Adaptive MCMC For Everyone

Jeffrey S. Rosenthal, University of Toronto

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

• G.O. Roberts and J.S. Rosenthal, Coupling and Ergodicity of Adaptive MCMC. J. Appl. Prob. **44** (2007), 458–475. • G.O. Roberts and J.S. Rosenthal, Examples of Adaptive MCMC. J. Comp. Graph. Stat. **18** (2009), 349–367. • Y. Bai, G.O. Roberts, and J.S. Rosenthal, On the Containment Condition for Adaptive Markov Chain Monte Carlo Algorithms. Adv. Appl. Stat. **21(1)** (2011), 1–54. • S. Richardson, L. Bottolo, and J.S. Rosenthal, Bayesian Models for Sparse Regression Analysis of High Dimensional Data. *Bayesian Statistics 9* (Valencia 2010). • K. Latuszynski, G.O. Roberts, and J.S. Rosenthal, Adaptive Gibbs Samplers and Related MCMC Methods. Ann. Appl. Prob. **23(1)** (2013), 66–98. • K. Latuszynski and J.S. Rosenthal, The Containment Condition and AdapFail Algorithms. J. Appl. Prob. **51(4)** (2014), 1189–1195. • R.V. Craiu, L. Gray, K. Latuszynski, N. Madras, G.O. Roberts, and J.S. Rosenthal, "Stability of Adversarial Markov Chains, with an Application to Adaptive MCMC Algorithms". Ann. Appl. Prob. **25(6)** (2015), 3592–3623.

Background / 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:

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

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. e.g. $K = 19$, so $d = 22$.

High-dimensional! Complicated! How to compute?

Estimation from sampling: Monte Carlo

Can try to sample from π, i.e. generate on a computer

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

(meaning that $\mathbf{P}(X_i \in A) = \int_A \pi(x)\, dx$).

Then can estimate by e.g.

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

Good. But how to sample? Often infeasible!

Instead ...

Markov Chain Monte Carlo (MCMC)

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

define an ergodic Markov chain (random process) $X_0, X_1, X_2, \ldots$, which converges in distribution to $\pi(\cdot)$.

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

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

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

How to find the good chains among the bad ones?

Ex.: Random-Walk Metropolis Algorithm (1953)

Define 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 $\alpha = \min\left[1, \; \frac{\pi(Y_n)}{\pi(X_{n-1})}\right]$.
- With probability α, accept the proposal (set $X_n = Y_n$).
- Else, with prob. $1 - \alpha$, reject the proposal (set $X_n = X_{n-1}$).

FACT: α is chosen just right so this Markov chain is reversible with respect to $\pi(\cdot)$. Hence, $\pi(\cdot)$ is a stationary distribution.

Also aperiodic and (usually) irreducible.

So, $X_n \to \pi(\cdot)$. [APPLET]

Optimising MCMC?

What choices are optimal for MCMC?

e.g. Metropolis: what is optimal choice of proposal $Q(X_{n-1}, \cdot)$?

Even if $Q(X_{n-1}, \cdot) = N(X_{n-1}, \Sigma_p)$, what is smart choice of Σ_p?

Even if $\Sigma_p = \sigma I$, how large should σ be?

Important – can vary from efficient to infeasible! [APPLET]

Idea: Can we use acceptance rate for guidance?

Intuition: For Metropolis algorithms, want acceptance rate to be far from zero (so it doesn't get stuck), and also far from one (so it tries to take big steps).

More precisely?

Optimising MCMC (cont'd)

What is known about optimising MCMC?

Dim=1: Numerical studies: want acceptance rate ≈ 0.44.

Large dim ($d \to \infty$): Use diffusion limits! (Roberts-Gelman-Gilks 1997, Roberts-R. 2001, Bédard 2006, ...)

Under various strong assumptions, as $d \to \infty$ the algorithm will converge (after rescaling) to an explicit diffusion process. So, choose proposal to maximise the speed of the limiting diffusion.

Conclusions:

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 $\pi(\cdot)$.

Can We USE This Optimality Information?

So, we have guidance about optimising MCMC in terms of acceptance rate, target covariance matrix Σ_t, etc.
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!

Specifically, suppose we have a family $\{P_\gamma\}_{\gamma \in \mathcal{Y}}$ of possible Markov chains, each with stationary distribution $\pi(\cdot)$. 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!)

Example: High-Dimensional Adaptive Metropolis

Dim $d = 100$, with target $\pi(\cdot)$ 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

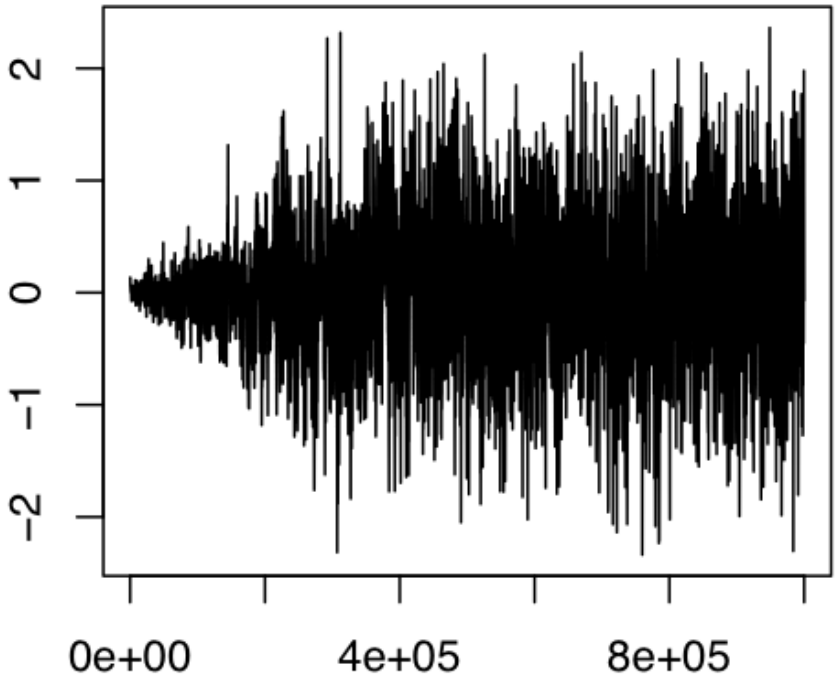
$$Q_n(x, \cdot) \;=\; (1-\beta)\, N\Big(x,\, (2.38)^2\, d^{-1}\, \Sigma_n\Big) + \beta\, N\Big(x,\, (0.1)^2\, d^{-1}\, I_d\Big)\,,$$

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

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

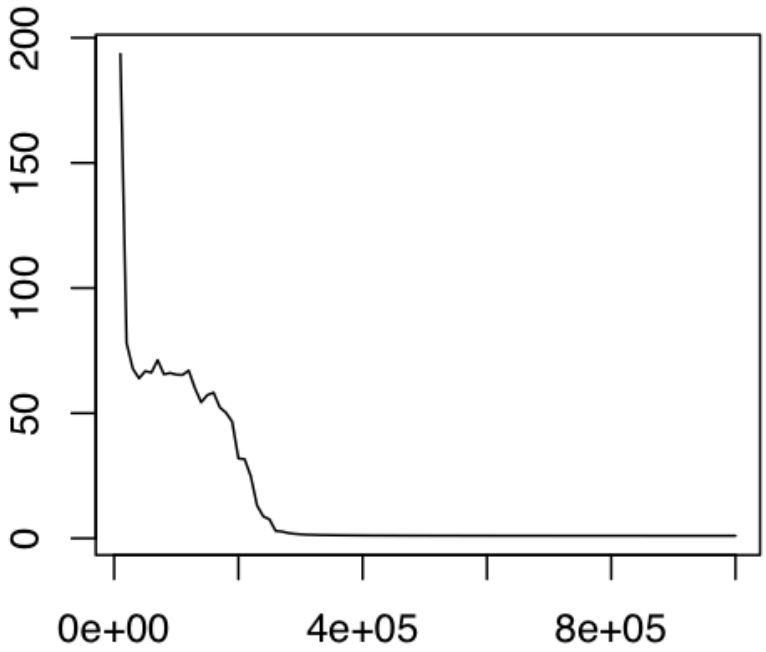
Let's try it ...

High-Dimensional Adaptive Metropolis (cont'd)



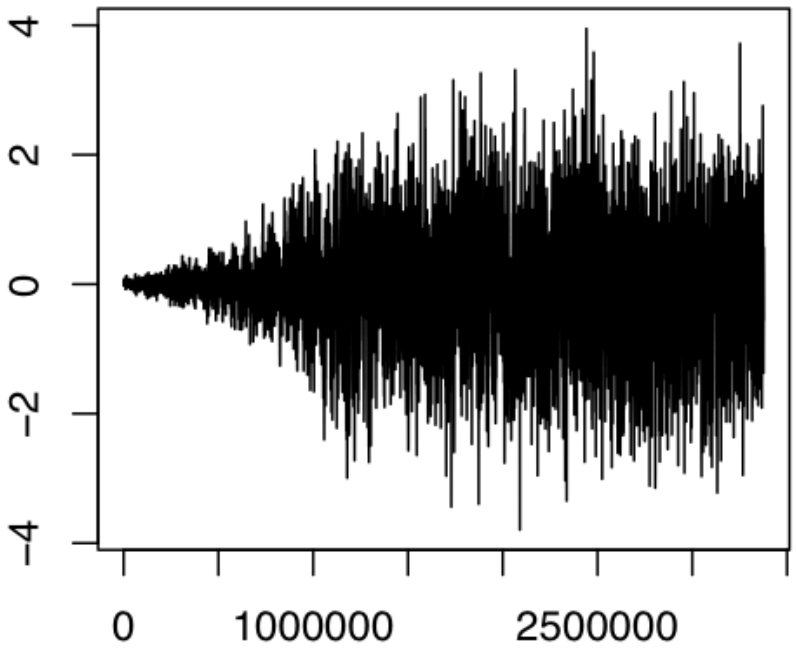
Plot of first coord. Takes about 300,000 iterations, then "finds" good proposal covariance and starts mixing well.

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).

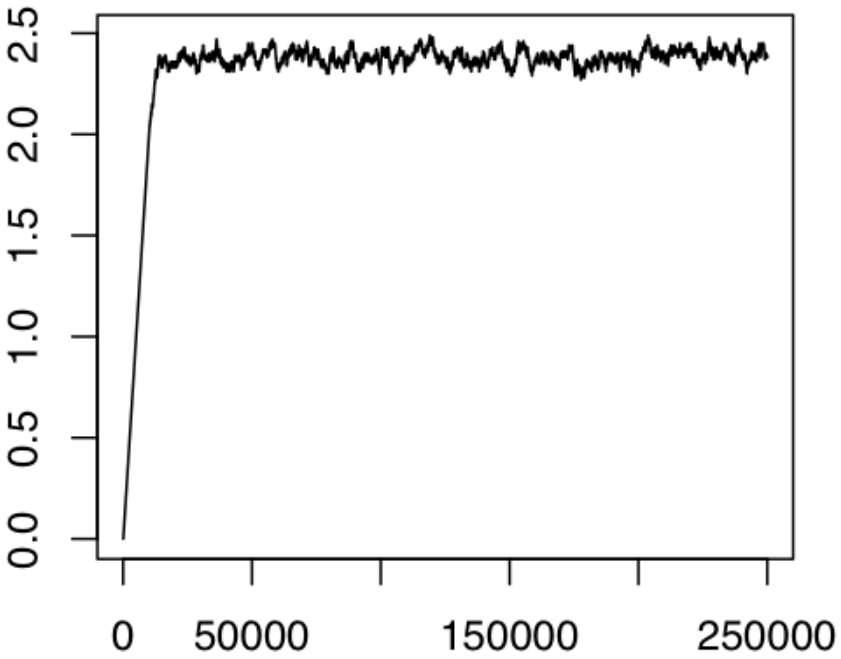
Adapt each ls_i, in batches, to seek 0.44 acceptance rate:

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 $5 \leq J_i \leq 500$, and simulated data $\{Y_{ij}\}$.



Adaption seems to find "good" values for the ls_i values.

Componentwise Metropolis: Comparisons

Variable	J_i	Algorithm	ls_i	ACT	Avr Sq Dist
θ_1	5	Adaptive	2.4	2.59	14.932
θ_1	5	Fixed	0	31.69	0.863
θ_2	50	Adaptive	1.2	2.72	1.508
θ_2	50	Fixed	0	7.33	0.581
θ_3	500	Adaptive	0.1	2.72	0.150
θ_3	500	Fixed	0	2.67	0.147

The Adaptive algorithm mixes much more efficiently than the Fixed algorithm, with smaller integrated autocorrelation time (good) and larger average squared jumping distance (good). And coordinates (e.g. θ_3) that started good, stay good.

Great . . . but is it Ergodic?

So, adaptive MCMC seems to work well in practice.

But will it be ergodic, i.e. converge to $\pi(\cdot)$?

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 $\pi(\cdot)$ stationary).

But adaptive algorithms are more subtle, since the Markov property and stationarity are destroyed by using an adaptive scheme.

e.g. if the adaption of γ is such that P_γ moves slower when x is in a certain subset $\mathcal{X}_0 \subseteq \mathcal{X}$, then the algorithm will tend to spend much more than $\pi(\mathcal{X}_0)$ of the time inside $\mathcal{X}_0$. [APPLET]

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

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. $\lim_{n\to\infty} \|\mathcal{L}(X_n) - \pi(\cdot)\| = 0$, 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, uniformly.

And, Diminishing Adaptation says that we adapt less and less, so that for large enough N,

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

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 $\pi(\cdot)$ 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).

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.

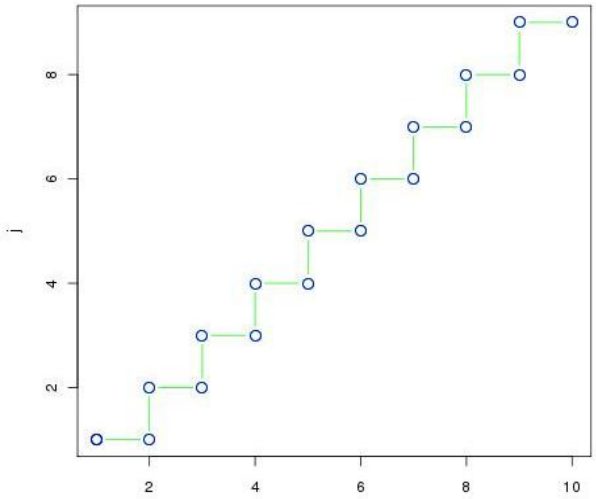
She asked: Do we need to visit every coordinate equally often, or can we gradually "learn" which ones usually don't change and downweight them? Good question – how to proceed?

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: 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??

Counter-example! (K. Latuszyński and R., 2009)
$\mathcal{X} = \{(i,j) \in \mathbf{N} \times \mathbf{N} : i = j \text{ or } i = j+1\}$ ("Stairway to Heaven").



Target $\pi(i,j) = C/j^2$, with adaptive coordinate weights given by:

$$\alpha_{n,1} = \begin{cases} (1/2) + \epsilon_n\,, & X_{n,1} = X_{n,2} \\ (1/2) - \epsilon_n\,, & X_{n,1} = X_{n,2} + 1 \end{cases}$$

and $\alpha_{n,2} = 1 - \alpha_{n,1}$, where $\epsilon_n \searrow 0$ sufficiently slowly.

Then $\alpha_{n,i} \to 1/2 =: \alpha_i^*$, which is indeed ergodic. However, the extra ϵ_n makes $\mathbf{P}(X_n \to \infty) > 0$, i.e. chain is transient.

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

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

For our counter-example, (i) fails because of infinite tails, and (ii) fails because of a slight outward kick.

But if careful about continuity, boundedness, etc., then 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 $\epsilon > 0$,

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

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??

Verifying Containment: "For Everyone"

• We proved general theorems about stability of "adversarial" Markov chains under various conditions (Craiu, Gray, Latuszynski, Madras, Roberts, and R., A.A.P. 2015).

• Then we applied them to adaptive MCMC, to get a list of directly-verifiable conditions which guarantee Containment:

⇒ Never move more than some (big) distance D.

⇒ Outside (big) rectangle K, use fixed kernel (no adapting).

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

⇒ 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 used by applied MCMC users. "Adaptive MCMC for everyone!"

Summary

• MCMC is extremely popular for estimating expectations.

• 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 $\pi(\cdot)$, suffices to have stationarity of each P_γ, plus (a) Diminishing Adaptation (important), and (b) Containment (technical condition, usually satisfied, necessary). Good.

• This demonstrates convergence of adaptive Metropolis, coordinatewise adaptation, adaptive coordinate weights, etc.

• New "adversarial" conditions can easily verify Containment.

• Hopefully can use adaption on many other examples – try it!

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