Optimising and Adapting Metropolis Algorithms

Jeffrey S. Rosenthal
University of Toronto

jeff@math.toronto.edu
http://probability.ca/jeff/

(LMS/CRiSM Summer School, Warwick, July 2018)

(Brief) Background / Context / Motivation

Often have complicated, high-dimensional density functions $\pi : \mathcal{X} \to [0, \infty)$, for some $\mathcal{X} \subseteq \mathbf{R}^d$ with d large.

(e.g. Bayesian posterior distribution)

Want to compute probabilities like:

$$\Pi(A) \;:=\; \int_A \pi(x)\, dx\,,$$

and/or expected values of functionals like:

$$\mathbf{E}_\pi(h) \;:=\; \int_{\mathcal{X}} h(x)\, \pi(x)\, dx\,.$$

Or, if π is unnormalised:

$$\mathbf{E}_\pi(h) \;:=\; \int_{\mathcal{X}} h(x)\, \pi(x)\, dx \Big/ \int_{\mathcal{X}} \pi(x)\, dx\,.$$

Calculus? Numerical integration?

Impossible, if π is something like . . .

Typical π: Variance Components Model

State space $\mathcal{X} = (0,\infty)^2 \times \mathbf{R}^{K+1}$, so $d = K+3$, with

$$\begin{aligned} &\pi(V, W, \mu, \theta_1, \ldots, \theta_K) \\ &= \; C\, e^{-b_1/V}\, V^{-a_1-1} e^{-b_2/W}\, W^{-a_2-1} \\ &\quad \times e^{-(\mu-a_3)^2/2b_3}\, V^{-K/2} W^{-\frac{1}{2}\sum_{i=1}^{K} J_i} \\ &\times \; \exp\left[-\sum_{i=1}^{K} (\theta_i-\mu)^2/2V - \sum_{i=1}^{K}\sum_{j=1}^{J_i} (Y_{ij}-\theta_i)^2/2W \right], \end{aligned}$$

where a_i and b_i are fixed constants (prior), and $\{Y_{ij}\}$ are the data.

In the application: $K = 19$, so $d = 22$.

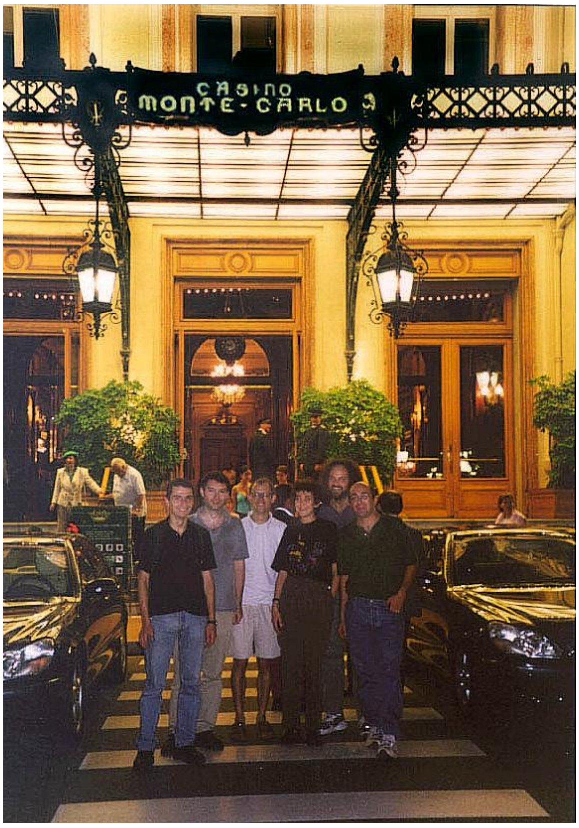
Integrate? Well, no problems *mathematically*, but ...

High-dimensional! Complicated! How to compute?

Try Monte Carlo!

Monte Carlo, Monaco

Nice Place for a Conference!



Estimation from sampling: Monte Carlo

Suppose we can sample from π, i.e. generate on a computer

$$X_1, X_2, \ldots, X_M \sim \pi \quad (i.i.d.)$$

(i.e., $\mathbf{P}(X_i \in A) = \int_A \pi(x)\, dx$ for each i, and independent).

Then can estimate by e.g.

$$\mathbf{E}_\pi(h) \approx \frac{1}{M} \sum_{i=1}^{M} h(X_i)\,.$$

As $M \to \infty$, the estimate converges to $\mathbf{E}_\pi(h)$ (by the Law of Large Numbers), which good error bounds / confidence intervals (by the Central Limit Theorem).

Good. But how to sample from π?

Often infeasible! (e.g. above example!)

Instead . . .

Markov Chain Monte Carlo (MCMC)

Given a complicated, high-dimensional target distribution $\pi(\cdot)$:

Find an ergodic Markov chain (random process) $X_0, X_1, X_2, \ldots$, which is easy to run on a computer, and which converges in distribution to π as $n \to \infty$.

Then for "large enough" B, $\mathcal{L}(X_B) \approx \pi$, so X_B, X_{B+1}, ... are approximate samples from π, and e.g.

$$\mathbf{E}_\pi(h) \approx \frac{1}{M} \sum_{i=B+1}^{B+M} h(X_i)\,, \quad \text{etc.}$$

Extremely popular: Bayesian inference, computer science, statistical genetics, statistical physics, finance, insurance, ...

But how to create such a Markov chain?

Random-Walk Metropolis Algorithm (1953)

This algorithm defines the chain $X_0, X_1, X_2, \ldots$ as follows.

Given X_{n-1}:

- Propose a new state $Y_n \sim Q(X_{n-1}, \cdot)$, e.g. $Y_n \sim N(X_{n-1}, \Sigma_p)$.
- Let $\alpha = \min\left[1, \frac{\pi(Y_n)}{\pi(X_{n-1})}\right]$. (Assuming Q is symmetric.)
- With probability α, accept the proposal (set $X_n = Y_n$).
- Else, with prob. $1-\alpha$, reject the proposal (set $X_n = X_{n-1}$).

Try it: [APPLET] Converges to π!

Why? α is chosen just right so this Markov chain is reversible with respect to π, i.e. $\pi(dx)\,P(x,dy) = \pi(dy)\,P(y,dx)$. Hence, π is a stationary distribution. Also, chain will be aperiodic and (usually) irreducible. So, by general Markov chain theory, it converges to π in total variation distance: $\lim_{n\to\infty} \sup_A |\mathbf{P}(X_n \in A) - \pi(A)| = 0$.

More complicated example?

Example: Particle Systems

Suppose have n independent particles, each uniform on a region.

What is, say, the average "diameter" (maximal distance)?

Sample and see! [pointproc.java] Works! Monte Carlo!

Now suppose instead that the particles are not independent, but rather interact with each other, with the configuration probability proportional to e^{-H}, where H is an energy function, e.g.

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

A large: particles like to be close together.
B large: particles like to be far apart.
C large: particles like to be towards the left.

Can't directly sample, but can use Metropolis! [pointproc.java]

Okay, but Where's the Math?

MCMC's greatest successes have been in . . . applications!

- Medical Statistics / Statistical Genetics / Bayesian Inference / Chemical Physics / Computer Science / Mathematical Finance

So, what is MCMC mathematical theory good for?

- Informs and justifies the basic algorithms. (** Above Introduction)
- Quantifies how well the algorithms work. (** Quantitative Bounds)
- Suggests new modifications of the algorithms.
- Determines which algorithm choices are best. (** Optimal Scaling)
- Investigates high-dimensional behaviour. (** Complexity)
- Develops new MCMC directions. (** Adaptive MCMC)

First Topic: Quantitative Convergence Bounds

MCMC works eventually, i.e. $\mathcal{L}(X_n) \Rightarrow \pi$. Good!

But what about quantitative bounds, i.e. a specific number n_* such that, say, $|\mathbf{P}(X_{n_*} \in A) - \pi(A)| < 0.01 \quad \forall\ A$?

(Not just "as $n \to \infty$".)

One method: coupling. (Many other methods: spectral, . . .)

Consider two copies of the chain, $\{X_n\}$ and $\{X'_n\}$.

Assume that $X'_0 \sim \pi$ (so $X'_n \sim \pi\ \forall n$).

If we can "make" the two copies become equal for $n \geq T$, while respecting their marginal update probabilities, then $X_n \approx \pi$ too.

Specifically, the coupling inequality says:

$$|\mathbf{P}(X_n \in A) - \pi(A)| \ \equiv\ |\mathbf{P}(X_n \in A) - \mathbf{P}(X'_n \in A)| \ \leq\ \mathbf{P}(T > n)\,.$$

But how to apply this to a complicated MCMC algorithm?

Quantitative Bounds: Minorisation

Suppose there is $\epsilon > 0$, and a probability measure ν, such that $P(x, y) \geq \epsilon\, \nu(y)$ for all $x, y \in \mathcal{X}$.

This "minorisation condition" gives an ϵ-sized "overlap" between the transition distributions $P(x, \cdot)$ and $P(x', \cdot)$.

That means at each iteration, we can make the two copies become equal with probability at least ϵ. Hence, $\mathbf{P}(T > n) \leq (1 - \epsilon)^n$.

Therefore, $|\mathbf{P}(X_n \in A) - \pi(A)| \leq\ (1 - \epsilon)^n, \ \ \forall\, A$.

e.g. [APPLET], with that π, and $\gamma = 3$: find that $P(x, y) \geq \epsilon\ \nu(y)$ for all x, y, where $\epsilon = 0.2$, and $\nu(3) = \nu(4) = 1/2$.

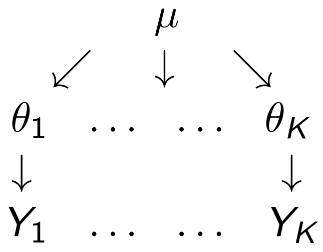
- So $|P^n(x, A) - \pi(A)| \leq (1 - \epsilon)^n = (1 - 0.2)^n = (0.8)^n$.
- Hence, $|P^n(x, A) - \pi(A)| < 0.01$ whenever $n \geq 21$.
- So $n_* = 21$. "The chain converges in 21 iterations." Good!

But what about a harder example??

Example: Baseball Data Model

Hierarchical model for baseball hitting percentages (J. Liu): observed hitting percentages satisfy $Y_i \sim N(\theta_i, c)$ for $1 \le i \le K$, where $\theta_1, \ldots, \theta_k \sim N(\mu, V)$, c is a given constant, with $V, \mu, \theta_1, \ldots, \theta_K$ to be estimated. Priors: $\mu \sim$ flat, $V \sim IG(a, b)$.

Diagram:



$$\theta_i \sim N(\mu, V)$$

$$Y_i \sim N(\theta_i, c)$$

For our data, $K = 18$, so dimension $= 20$.

High dimensional! How to estimate?

Baseball Data Model (cont'd)

MCMC solution: Run a Gibbs sampler for π.

Markov chain is $X_k = (V^{(k)}, \mu^{(k)}, \theta_1^{(k)}, \ldots \theta_K^{(k)})$, updated by:

$$V^{(n)} \ \sim \ IG\left(a + \frac{K-1}{2},\ b + \frac{1}{2}\sum(\theta_i^{(n-1)} - \overline{\theta}^{(n-1)})^2\right) ;$$

$$\mu^{(n)} \ \sim \ N\left(\overline{\theta}^{(n-1)},\ \frac{V^{(n)}}{K}\right) ;$$

$$\theta_i^{(n)} \ \sim \ N\left(\frac{\mu^{(n)}c + Y_i V^{(n)}}{c + V^{(n)}},\ \frac{V^{(n)}c}{c + V^{(n)}}\right) \quad (1 \le i \le K) ;$$

where $\overline{\theta}^{(n)} = \frac{1}{K}\sum \theta_i^{(n)}$.

Recall that $K = 18$, so dimension $= 20$.

Complicated! How to analyze convergence?

Example: Baseball Data Model (cont'd)

Here we can find a minorisation $P(x,y) \geq \epsilon\,\nu(y)$, but only when $x \in C$ for a subset $C \subseteq \mathcal{X}$. ("small set")

But also find a "drift condition" $\mathbf{E}[f(X_1)\,|\,X_0 = x] \leq \lambda f(x) + \Lambda$, for some $\lambda < 1$ and $\Lambda < \infty$, where $f(x) = \sum_{i=1}^{K}(\theta_i - \overline{Y})^2$; this "forces" returns to $C \times C$.

Can compute (R., Stat & Comput. 1996):

- a drift condition towards $C = \left\{\sum_i(\theta_i - \overline{Y})^2 \leq 1\right\}$, with $\lambda = 0.000289$ and $\Lambda = 0.161$;
- a minorisation with $\epsilon = 0.0656$, at least for $x \in C \subseteq \mathcal{X}$.

Then can use coupling to prove (R., JASA 1995) that

$$|\mathbf{P}(X_n \in A) - \pi(A)| \;\leq\; (0.967)^n + (1.17)(0.935)^n\,, \quad n \in \mathbf{N}\,,$$

so e.g. $|\mathbf{P}(X_n \in A) - \pi(A)| < 0.01$ if $n \geq 140$.

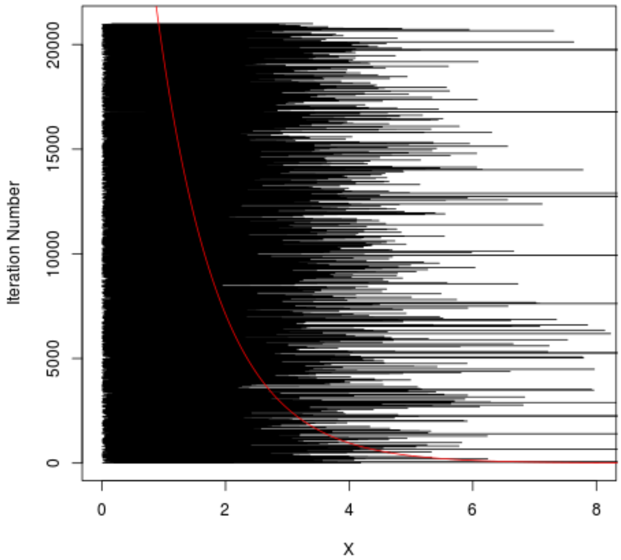
- So $n_* = 140$. "The chain converges in 140 iterations." Good!

Realistic bounds for complicated statistical models!
(See also Jones & Hobert, Stat Sci 2001, . . .)

Does it Matter? Case Study: Independence Sampler

Consider Metropolis-Hastings where $\pi(x) = e^{-x}$, and proposals are chosen i.i.d. $\sim \mathrm{Exp}(k)$ with density ke^{-ky}, for some $k > 0$.

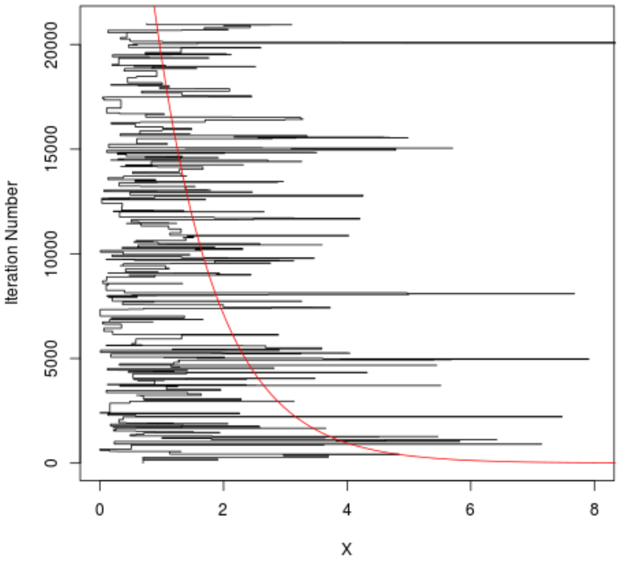
- $k = 1$ (i.i.d. sampling)



$\mathbf{E}(X) = 1$; estimate $= 1.001$. Excellent! Other k?

Independence Sampler (cont'd)

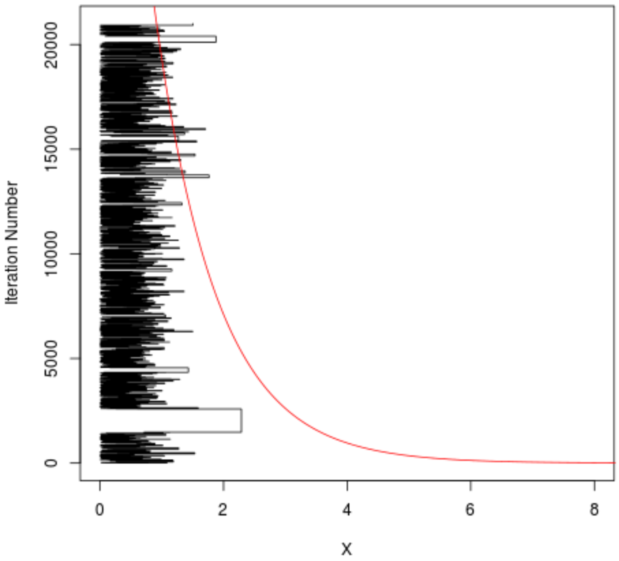
- $k = 0.01$



$\mathbf{E}(X) = 1$; estimate $= 0.993$. Quite good.

Independence Sampler (cont'd)

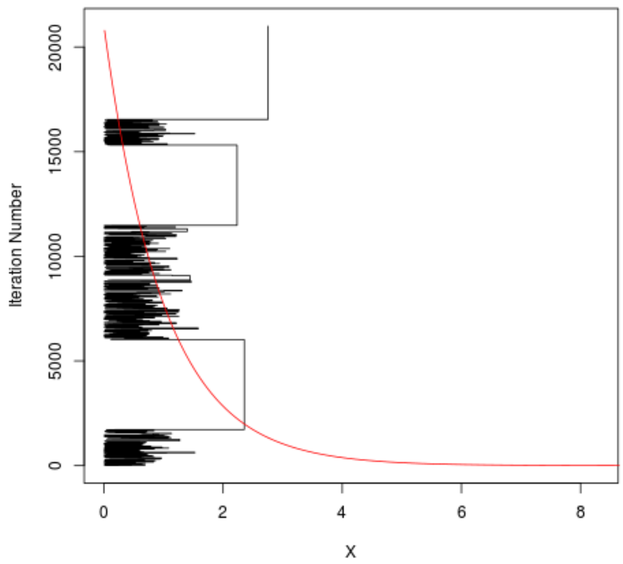
- $k = 5$



$\mathbf{E}(X) = 1$; estimate $= 0.687$. Terrible: way too small!

What happened? Maybe we just got unlucky? Try again!

- Another try with $k = 5$:



$\mathbf{E}(X) = 1$; estimate $= 1.696$. Terrible: way too big!

So, not just bad luck: $k = 5$ is really bad. But why??

Independence Sampler: Theory

Why is $k = 0.01$ pretty good, and $k = 5$ so terrible?

Well, if $k \leq 1$, then $\forall x$, $q(x) = ke^{-kx} \geq ke^{-x} = k\pi(x)$. Then

$$\alpha(x,y) = \min(1, \frac{\pi(y)\, q(x)}{\pi(x)\, q(y)}) = \min(1, \frac{\pi(y)/q(y)}{\pi(x)/q(x)})$$

$$\geq \min(1, \frac{\pi(y)/q(y)}{(1/k)}) = k\,(\pi(y)/q(y))\,.$$

Then $P(x,y) \geq q(y)\,\alpha(x,y) \geq k\,\pi(y)$. Minorisation with $\epsilon = k$!

So, $|P^n(x,A) - \pi(A)| \leq (1-k)^n$.

- $k = 1$: yes, $\epsilon = 1$; converges immediately (of course). $n_* = 1$.
- $k = 0.01$: yes, $\epsilon = 0.01$; and $(1 - 0.01)^{459} < 0.01$, so $n_* = 459$; "chain converges within 459 iterations". (Pretty good.)
- $k = 5$: no such ϵ. Not geometrically ergodic. In fact, we can prove (Roberts and R., MCAP, 2011) that with $k = 5$, have $4,000,000 \leq n_* \leq 14,000,000$, i.e. takes millions of iterations!

Main Topic: How to Optimise MCMC Choices?

In theory, MCMC works with essentially any update rules, as long as they leave π stationary.

- Any symmetric proposal distribution Q. (Choices!)
- Non-symmetric proposals, with a suitably modified acceptance probability. ("Metropolis-Hastings") (e.g. Independent, Langevin)
- Update one coordinate at a time. ("Componentwise")
- Update from full conditional distributions. ("Gibbs Sampler")

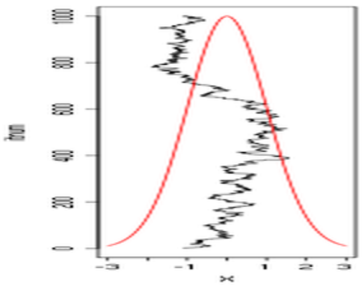
But what choice works best? e.g. What γ in [APPLET]?

- If γ too small (say, $\gamma = 1$), then usually accept, but move very slowly. (Bad.)
- If γ too large (say, $\gamma = 50$), then usually $\pi(Y_{n+1}) = 0$, i.e. hardly ever accept. (Bad.)
- Best γ is between the two extremes, i.e. acceptance rate should be far from 0 and far from 1. ("Goldilocks Principle")

Example: Metropolis for N(0,1)

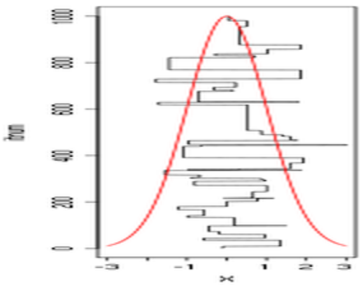
Target $\pi = N(0,1)$. Proposal $Q(x,\cdot) = N(x,\sigma^2)$.

How to choose σ? Big? Small? What acceptance rate (A.R.)?



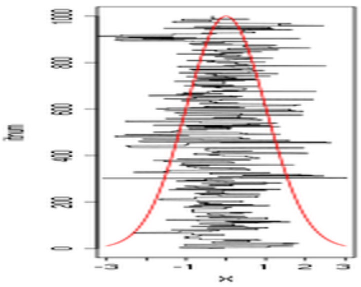
$\sigma = 0.1$?
too small!

A.R. = 0.962



$\sigma = 25$?
too big!

A.R. = 0.052



$\sigma = 2.38$?
just right!

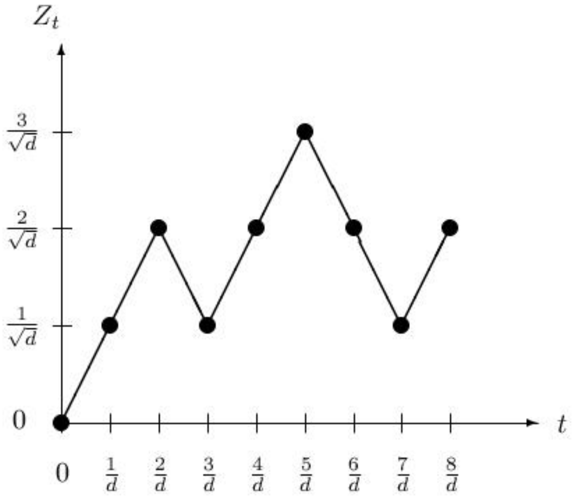
A.R. = 0.441

The Goldilocks Principle in action!

What about higher-dimensional examples? If d increases, then σ should: decrease. But how quickly? On what scale? Theory?

Theoretical Progress: 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). [GRAPHS]



Do similar limits hold for a Metropolis algorithm, in dimension d, as $d \to \infty$? Yes!

Diffusion Limits for the Metropolis Algorithm

[Roberts, Gelman, Gilks, AAP 1997]

- Consider a d-dimensional Metropolis algorithm $\{X_t^d\}_{t \geq 0}$, with proposal distribution $N(x, (\ell^2/d) I_d)$ for some fixed $\ell > 0$ (i.e., with proposal size shrinking as $1/\sqrt{d}$).
- Assume it starts in stationarity, i.e. $X_0^d \sim \pi$.
- Let $U_t^d = X_{PP(td),1}^d$ be the first component of the algorithm, at time $t \times d$ (i.e., U^d is sped up by a factor of d, and is converted to continuous-time via a Poisson Process).
- Assume (for now) that the target density π^d takes on a very special/unrealistic form, namely $\pi^d(x) = \prod_{i=1}^d f(x_i)$ where f is a fixed positive one-dimensional well-behaved (i.e., f'/f Lipschitz, $\mathbf{E}_f[(f'/f)^8] < \infty$, $\mathbf{E}_f[(f''/f)^4] < \infty$) density function.
- Then as $d \to \infty$, the process U^d converges (weakly, in the Skorokhod topology) to a fixed one-dimensional diffusion process U, defined by ...

Diffusion Limits for Metropolis (cont'd)

- This limiting process U has dynamics

$$dU_t \;=\; \sqrt{h(\ell)}\, dB_t \,+\, h(\ell)\,\frac{f'(U_t)}{2\,f(U_t)}\,dt\,,$$

where $h(\ell) = 2\,\ell^2\,\Phi(-\ell\sqrt{\mathcal{I}}\,/2)$ with $\Phi(y) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$ and $\mathcal{I} = \mathbf{E}_f[(f'/f)^2]$.

- The process U is thus a Langevin diffusion, with stationary density f, and "speed" $h(\ell)$.

- Indeed, equivalently, $U_t = V_{h(\ell)\,t}$ is a speeded up (by a factor of $h(\ell)$) version of a Langevin diffusion V of unit speed, satisfying

$$dV_t \;=\; dB_t \,+\, \frac{f'(V_t)}{2\,f(V_t)}\,dt\,.$$

- So, to optimise the algorithm, we should maximise $h(\ell)$.
- The maximisation gives: $\ell_{opt} \doteq 2.38/\sqrt{\mathcal{I}}$.
- Then we compute that: $AR(\ell_{opt}) \doteq 0.234$. (constant!)

Diffusion Limits for Metropolis (cont'd)

- So, for a Metropolis algorithm in d dimensions, with $Q(x,\cdot) = N(x, \sigma^2 I_d)$, it is optimal to choose $\sigma^2 = \ell_{opt}^2\,/\,d$ $\doteq (2.38)^2\,/\,\mathcal{I}d$, corresponding to an (optimal) acceptance rate of 0.234. Clear, simple "0.234" rule. Good! Useful! (Used in BUGS!)

- The unrealistic form of π^d was later generalised to: inhomogeneous product form (Bédard & R., CJS 2008), infinite-dimensional absolutely continuous distributions (Stuart et al.), discrete hypercubes (Roberts, Stoch Rep 1998), spherical targets (Neal and Roberts, MCAP 2008), elliptically symmetric targets (Sherlock and Roberts, Bernoulli 2009), and discontinuous targets (Neal et al., AAP 2012).

- Numerical studies (e.g. Roberts and R., Stat Sci 2001): same optimality appears to "approximately" hold for more general π^d.

- Different optimal AR of 0.574 for Langevin diffusion algorithms (Roberts & R., JRSSB 1998).

New Generalisations?

(Yang, Negrea, Roberts, R., work in progress)

In the original RGG result, the unrealistic i.i.d. nature of π^d was used to apply Laws of Large Numbers when taking limits of the generators of the processes U^d.

Can the same proof techniques be used under weaker conditions?

It appears that if, as $d \to \infty$:

- in π^d, the dependence of x_1 on $x_2, \ldots, x_d$ goes to zero, and
- π^d and its derivatives satisfy strong moment order bounds,

then diffusion limits similar to the i.i.d. case still hold.

In particular, 0.234 is still the optimal acceptance rate.

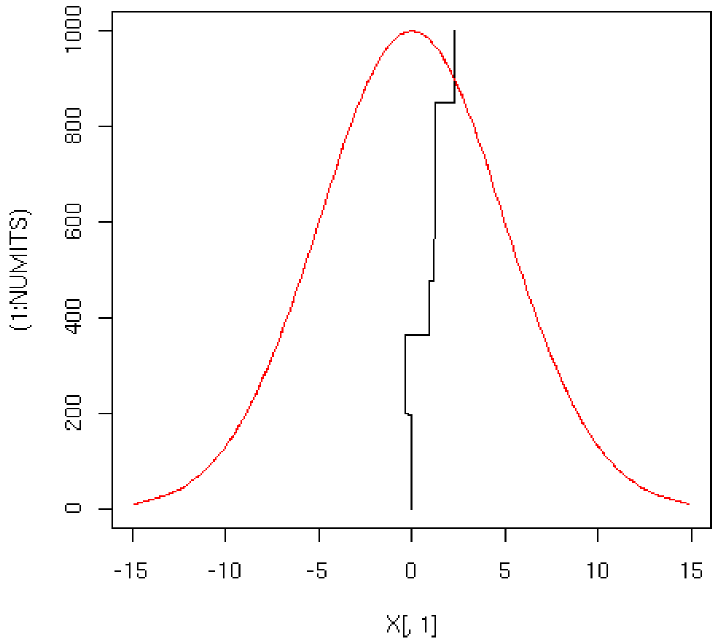
Anyway, 0.234 is a very useful rule of thumb.

But it is just a "one-dimensional" guideline.

What about further optimality, beyond "0.234"?

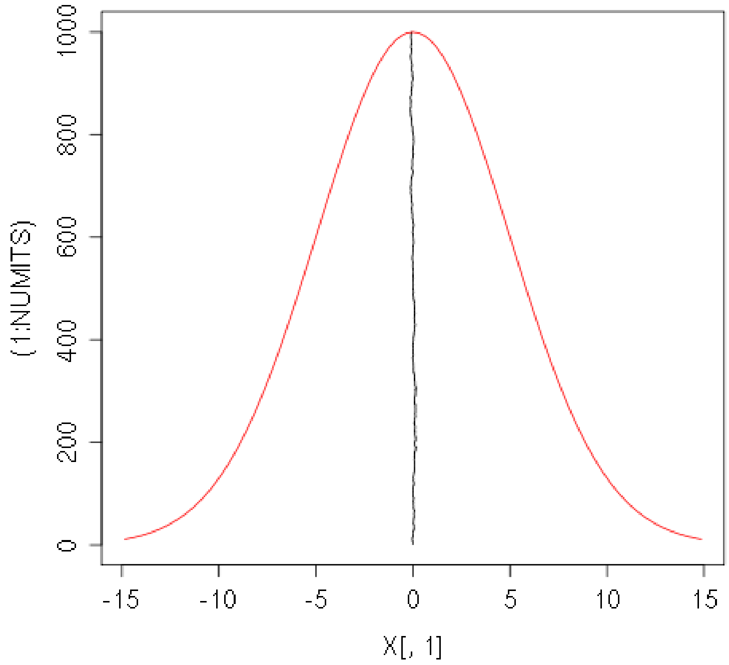
Example: $\pi = N(0, \Sigma)$ in dimension 20

First try: $Q(x, \cdot) = N(x, I_{20})$ (A.R. $= 0.006$)



Horrible: $\Sigma_{11} = 24.54$, $E(X_1^2) = 1.50$. Need smaller proposal!

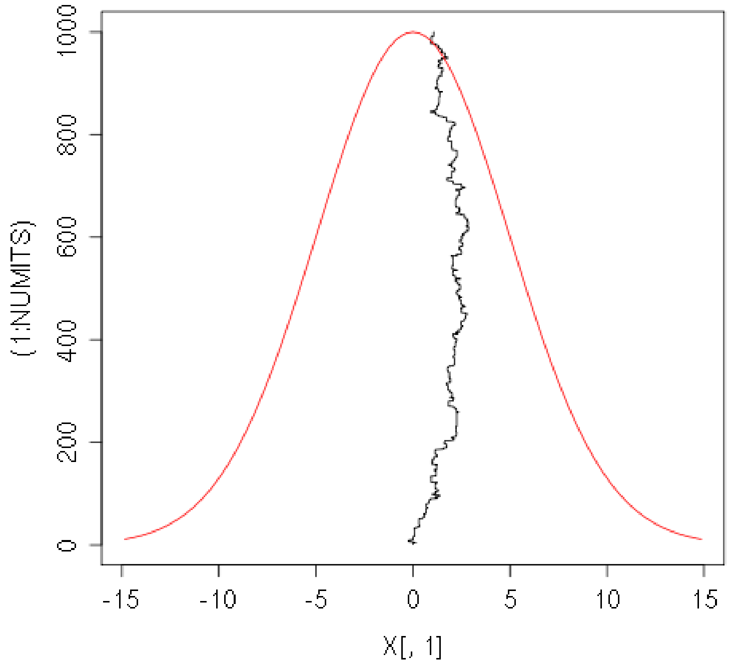
Second try: $Q(x, \cdot) = N\Big(x, (0.0001)^2 I_{20}\Big)$ (A.R.=0.9996)



Also horrible: $\Sigma_{11} = 24.54$, $E(X_1^2) = 0.0053$.

Need bigger proposal!

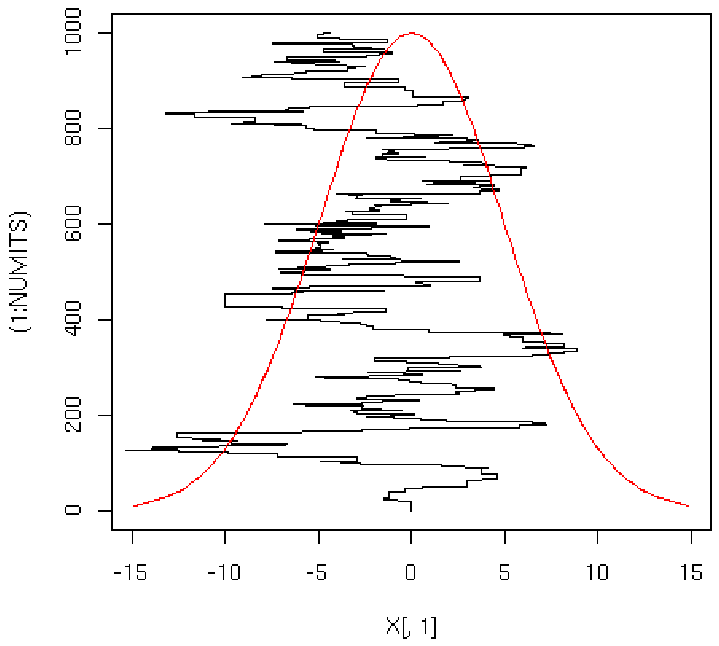
Third try: $Q(x, \cdot) = N\Big(x, (0.02)^2 I_{20}\Big)$ (A.R.=0.234)



Still terrible: $\Sigma_{11} = 24.54$, $E(X_1^2) = 3.63$.

But acceptance rate is "just right". What gives?

Fourth try: $Q(x,\cdot) = N\Big(x, [(2.38)^2/20]\,\Sigma\Big)$ (A.R.=0.263)



Much better: $\Sigma_{11} = 24.54$, $E(X_1^2) = 25.82$.

Not perfect, but fairly good. Why?

Optimising the Proposal Covariance (Shape)

Theorem [Roberts and R., Stat Sci 2001]: If π is any orthogonal transform of any density satisfying the RGG conditions, then the optimal Gaussian proposal distribution as $d \to \infty$ is:

$$Q(x,\cdot) \;=\; N\Big(x,\; [(2.38)^2/d]\,\Sigma_t\Big)$$

where Σ_t is the target covariance. (Not $N(x, \sigma^2 I_d)$.)

So, want proposal covariance proportional to target covariance!

The corresponding asymptotic acceptance rate is again 0.234.

This turns out to be nearly optimal for many other high-dimensional densities, too. Very useful advice ... if Σ_t is known!

But what if the target covariance Σ_t is unknown?

Can we make use of this optimality result anyway?

Perhaps ... if we "adapt" ... (coming soon!).

Implications for Computational Complexity

- Above results say, if we speed up the Metropolis algorithm by a factor of $O(d)$, then it converges to a dimension-free diffusion, and hence must converge in time $O(1)$.

- So, this seems to imply that Metropolis converges in $O(d)$. Right?

- Problem #1: Result is only for very special forms of the target π. (But we're working to generalise this!)

- Problem #2: Result just gives weak convergence, not total variation distance. (But we can work with that!)

- Problem #3: How to define computational complexity on continuous unbounded state spaces? What initial distribution should be used? (Can't use "worst case".)

What to do?

Weak Convergence Complexity Result

- Use the Kantorovich-Rubinstein (KR) distance measure,

$$\|\mathcal{L}_x(X_t) - \pi\|_{KR} \;:=\; \sup_{f \in \mathrm{Lip}_1^1} \Big| \mathbf{E}_x[f(X_t) - \pi(f) \Big|$$

where $\mathrm{Lip}_1^1 = \{f : \mathcal{X} \to \mathbf{R},\ |f(x)| \le 1,\ |f(x) - f(y)| \le dist(x,y)\}$, which metricises weak convergence.

- And average over starting values $X_0 \sim \pi$, i.e. use

$$\mathbf{E}_{X_0 \sim \pi} \|\mathcal{L}_{X_0}(X_t) - \pi\|_{KR} \;:=\; \int_{x \in \mathcal{X}} \pi(dx)\, \|\mathcal{L}_x(X_t) - \pi\|_{KR}\,.$$

- Theorem [Roberts and Rosenthal, JAP 2016]: If $X^{(d)} \to X^{(\infty)}$ weakly, for any choice of $X_0^{(d)}$, and $X^{(\infty)}$ is càdlàg (or continuous), and $X^{(\infty)} \to \pi$, then $\mathbf{E}_{X_0^{(d)} \sim \pi} \|\mathcal{L}_{X_0^{(d)}}(X_t^{(d)}) - \pi\|_{KR} \to 0$ in $O(1)$ time, i.e. for any $\epsilon > 0$, there are $D < \infty$ and $T < \infty$ such that

$$\mathbf{E}_{X_0^{(d)} \sim \pi} \|\mathcal{L}_{X_0^{(d)}}(X_t^{(d)}) - \pi\|_{KR} < \epsilon\,, \quad \forall\, t \ge T,\ d \ge D\,.$$

Computational Complexity of Metropolis

- Combining this complexity result with the Metropolis weak convergence results immediately shows that:
  - The speeded-up processes U_t^d converge to π in $O(1)$ time.
- But U_t^d equals the original Metropolis algorithm's first coordinate process $X_{n,1}^d$, sped up by a factor of d.
- Hence, the original Metropolis algorithm's first coordinate process $X_{n,1}^d$ must converge to π in $O(d)$ time.
- Hence, the Metropolis algorithm converges (coordinatewise at least) in time $O(d)$. Right?
- One technicality: we need weak convergence from any starting point X_0, not from stationarity $X_0 \sim \pi$ ... but that also holds if the powers of the target density f in the moment assumptions are increased slightly (from 8 and 4, to 12 and 6). Phew!
- Also, still requires unrealistic conditions on π ... but we're working on that. Then have: convergence in $O(d)$ iterations!

How to Use the Optimality Information?

Recall: We have guidance about optimising MCMC in terms of acceptance rate, target covariance matrix Σ_t, etc.

In particular:

1. Want acceptance rate around 0.234.

2. Optimal Gaussian RWM proposal is $N\Big(x,\, (2.38)^2\, d^{-1}\, \Sigma_t\Big)$, where Σ_t is the covariance matrix of the target π.

Great, except ... we don't know what proposal will lead to a desired acceptance rate. And, we don't know how to compute Σ_t.

So, what to do?

Trial and error? (difficult, especially in high dimension)

Or ... let the computer decide, on the fly!

Adaptive MCMC

Suppose we have a family $\{P_\gamma\}_{\gamma \in \mathcal{Y}}$ of possible Markov chains, each with stationary distribution π.

Let the computer choose among them!

At iteration n, use Markov chain P_{Γ_n}, where $\Gamma_n \in \mathcal{Y}$ chosen according to some adaptive rules (depending on chain's history, etc.). [APPLET]

Can this help us to find better Markov chains? (Yes!)

On the other hand, the Markov property, stationarity, etc. are all destroyed by using an adaptive scheme.

Is the resulting algorithm still ergodic? (Sometimes!)

We begin with some simulation examples …

Example: High-Dimensional Adaptive Metropolis

Dim $d = 100$, with target π having target covariance Σ_t.
Here Σ_t is 100×100 (i.e., 5,050 distinct entries).

Here optimal Gaussian RWM proposal is $N\Big(x,\, (2.38)^2\, d^{-1}\, \Sigma_t\Big)$.

But usually Σ_t unknown. Instead use empirical estimate, Σ_n, based on the observations so far $(X_1, X_2, \ldots, X_n)$. Then let

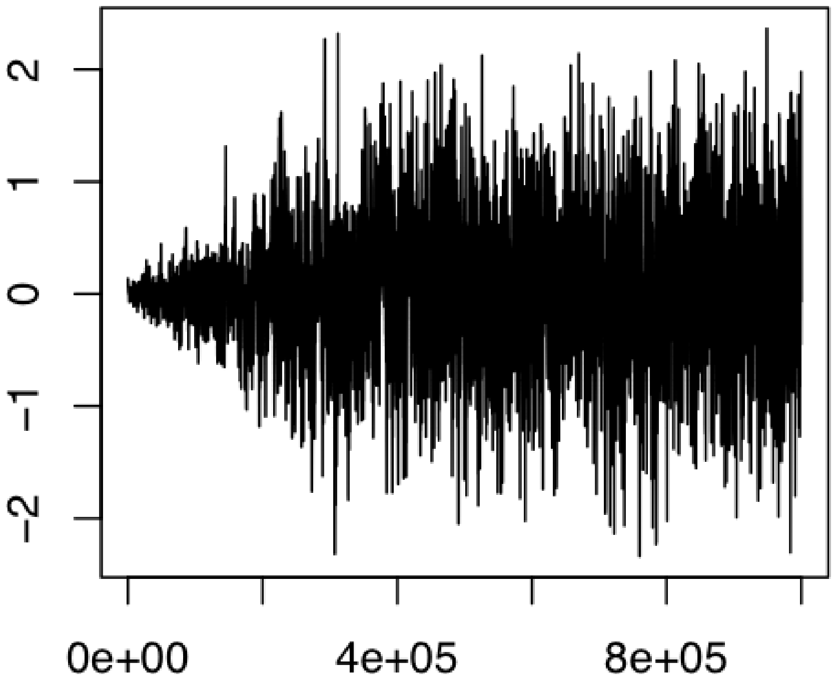
$$Q_n(x,\cdot) \;=\; (1-\beta)\, N\Big(x,\, (2.38)^2\, d^{-1}\, \Sigma_n\Big) + \beta\, N\Big(x,\, (0.1)^2\, d^{-1}\, I_d\Big)\,,$$

where e.g. $\beta = 0.05$.

(Slight variant of the algorithm of Haario et al., Bernoulli 2001.)

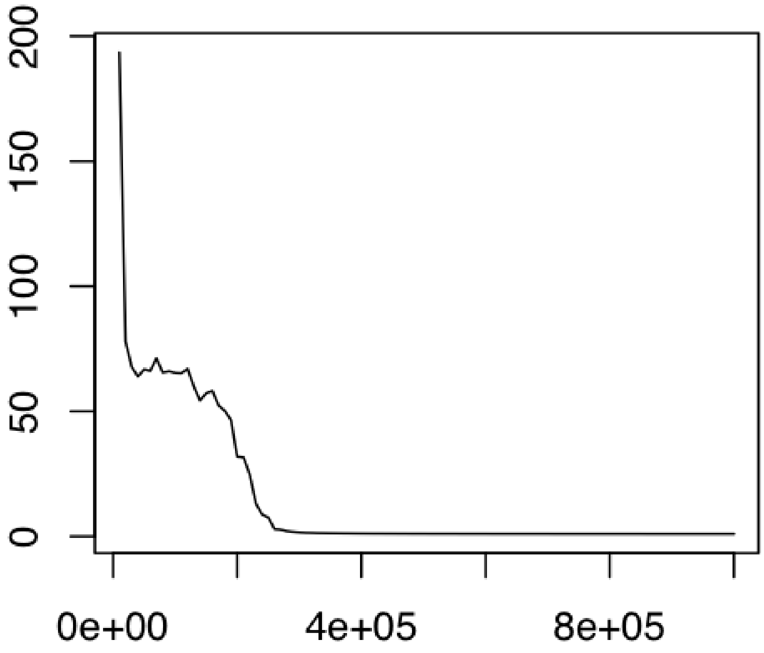
Let's try it …

High-Dimensional Adaptive Metropolis (cont'd)



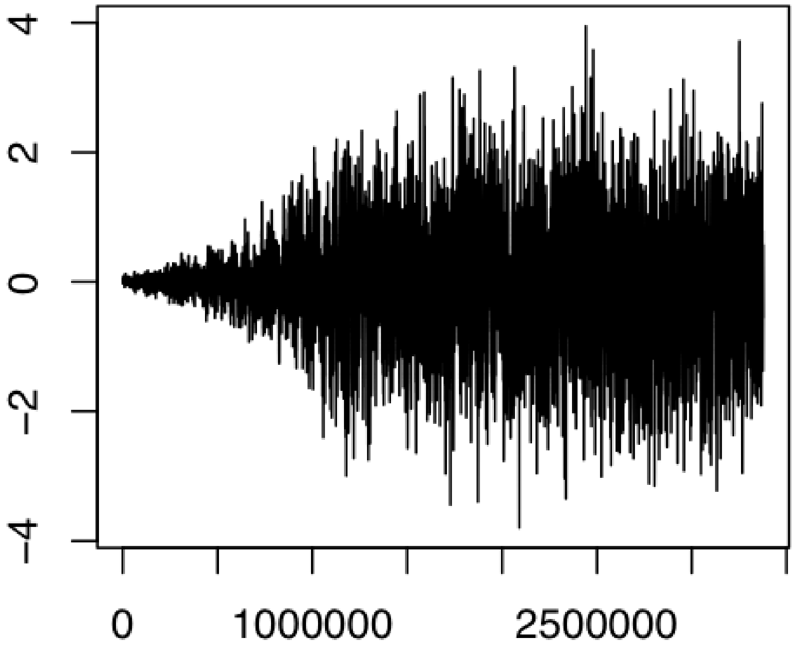
Plot of first coord. Takes about 300,000 iterations, then "finds" good proposal covariance and starts mixing well.

High-Dimensional Adaptive Metropolis (cont'd)



Plot of sub-optimality factor $b_n \equiv d\left(\sum_{i=1}^{d}\lambda_{in}^{-2}\,/\,(\sum_{i=1}^{d}\lambda_{in}^{-1})^2\right)$, where $\{\lambda_{in}\}$ eigenvals of $\Sigma_n^{1/2}\,\Sigma^{-1/2}$. Starts large, converges to 1.

Even Higher-Dimensional Adaptative Metropolis



In dimension 200, takes about 2,000,000 iterations, then finds good proposal covariance and starts mixing well.

Another Example: Componentwise Adaptive Metropolis

Propose new value $y_i \sim N(x_i,\, e^{2\, ls_i})$ for the i^{th} coordinate, leaving the other coordinates fixed; then repeat for different i.

Choice of scaling factor ls_i?? (i.e., "$\log(\sigma_i)$")

Recall: optimal one-dim acceptance rate is ≈ 0.44. So:

Start with $ls_i \equiv 0$ (say).

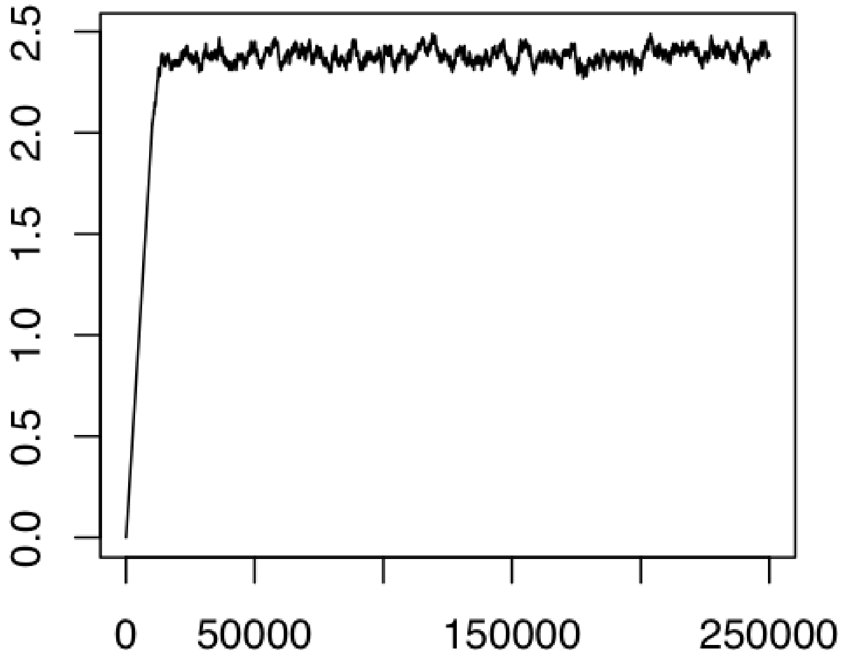
Adapt each ls_i, in batches, to seek 0.44 acceptance rate:

After the j^{th} batch of 100 (say) iterations, decrease each ls_i by $1/j$ if the acceptance rate of the i^{th} coordinate proposals is < 0.44, otherwise increase it by $1/j$.

Let's try it . . .

Adaptive Componentwise Metropolis (cont'd)

Test on Variance Components Model, with $K = 500$ (dim=503), J_i chosen with $5 \le J_i \le 500$, and simulated data $\{Y_{ij}\}$.



Adaption quickly finds "good" values for the ls_i values.

Great ... but is it Ergodic?

Adaptive MCMC seems to work well in practice.

But will it be ergodic, i.e. converge to π?

Ordinary MCMC algorithms, i.e. with fixed choice of γ, are automatically ergodic by standard Markov chain theory (since they're irreducible and aperiodic and leave π stationary).

But adaptive algorithms are more subtle, since the Markov property and stationarity are destroyed by the adaptive scheme. [APPLET]

WANT: Simple conditions guaranteeing $\|\mathcal{L}(X_n) - \pi\| \to 0$, where $\|\mathcal{L}(X_n) - \pi\| \equiv \sup_{A \subseteq \mathcal{X}} |\mathbf{P}(X_n \in A) - \pi(A)|$.

(Alternative: Just do "finite adaptation" and diagnose when to stop, e.g. Yang & R., Comp. Stat. 2017; R package "atmcmc".)

One Simple Convergence Theorem

THEOREM [Roberts and R., J.A.P. 2007]: An adaptive scheme using $\{P_\gamma\}_{\gamma \in \mathcal{Y}}$ will converge, i.e. $\lim_{n\to\infty} \|\mathcal{L}(X_n) - \pi\| = 0$, if:

(a) [Diminishing Adaptation] Adapt less and less as the algorithm proceeds. Formally, $\sup_{x \in \mathcal{X}} \|P_{\Gamma_{n+1}}(x,\cdot) - P_{\Gamma_n}(x,\cdot)\| \to 0$ in prob.

[Can always be made to hold, since adaption is user controlled.]

(b) [Containment] Times to stationary from X_n, if fix $\gamma = \Gamma_n$, remain bounded in probability as $n \to \infty$. [Technical condition, to avoid "escape to infinity". Holds if e.g. $\mathcal{X}$ and $\mathcal{Y}$ finite, or compact, or sub-exponential tails, or ... (Bai, Roberts, and R., Adv. Appl. Stat. 2011). And always seems to hold in practice.]

(Also guarantees WLLN for bounded functionals. Various other results about LLN / CLT under stronger assumptions.)

Other results by: Haario, Saksman, Tamminen, Vihola; Andrieu, Moulines, Robert, Fort, Atchadé; Kohn, Giordani, Nott; ...

Outline of Proof (one page only!)

Define a second chain $\{X'_n\}$, which begins like $\{X_n\}$, but which stops adapting after time N. ("coupling")

Containment says that the (ordinary MCMC) convergence times are bounded, so that for large enough M, we "probably" have $\mathcal{L}(X'_{N+M}) \approx \pi(\cdot)$, i.e. $\mathbf{P}(X'_{N+M} \in A) \approx \pi(A)$ for all A and N.

And, Diminishing Adaptation says that we adapt less and less, so that for large enough N (depending on M),

$$(X_N, X_{N+1}, \ldots, X_{N+M}) \approx (X'_N, X'_{N+1}, \ldots, X'_{N+M}).$$

Combining these, for large enough N and M, we "probably" have

$$\mathcal{L}(X_{N+M}) \approx \mathcal{L}(X'_{N+M}) \approx \pi(\cdot), \quad \text{Q.E.D.}$$

Implications of Theorem

Adaptive Metropolis algorithm:

- Empirical estimates satisfy Diminishing Adaptation.
- And, Containment easily guaranteed if we assume π has bounded support (Haario et al., 2001), or sub-exponential tails (Bai, Roberts, and R., 2011).
- COR: Adaptive Metropolis is ergodic under these conditions.

Adaptive Componentwise Metropolis:

- Satisfies Diminishing Adaption, since adjustments $\pm 1/j \to 0$.
- Satisfies Containment under boundedness or tail conditions.
- COR: Ad. Comp. Metr. also ergodic under these conditions.

So, previous adaptive algorithms work (at least asymptotically).

Similar convergence results for: regional adaptation (Craiu, R., C. Yang, JASA 2009), and adaptive multiple-try Metropolis (J. Yang, Levi, Craiu, R., under revision). Good!

Choosing Which Coordinates to Update When

S. Richardson (statistical geneticist): Successfully ran adaptive Componentwise Metropolis algorithm on genetic data with thousands of coordinates. Good!

But many of the coordinates are binary, and usually do not change.

She asked: Do we need to visit every coordinate equally often, or can we gradually "learn" which ones usually don't change and downweight them? Good question – how to proceed?

Suppose at each iteration n, we choose to update coordinate i with probability $\alpha_{n,i}$, and then we update the random-scan coordinate weights $\{\alpha_{n,i}\}$ on the fly.

What conditions ensure ergodicity?

Seemed hard! Then we found a claim [J. Mult. Anal. **97** (2006), p. 2075]: Suffices that $\lim_{n\to\infty} \alpha_{n,i} = \alpha_i^*$, where the Gibbs sampler with fixed weights $\{\alpha_i^*\}$ is ergodic.

Really?? No, counter-example! (K. Latuszyński)

Ergodicity with Adaptive Coordinate Weights

So, we had to be smarter than that!

We proved (Latuszynski, Roberts, and R., Ann. Appl. Prob. 2013) that adaptively weighted samplers are ergodic if either:

(i) some choice of weights $\{\alpha_i^*\}$ make it uniformly ergodic, or

(ii) there is simultaneous inward drift for all the kernels P_γ, i.e. there is $V : \mathcal{X} \to [1, \infty)$ with

$$\limsup_{|x|\to\infty} \sup_{\gamma\in\mathcal{Y}} \frac{(P_\gamma V)(x)}{V(x)} < 1 .$$

Then, by being careful about continuity, boundedness, etc., can guarantee ergodicity in many cases, including for high-dimensional genetics data (Richardson, Bottolo, R., Valencia 2010). Good!

What about that "Containment" Condition?

Recall: adaptive MCMC is ergodic if it satisfied Diminishing Adaptation (easy: user-controlled) and Containment (technical).

Is Containment just an annoying artifact of the proof? No!

THEOREM (Latuszynski and R., J.A.P. 2014): If an adaptive algorithm does not satisfy Containment, then it is "infinitely inefficient": that is, for all $\epsilon > 0$,

$$\lim_{L\to\infty} \limsup_{n\to\infty} \mathbf{P}(M_\epsilon(X_n, \gamma_n) > L) > 0 ,$$

where $M_\epsilon(x, \gamma) = \inf\{n \geq 1 : \|P_\gamma^n(x, \cdot) - \pi(\cdot)\| < \epsilon\}$ is the time to converge to within ϵ of stationarity. Bad!

Conclusion: Yay Containment!?!

But how to verify it??

A Method For Verifying Containment

(Craiu, Gray, Latuszynski, Madras, Roberts, and R., A.A.P. 2015)

- We first proved general theorems about stability of "adversarial" Markov chains under various conditions.

- Suppose a random process $\{X_n\}$ on $\mathcal{X}$ satisfies:

$\Rightarrow$ We always have $\mathrm{dist}(X_{n+1}, X_n) \leq D$, for some fixed (large) constant $D < \infty$.

$\Rightarrow$ Outside of some fixed (large) bounded subset $K \subseteq \mathcal{X}$, $\{X_n\}$ follows a fixed ergodic Markov transition kernel P.

(But within K, an adversary can make it do anything ...)

$\Rightarrow$ There is a fixed probability measure μ_* on $\mathcal{X}$ with $P(x, dz) \leq M\, \mu_*(dz)$, and $P^{n_0}(x, dz) \geq \epsilon\, \mu_*(dz)$, for $x \in K_{2D} \setminus K$.

THEOREM: Then $\{X_n\}$ is tight, i.e. the sequence $\{\mathrm{dist}(X_n, \mathbf{0})\}_{n=0}^{\infty}$ remains bounded in probability as $n \to \infty$.

Verifying Containment (cont'd)

- We then applied this to adaptive MCMC, to get a list of directly-verifiable conditions which guarantee Containment:

$\Rightarrow$ Never move more than some (big) distance D.

$\Rightarrow$ Outside (big) rectangle K, use fixed kernel (no adapting).

$\Rightarrow$ The transition or proposal kernels have continuous densities wrt Lebesgue measure. (or piecewise continuous: Yang & R. 2015)

$\Rightarrow$ The fixed kernel is bounded, above and below (on compact regions, for jumps $\leq \delta$), by constants times Lebesgue measure. (Easily verified under continuity assumptions.)

- Can directly verify these conditions in practice.
- So, this can be easily used by applied MCMC users.
- "Adaptive MCMC for everyone!"

See also the nice recent "AIR MCMC" approach of Chimisov, Latuszynski, and Roberts, arXiv 2018.

Summary

- MCMC is extremely popular for estimating expectations.
- Basic Markov chain theory establishes convergence.
- Quantitative convergence bounds can sometimes be obtained using coupling with minorisation (and drift) conditions.
- Rescaled MCMC sometimes converges to diffusion limits.
- MCMC can be optimised by maximising the speed.
- Metropolis (with special forms of π) has an explicit maximisation, corresponding to AR $= 0.234$.
- Best proposal covariance is proportional to the target π.
- Weak convergence implies computation complexity is $O(d)$.
- Working on extending the diffusion limits to more general target distributions.
- But how to use the optimality information?

Summary (cont'd)

- Adaptive MCMC tries to "learn" how to sample better. Good.
- Works well in examples like Adaptive Metropolis (200 × 200 covariance) and Componentwise Metropolis (503 dimensions).
- But must be done carefully, or it will destroy stationarity. Bad.
- To converge to π, suffices to have stationarity of each P_γ, plus (a) Diminishing Adaptation (important), and (b) Containment (technical condition, usually satisfied, necessary). Good.
- This can demonstrate convergence for adaptive Metropolis, coordinatewise adaptation, adaptive coordinate weights, etc.
- New "adversarial" conditions more easily verify Containment.
- Hopefully can use adaption on many other examples – try it!

All my papers, applets, software: probability.ca/jeff