Adaptive MCMC:
A Java Applet's Perspective

Jeffrey S. Rosenthal

University of Toronto
jeff@math.toronto.edu
http://probability.ca/jeff/

(Based partially upon joint work with G.O. Roberts.)

Warwick University, Coventry, U.K., August 2006

How to improve/optimize MCMC?

Can just choose a "reasonable" algorithm, and hope for the best.

Or: can examine many different trial runs, to attempt to e.g. minimize autocorrelations, or maximize average step sizes, or achieve a theoretically determined optimal acceptance rate ("0.234", etc.: Roberts, Gelman, and Gilks; Roberts and R.; Bédard; Sherlock). Time-consuming, difficult, unreliable.

Or, can adapt, by having the computer modify the chain adaptively, i.e. choose a sequence $\{\Gamma_n\}$ of values for γ "on the fly", to automatically seek better Markov chains.

Adaptive MCMC

The Dream: Given a distribution $\pi(\cdot)$, the computer:

- efficiently and cleverly tries out different MCMC algorithms;
- automatically "learns" the best one(s);
- runs the algorithms for "long enough";
- obtains excellent samples from $\pi(\cdot)$;
- uses these samples to do great estimation;
- reports the results clearly and concisely, with the user unaware of the complicated MCMC and adaption that was used.

The Reality: Easier said than done! But some hope ...

Java Applet Illustrative Example

$\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$, $\pi\{2\} = 0.0001$, $\pi\{1\} = \pi\{3\} = \pi\{4\} = \pi\{5\} = \pi\{6\} \doteq 0.2$. [And $\pi(x) = 0$ for $x \notin \mathcal{X}$.]

Let $\gamma \in \mathbf{N}$, $X_0 \in \mathcal{X}$. Do "random-walk Metropolis" (RWM):

- Given X_n, first propose a state $Y_{n+1} \in \mathbf{Z}$, with $Y_{n+1} \sim \text{Uniform}\{X_n - \gamma, \ldots, X_n - 1, X_n + 1, \ldots, X_n + \gamma\}$.
- Then, with probability $\min[1, \pi(Y_{n+1})/\pi(X_n)]$, accept proposal and set $X_{n+1} = Y_{n+1}$.
- Otherwise, with probability $1 - \min[1, \pi(Y_{n+1})/\pi(X_n)]$, reject proposal and set $X_{n+1} = X_n$.

Works: $\mathcal{L}(X_n) \to \pi(\cdot)$. [APPLET]

Adaption in Java Applet Example

Should $\gamma = 1$, or 50, or . . .??

Want acceptance rate to be not too small, not too big ("Goldilocks Principle"). Start with γ set to $\Gamma_0 = 2$ (say). Then:

Each time proposal is accepted, set $\Gamma_{n+1} = \Gamma_n + 1$ (so γ increases, and acceptance rate decreases).

Each time proposal is rejected, set $\Gamma_{n+1} = \max(\Gamma_n - 1, 1)$ (so γ decreases, and acceptance rate increases).

Logical, natural adaptive scheme. Computer performs "search" $\{\Gamma_n\}$ for a good γ, on the fly. But does it work? [APPLET]

NO IT DOESN'T!!

The chain eventually gets stuck with $X_n = \Gamma_n = 1$ for long stretches of time. [Asymmetric: entering $\{X_n = \Gamma_n = 1\}$ much easier than leaving it.]

Chain doesn't converge to $\pi(\cdot)$ at all.

The adaption has RUINED the algorithm. Disaster!!

[Could convolve with $N(0, 10^{-6})$, to make it continuous . . .]

When Does Adaptation Preserve Stationarity?

- Can adapt at regeneration times $T_1, T_2, \ldots$ with $X_{T_i} \sim \nu(\cdot)$ (Gilks, Roberts, and Sahu, 1998; Brockwell and Kadane, 2002).
- "Adaptive Metropolis" algorithm [Haario, Saksman, and Tamminen, 2001]: Use proposal distribution $MVN(\mathbf{x}, (2.38)^2\Sigma_n \,/\, d)$.
- More general/flexible adaptive schemes [Atchadé and R., Andrieu and Moulines, Andrieu and Robert, Andrieu and Atchadé]. Require complicated conditions involving drift functions and convergence rates. Also require that $\{\Gamma_n\} \to \gamma_*$, i.e. no infinite adaptation.

Simple Convergence Theorem: Uniform Case

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

- [Stationarity] $\pi(\cdot)$ is stationary for each P_γ. [Of course.]
- [Diminishing Adaptation] $\sup_{x\in\mathcal{X}} \|P_{\Gamma_{n+1}}(x,\cdot) - P_{\Gamma_n}(x,\cdot)\| \to 0$ as $n \to \infty$ (at least, in probability). [At any rate ... adaptations can be small, or done with prob $p(n) \to 0$. User controlled.]
- [Uniform Convergence Rate] For all $\epsilon > 0$, there is $N = N(\epsilon) \in \mathbf{N}$ such that $\|P_\gamma^N(x,\cdot) - \pi(\cdot)\| \leq \epsilon$ for all $x \in \mathcal{X}$ and $\gamma \in \mathcal{Y}$. [Too strong! But still useful ...]

Corollaries of Simple Convergence Theorem

COR: Have $\lim_{n\to\infty} \|\mathcal{L}(X_n) - \pi(\cdot)\| = 0$ if have Stationarity (of course), and Diminishing Adaptation, and either
(a) $\mathcal{X}$ and $\mathcal{Y}$ are both finite, or
(b) $\mathcal{X}$ and $\mathcal{Y}$ are both compact, and transition densities continuous.

COR: Validity of "Adaptive Metropolis" algorithm.
(So, the theorem provides easier proof of previously-known result.)

COR: Can replace Uniform Convergence Rate condition with: $\forall \epsilon > 0$, $\{T_\epsilon(X_n, \Gamma_n)\}_{n=0}^{\infty}$ is Bounded in Probability, where

$$T_\epsilon(x, \gamma) \;=\; \inf\{n \geq 1 : \|P_\gamma^n(x, \cdot) - \pi(\cdot)\| \leq \epsilon\}\,.$$

(This also follows from various drift conditions, etc.)

What about Java Applet Example?

Stationarity: Yes (of course).

Diminishing Adaptation: No. But yes if at time n, adapt only with probability $p(n) \to 0$, otherwise leave γ unchanged. [e.g. $p(n) = 1/n$, so $\Sigma_n\, p(n) = \infty$, i.e. infinite adaptation.]

Convergence Rate: Not quite Uniform (since γ unbounded), but still satisfies "Bounded in Probability" condition.

COR: Adaptation in Java Applet example is valid if Diminishing Adaptation modification is made. Phew!

Example: $MVN(0,\, I_{10})$

$\mathcal{X} = \mathbf{R}^{10}$; $\quad \pi(\cdot) = MVN(0,\, I_{10})$.

$P_{a,b}$ is Metropolis-Hastings algorithm with proposal

$$Q_{a,b}(x, \cdot) \;=\; \begin{cases} MVN(x,\, e^{2a})\,, & \|x\|^2 \le 10 \\ MVN(x,\, e^{2b})\,, & \|x\|^2 > 10\,. \end{cases}$$

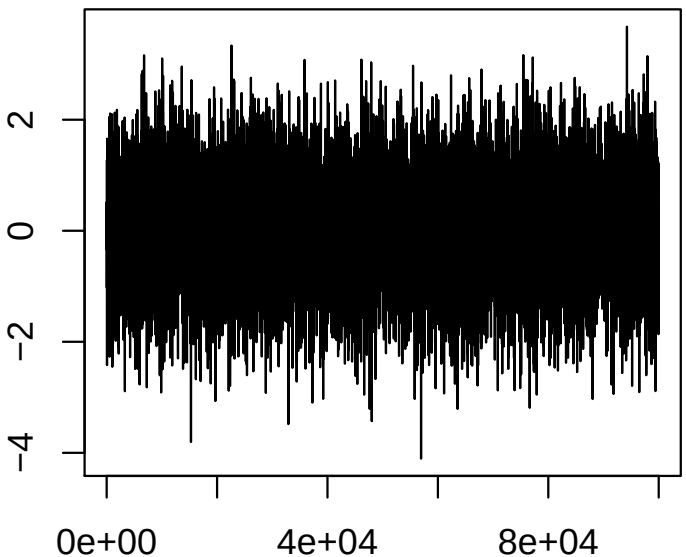
Begin with $a = b = 0$. After n^{th} "batch" of 100 (say) iterations, add or subtract $\delta(n)$ to a, to move acceptance rate on $\{\|x\|^2 \le 10\}$ closer to 0.234.

Similarly for b and $\{\|x\|^2 > 10\}$.

Diminishing Adaptation: $\delta(n) \to 0$, e.g. $\delta(n) = \min(0.01,\, n^{-1/2})$.

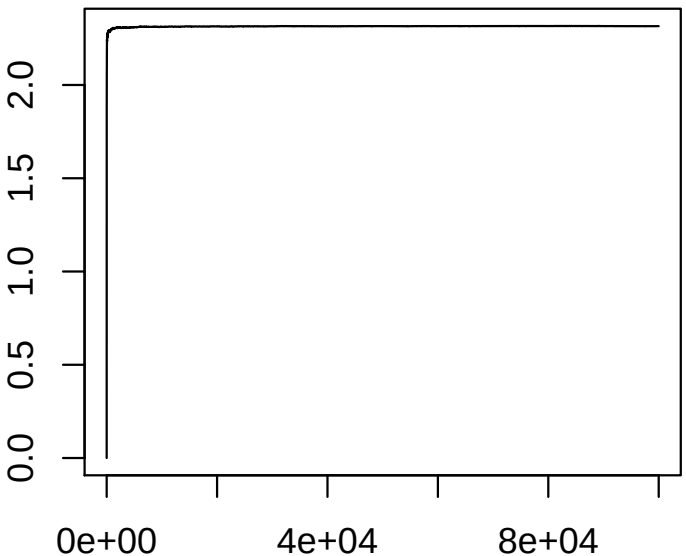
Does it work?

$MVN(0, I_{10})$: Mixing of First Coordinate



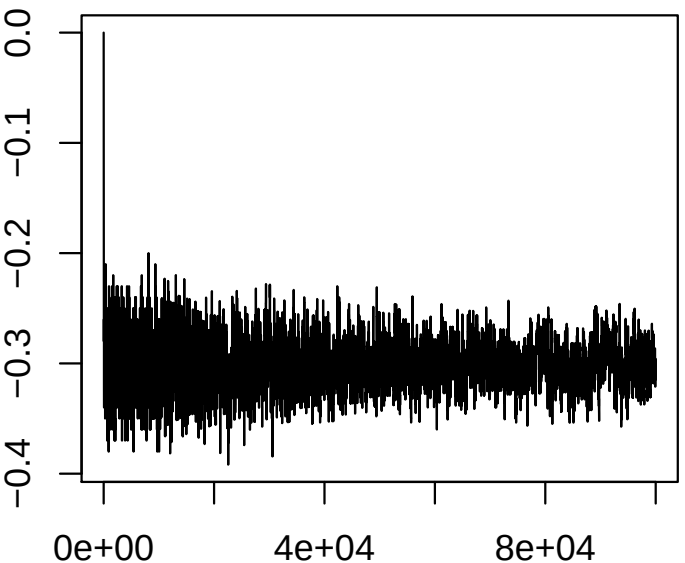
(Shows excellent mixing.)

$MVN(0,\, I_{10})$: Convergence of $\mathbf{E}[\log(1 + \|x\|^2)]$



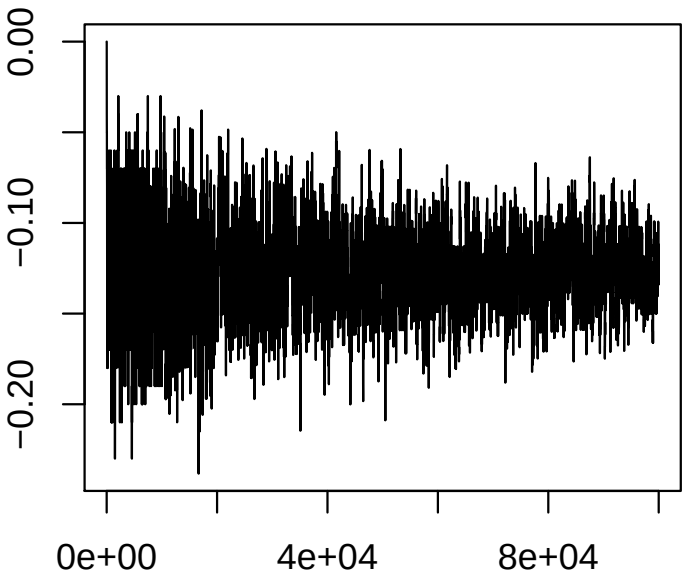
(Rapid convergence to the true value of 2.3152.)

$MVN(0, I_{10})$: Behaviour of parameter "a"



(Quick approach to values near -0.3, but with some oscillation.)

$MVN(0, I_{10})$: Behaviour of parameter "b"



(Quick approach to values near -0.13; again some oscillation.)

$MVN(0, I_{10})$: Comparison to fixed a, b

a, b	ACT	Avr Sq Dist
adaptive (as above)	15.54	0.125
-0.3, -0.13	15.07	0.126
0.0, 0.0	17.04	0.110
-0.3, -0.3	16.01	0.122
-0.13, -0.13	15.76	0.121
-0.284, -0.284	15.91	0.123

Adaptive algorithm (top line) quite competitive with corresponding fixed-parameter choice (second line), which is better than any other fixed a and b (including bottom line: optimal homogeneous with $e^a = e^b = 2.38 / \sqrt{10}$).

Example: Variance Components Model

$$
\begin{array}{ccccc}
 & & \mu & & \\
 & \swarrow & \downarrow & \searrow & \\
\theta_1 & \ldots & & \ldots & \theta_K \\
\downarrow & \ldots & & \ldots & \downarrow \\
Y_1 & \ldots & & \ldots & Y_K
\end{array}
\qquad\qquad
\begin{array}{l}
\theta_i \sim N(\mu, A) \\
\\
Y_i \sim N(\theta_i, V)
\end{array}
$$

Priors: $\mu \sim N(0,1)$; $A \sim IG(-1,2)$; V fixed (emp. Bayes).

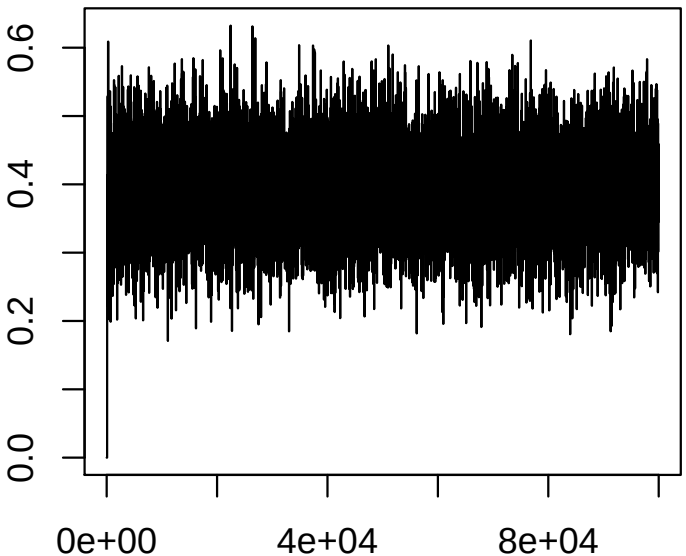
$\pi(\cdot) =$ resulting posterior distribution for $(A, \mu, \theta_1, \ldots, \theta_K)$.

$K = 18$, so $\mathcal{X} = [0,\infty) \times \mathbf{R}^{19} \subseteq \mathbf{R}^{20}$.

$Y_1, \ldots, Y_{18}$: baseball data of Morris (1983, Table 1)

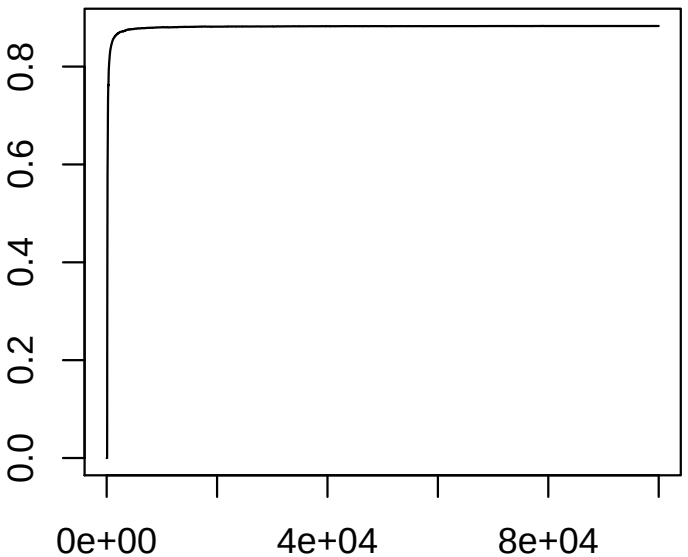
Add or subtract $\delta(n)$ to a and to b, to make acceptance rates in $\{\Sigma_i(\theta_i)^2 \leq 0.15\}$ and in $\{\Sigma_i(\theta_i)^2 > 0.15\}$ closer to 0.234.

Variance Components: θ_1



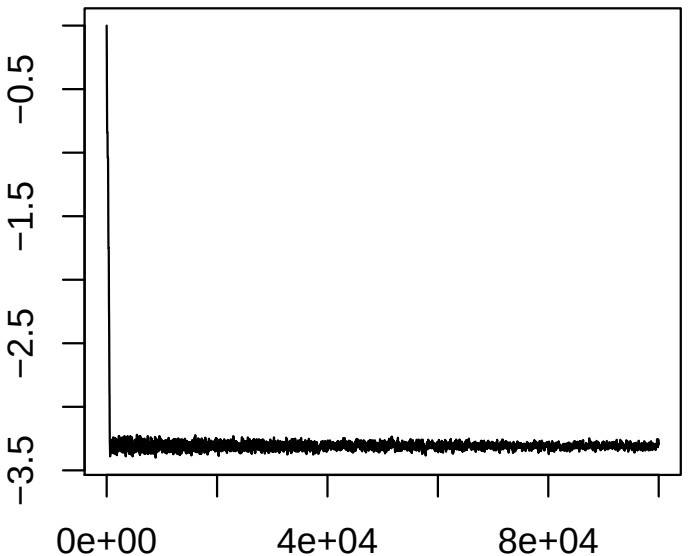
(Excellent mixing; mean very near true mean of 0.394.)

Variance Components: $\mathbf{E}[\log(1+\Sigma_i(\theta_i)^2)]$



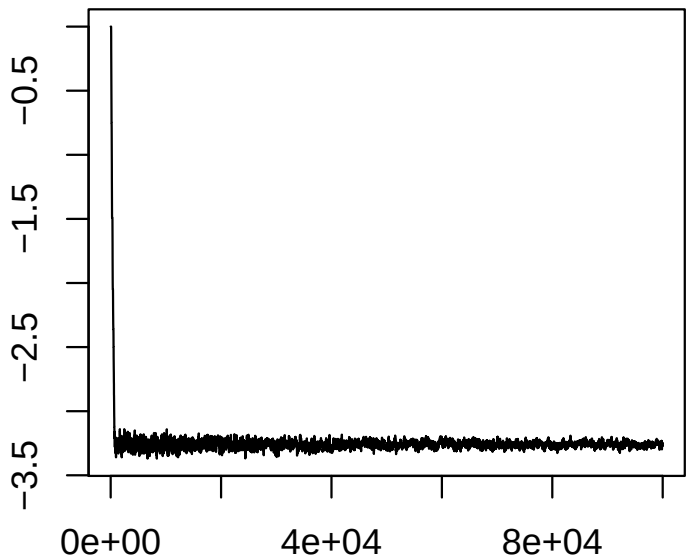
(Excellent convergence to true value of 0.885.)

Variance Components: Parameter "a"



(Rapid convergence to near -3.3.)

Variance Components: Parameter "b"



(Rapid convergence to near -3.2.)

Variance Components: Comparisons

a, b	ACT	Avr Sq Dist $\times 10^4$
adaptive (as above)	31.60	2.76
-3.3, -3.2	25.75	2.79
-2.3, -2.3	50.67	0.192
-4.3, -4.3	38.92	1.17
-3.3, -4.3	36.91	1.15
-4.3, -3.3	38.04	2.41
-0.63, -0.63	53.91	0.003

Adaptive algorithm (top line) competitive with corresponding fixed-parameter choice (second line); better than other choices (including bottom line: optimal homogeneous with $e^a = e^b = 2.38 \,/\, \sqrt{20}$).

Summary

Adaptive MCMC seems promising ("computer learning") – good.

But must be done carefully, or it will destroy stationarity – bad.

To converge to $\pi(\cdot)$, suffices to have (a) Stationarity, (b) Diminishing Adaptation, and (c) convergence times Bounded in Probability (guaranteed by e.g. compactness or drift conditions or ...).

Our adaptive schemes appear to perform better than arbitrarily-chosen RWM, at least as good as wisely-chosen RWM, and nearly as good as an ideally-chosen variable-σ^2 schemes. Promising!

Questions for Future

- Are Diminishing Adaptation and Bounded in Probability conditions really necessary? (Yes in Java Applet example, but . . .)
- Infinite Oscillation: Good or Bad? (Previous adaptive MCMC results assume $\{\Gamma_n\} \to \gamma_*$ [Applet: $\Sigma_n\, p(n) < \infty$]. Hopefully converge to "best" γ_*. But might not know best γ_*, or might converge to "wrong" value.)
- Generalisation: "Regional Adaptive Metropolis Algorithm" (RAMA): partition $\mathcal{X} = \mathcal{X}_1 \dot{\cup} \ldots \dot{\cup} \mathcal{X}_r$, and proposal $Q(x, \cdot) = N(x,\, e^{2a_i})$ for $x \in \mathcal{X}_i$, with each a_i adjusted after each batch.
- Better adaptive schemes? Better choice of regions? Other functional forms [e.g. $\sigma_x = e^a\,(\log(1 + |x|))^b$]? How to balance sophistication with simplicity (e.g. 2^d regions . . .)?