009; see also Haario, Saksman, Tamminen, Vihola, Andrieu, Moulines, Robert, Fort, Atchadé, Craiu, Bai, Kohn, Giordani, Nott, ...]

Later "adversarial Markov chain" probabilistic arguments can verify Containment [Craiu, Gray, Latuszynski, Madras, Roberts, R., Ann. Appl. Prob. 2015; R. & Yang, submitted]. "Adaptation for everyone"!

Practical alternative: Automatically cease adapting once the adapting has "stabilised", to guarantee convergence. [Yang & R., Comp. Stat. 2017].

Summary

- Monte Carlo and MCMC algorithms (e.g. Metropolis) are very widely used, in many areas, to sample from complicated distributions π. Magic!
- Can sometimes prove quantitative convergence bounds. (Difficult.)
- Can use diffusion limits to establish optimal acceptance rate (0.234; Goldilocks Principle) and proposal covariance $\Sigma_{opt} = \frac{(2.38)^2}{d}\,\Sigma_\pi$.
- Makes a big practical difference to the speed of convergence.
- Can adapt the update rules to converge faster. Works well! And, under appropriate conditions, it is still guaranteed to converge to π.

- Lots more Monte Carlo applications! Lots more theory to develop!

All my papers, simulations, software, info: www.probability.ca

Email jeff@math.toronto.edu / Twitter @ProbabilityProf