An introduction to adaptive MCMC

Gareth Roberts

MIRAW Day on Monte Carlo methods
March 2011

Mainly joint work with Jeff Rosenthal.

`http://www2.warwick.ac.uk/fac/sci/statistics/crism/`

- Conferences and workshops
- Academic visitor programme.
- Workshops coming up:
  - March 28 - April 1, INfER, Statistics for infectious diseases
  - April 5 - 7, WOGAS III, geometry and statistics
  - April 13, Prior elicitation

Chosing parameters in MCMC

Target density π on statespace $\mathcal{X}(= \mathbf{R}^d)$

Most MCMC algorithms offer the user **choices** upon which convergence properties depend crucially.

Random walk Metropolis (**RWM**) might propose moves fromn x to $Y \sim N(x, \sigma^2 I_d)$, accepting it with probability

$$\min \left\{ 1, \frac{\pi(Y)}{\pi(x)} \right\}$$

otherwise remaining at x.

This is both a **problem** and an **opportunity**!

Adaptive MCMC

The Goal: Given a model (i.e., $\mathcal{X}$ and $\pi(\cdot)$), the computer:

- efficiently and cleverly tries out different MCMC algorithms;
- automatically "learns" the good ones;
- runs the algorithm for "long enough";
- obtains excellent estimates together with error bounds;
- reports the results clearly and concisely, while user unaware of the complicated MCMC and adaption that was used.

Easier said than done ...

A surprising example

Let $\mathcal{Y} = \{-1, +1\}$, $\pi(x) \propto (1 + x^2)^{-1}$, $x > 0$. Then if $X \sim \pi$ then $X^{-1} \sim \pi$.

If $\Gamma = 1$ the algorithm carries out a RWM step with proposal $N(0, 1)$. If $\Gamma = -1$ the algorithm carries out a RWM step for X^{-1}, ie given a standard normal Z, we propose to move to $(x^{-1} + Z)^{-1}$.

Now consider the adaptive strategy:

- If $\Gamma = 1$, change to $\Gamma = -1$ if and only if $X_n < n^{-1}$.
- If $\Gamma = -1$, change to $\Gamma = 1$ if and only if $X_n > n$.

stationarity and **diminishing adaptation** certainly hold for this adaptive schemes in this example.

How does it do?

For any set A bounded away from both 0 and ∞, $\mathbf{P}(X_n \in A) \to 0$ as $n \to \infty$!

For any set containing an open neighbourhood of 0, $\mathbf{P}(X_n \in A) \to 1/2$ as $n \to \infty$

The algorithm has a *null recurrent* character, always returning to any set of positive Lebesgue measure, but doing so with a probabiity that recedes to 0.

There are only **TWO** strategies.

They **BOTH** have identical convergence properties.

When Does Adaptation Preserve Stationarity?

- **Regeneration times** [Gilks, Roberts, and Sahu, 1998; Brockwell and Kadane, 2002]. Some success, but difficult to implement.
- **"Adaptive Metropolis" (AM) algorithm** [Haario, Saksman, and Tamminen, 2001]: proposal distribution $N(\mathbf{x}, (2.38)^2 \Sigma_n / d)$, where Σ_n = (bounded) empirical estimate of covariance of $\pi(\cdot)$ ...
- Generalisations of AM [Atchadé and R., Andrieu and Moulines, Andrieu and Robert, Andrieu and Atchadé, Kohn]. Require technical conditions.

We need **simple** conditions guaranteeing $\|\mathcal{L}(X_n) - \pi(\cdot)\| \to 0$.

Simple Adaptive MCMC Convergence Theorem

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

- **Stationarity** $\pi(\cdot)$ is stationary for each P_γ. [Of course.]
- **Diminishing Adaptation**
  $\lim_{n \to \infty} \sup_{x \in \mathcal{X}} \|P_{\Gamma_{n+1}}(x, \cdot) - P_{\Gamma_n}(x, \cdot)\| = 0$ (at least, in probability). [**Either** the adaptations need to be small with high probability **or** adaptation takes place with probability $p(n) \to 0$.]
- **Containment** Times to stationary from X_n, with transitions P_{Γ_n}, remain bounded in probability as $n \to \infty$.

Proof: **Couple** adaptive chain to another chain that eventually **stops** adapting . . .

Ergodicity

Under stationarity, adaptation and containment also get

$$\frac{\lim_{t\to\infty} \sum_{n=1}^{t} f(X_n)}{t} = \pi(f) \text{ in } \textbf{probability}$$

for any **bounded** function f.

However convergence for all $\mathbf{L^1}$ functions does **not** follow (counterexample by Chao, Toronto).

Satisfying containment

Diminishing adaptation and stationarity are straighforward to ensure. However containment is more complicated.
Here are some conditions which imply containment.

- **Simultaneous Uniform Ergodicity** 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 $\gamma \in \mathcal{Y}$;
- **Simultaneous geometrically ergodicity** (Roberts, Rosenthal, and Schwartz, 1998, JAP). There is $C \in \mathcal{Y}$, $V : \mathcal{X} \to [1, \infty)$, $\delta > 0$, $\lambda < 1$, and $b < \infty$, such that $\sup_C V = v < \infty$, and
  1. for each $\gamma \in \mathcal{Y}$, there exists a probability measure $\nu_\gamma(\cdot)$ on C with $P_\gamma(x, \cdot) \geq \delta \, \nu_\gamma(\cdot)$ for all $x \in C$; and
  2. $(P_\gamma)V \leq \lambda \, V + b \, \mathbf{1}_C$.
- A polynomnial version of the above..

Common drift function conditions

Simultaneous geometric ergodicity and simultaneous polynomial ergodicity are examples of **common drift function conditions**.

Such conditions say that all Markov kernels can be considered to be pushing towards the same small set as measured by the *common drift condition*.

When different chains have *different ways of moving towards* π, disaster can occur!

A surprising example (cont)

Let $\mathcal{Y} = \{-1, +1\}$, $\pi(x) \propto (1 + x^2)^{-1}$, $x > 0$. Then if $X \sim \pi$ then $X^{-1} \sim \pi$.

If $\Gamma = 1$ the algorithm carries out a RWM step with proposal $N(0, 1)$. If $\Gamma = -1$ the algorithm carries out a RWM step for X^{-1}, ie given a standard normal Z, we propose to move to $(x^{-1} + Z)^{-1}$.

Now consider the adaptive strategy:

- If $\Gamma = 1$, change to $\Gamma = -1$ if and only if $X_n < n^{-1}$.
- If $\Gamma = -1$, change to $\Gamma = 1$ if and only if $X_n > n$.

Thus **stationarity** and **diminishing adaptation** certainly hold for this adaptive schemes.

What goes wrong?

Containment must fail. But how?

For $\Gamma = 1$, any set bounded away from ∞ is small. [$P(x, \cdot) \geq \epsilon\nu(\cdot)$ for all $x \in$ the small set.]

For $\Gamma = -1$, any set bounded away from 0 is small.

A natural drift function for $\Gamma = 1$ diverges at $x \to \infty$, and is bounded on bounded regions. A natural drift function for $\Gamma = -1$ is unbounded as $x \to 0$, and is bounded on regions bounded away from zero.

Cannot do both with a single drift function!

Common drift functions for RWM

If $\{P_\gamma\}$ are a class of full-dimensional RWMs, a natural drift function is often $\pi(\mathbf{x})^{-1/2}$. For instance if the contours of π are *sufficiently regular* in the tails, and P_γ represents RWM with proposal variance matrix γ such that there exists a constant ϵ with

$$\epsilon I_d \leq \gamma \leq \epsilon^{-1} I_d$$

(with the inequalities understood in a positive-definite sense). This is useful for the **adaptive Metropolis (AM) example later**. [This is based on recent work by R + Rosenthal extending R+Tweedie, 1996, Biometrika.]

Common drift functions for single component Metropolis updates

Consider **single component** Metropolis updates.
Easiest to prove for *random scan* Metropolis which just chooses a component to update at random, and then updates according to a 1-dimensional Metropolis.

Again we can use $V(\mathbf{x}) = \pi(\mathbf{x})^{-1/2}$.

[This is based on recent work by Latuszynski, R + Rosenthal extending R+Rosenthal, 1998, AnnAP.]

Scaling proposals

π is a d-dimensional probability density function with respect to d-dimensional Lebesgue measure. Consider

- **RWM** Propose new value $\mathbf{Y} \sim N(\mathbf{x}, \gamma)$ [γ is a matrix here.]
- **MwG** Choose a component at random, i say, and update $X_i|\mathbf{X}_{-i}$ according to a one-dimensional Metropolis update with proposal $N(x_i, \gamma_i)$. [γ is a vector here!]
- **Lang** Langevin update: propose new value from $N(\mathbf{x} + \gamma \nabla \log \pi(\mathbf{x})/2, \gamma)$. [$\gamma$ is a matrix here!]

What scaling theory tells us, RWM case ...

R, Rosenthal, Sherlock, Pete Neal, Bedard, Gelman, Gilks...
For **RWM**, we set $\gamma = \delta^2 \times V$ for a scalar δ.

- For continuous densities, and for fixed V, the scalar optimisation is often achieved by finding the algorithm which achieves an acceptance rate somewhere between 0.15 and 0.4. In *many* limiting cases (as $d \to \infty$) the optimal value is 0.234.
- "0.234" breaks down only when V is an extremely poor approximation to the target distribution covariance Σ.
- For Gaussian targets, when $V = \Sigma$, the optimal value of δ is

$$\delta = 2.38/d^{1/2} \ .$$

- In the Gaussian case we can exactly quantify the cost of having V different to Σ:

$$R = \frac{(\sum \lambda_i^2/d)^{1/2}}{\sum \lambda_i/d} = \frac{L2}{L1}$$

  where $\{\lambda_i\}$ denote the eigenvalues of $V^{1/2}\Sigma^{-1/2}$.
- For densities with discontinuities, the problem is far more complex. For certain problems, the optimal limiting acceptance probability is 0.13.
- Optimal scaling is **robust to transience**.

What scaling theory tells us, MwG case ...

R, Pete Neal

- Sensibly scaled MwG is (at) least as good as RWM in virtually all cases.
- Moreover it is often more natural (conditional independence structure) and/or easier to implement.
- In the Gaussian case, it is right to tune each component to have acceptance rate 0.44
- The optimal scaling is $\gamma_i = \xi_i \times 2.4^2$ where ξ_i is the target **conditional** variance.

What scaling theory tells us, Lang case ...

R + Rosenthal

Again set $\gamma = \delta^2 \times V$.

- Generally **Lang** much more efficient than **RWM** or **MwG** (convergence time of optimal algorithm is $O(d^{1/3})$ rather than $O(d)$ for the other two).
- Somewhat less robust to light tails, zeros of the target, discontinuous densities, etc.., not robust to transience
- Single component at a time Langevin methods are very sub-optimal in general.
- "Optimal acceptance rate" is around 0.574.
- In Gaussian case, can quantify the penalty for V being different from Σ

$$R = \frac{(\sum \lambda_i^6/d)^{1/6}}{\sum \lambda_i/d} = \frac{L6}{L1}$$

Adaptive Metropolis (AM) Example

Dim $d = 100$, $\pi(\cdot) =$ randomly-generated, eratic MV normal.
So covariance is 100×100 matrix (5,050 different entries).
Do Metropolis, with proposal distribution given by:

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

(for $n > 2d$, say).
Here $\Sigma_n =$ current empirical estimate of covariance of $\pi(\cdot)$.
Also $\beta = 0.05 > 0$ to satisfy Containment condition.

Adaptive Metropolis Example (cont'd)

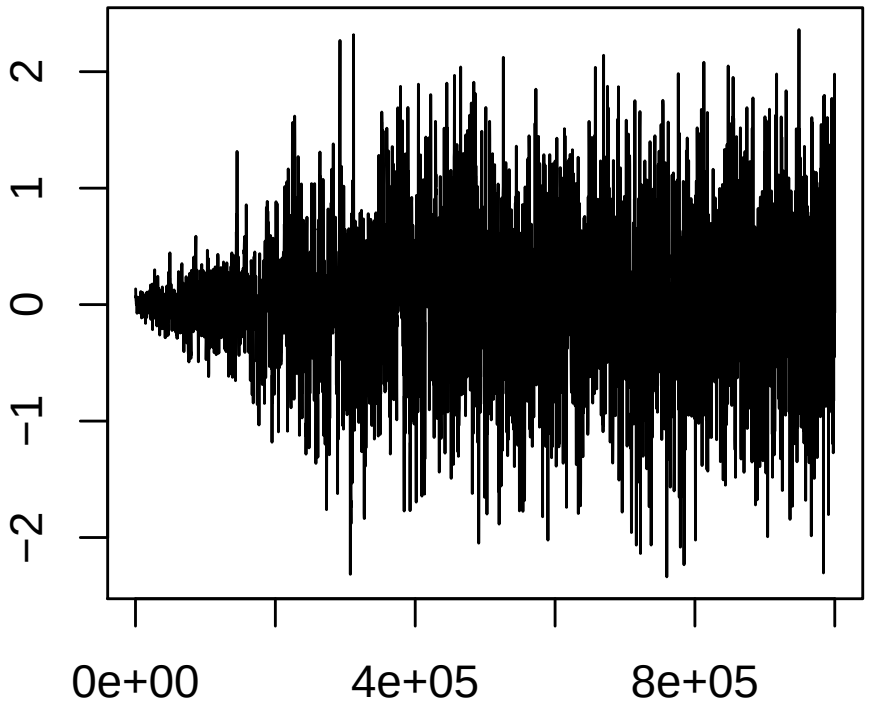


Figure: Plot of first coordinate Takes about 300,000 iterations, then "finds" good proposal covariance and starts mixing well.

Adaptive Metropolis Example (cont'd)

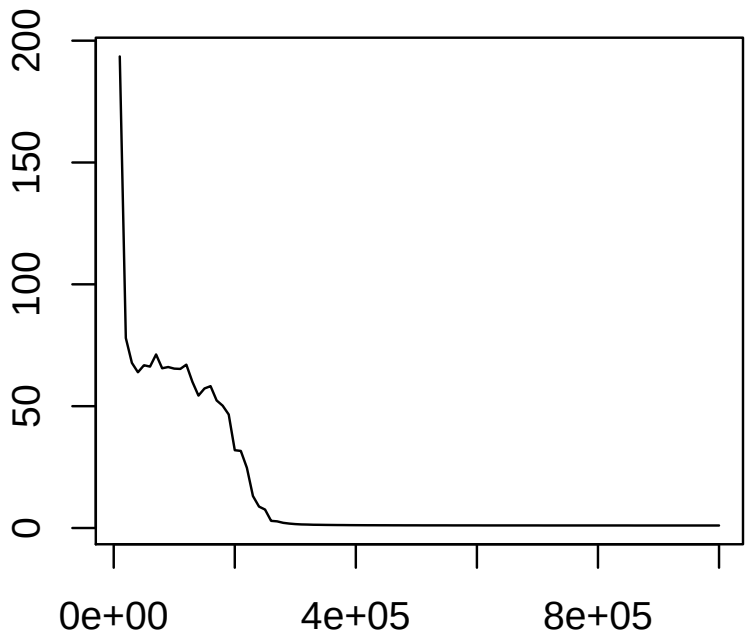


Figure: Plot of sub-optimality factor $R_n \equiv d \left(\sum_{i=1}^{d} \lambda_{in}^{-2} / (\sum_{i=1}^{d} \lambda_{in}^{-1})^2 \right)$, where $\{\lambda_{in}\}$ eigenvals of $\Sigma_n^{1/2} \Sigma^{-1/2}$. Starts large, converges to 1.

Adaptive Metropolis-within-Gibbs Strategies

Possible strategies for each component separately:

1. adapt proposal variance in particular component to match the empirical variance (SCAM);
2. adapt component acceptance probability to be around 0.44 (AMwG);
3. try to estimate some kind of average conditional variance empirically and fit to proposal variance.

Our empirical evidence suggests that AMwG uusally beats SCAM, **but** all strategies can be tested by problems for which heterogenous variances are required.

In the Gaussian case, can show that (2) and (3) converge to "optimal" MwG.

Adaptive Metropolis-within-Gibbs Example

Propose move for each coordinate **separately**.
Propose increment $N(0, e^{2\,ls_i})$ for i^{th} coord.

Start with $ls_i \equiv 0$ (say).
Adapt each ls_i, in batches, to seek 0.44 acceptance ratio (approximately optimal for one-dim proposals).

Test on Variance Components Model, with $K = 500$ location parameters (so dim $= 503$), and data $Y_{ij} \sim N(\mu_i, 10^2)$ for $1 \leq i \leq K$ and $1 \leq j \leq J_i$, where the J_i are chosen between 5 and 500.

Metropolis-within-Gibbs (cont'd)

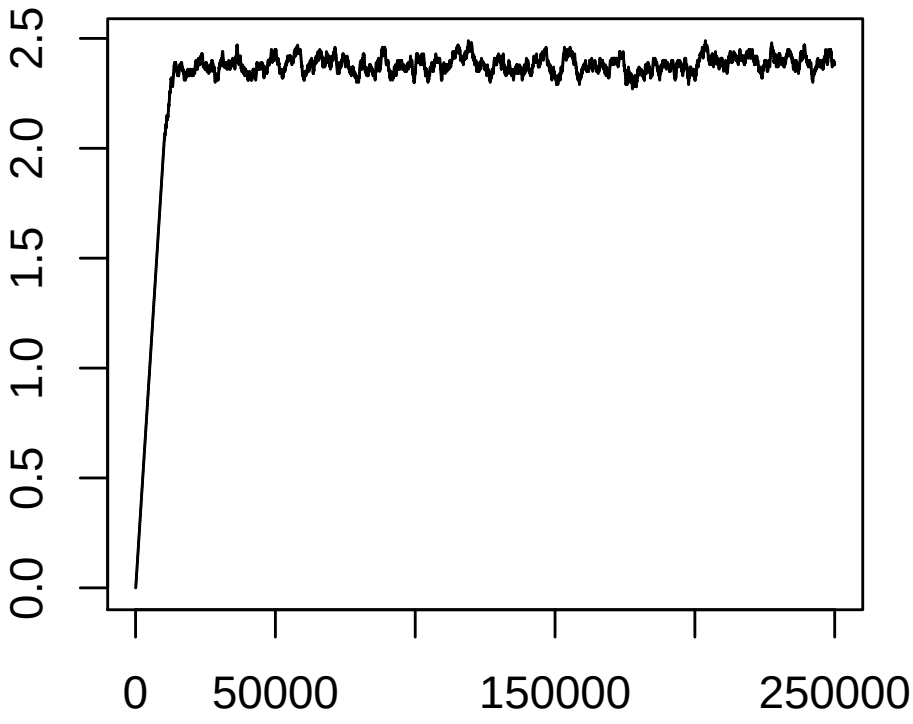


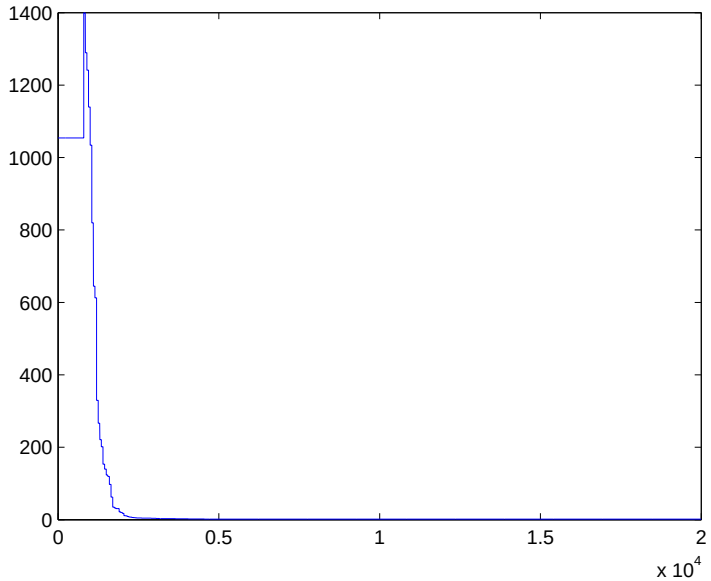
Figure: Adaptation finds "good" values for the ls_i values.

Metropolis-within-Gibbs: 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

Adaptive much better than Fixed, even in dimension 503.

Adaptive Langevin



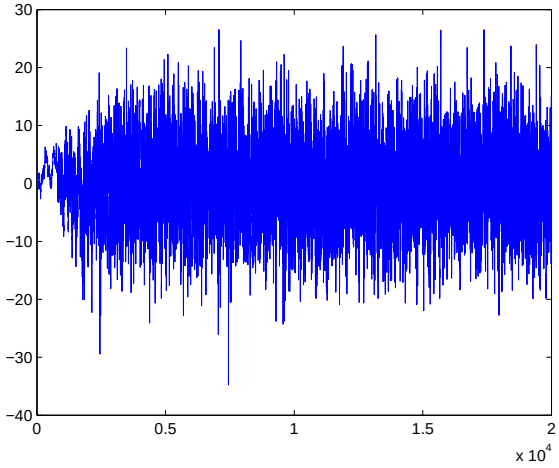


Figure: Adaptive Langevin traceplot, $d = 20$

Heterogenous scaling

Instead of random-walk Metropolis with **fixed** proposal increment distribution (e.g. $N(0, \sigma^2)$), allow $\sigma^2 = \sigma_x^2$ to depend on x, e.g.

$$\sigma_x^2 = e^a \left(1 + |x|\right)^b$$

(with suitably modified M-H acceptance probabilities).

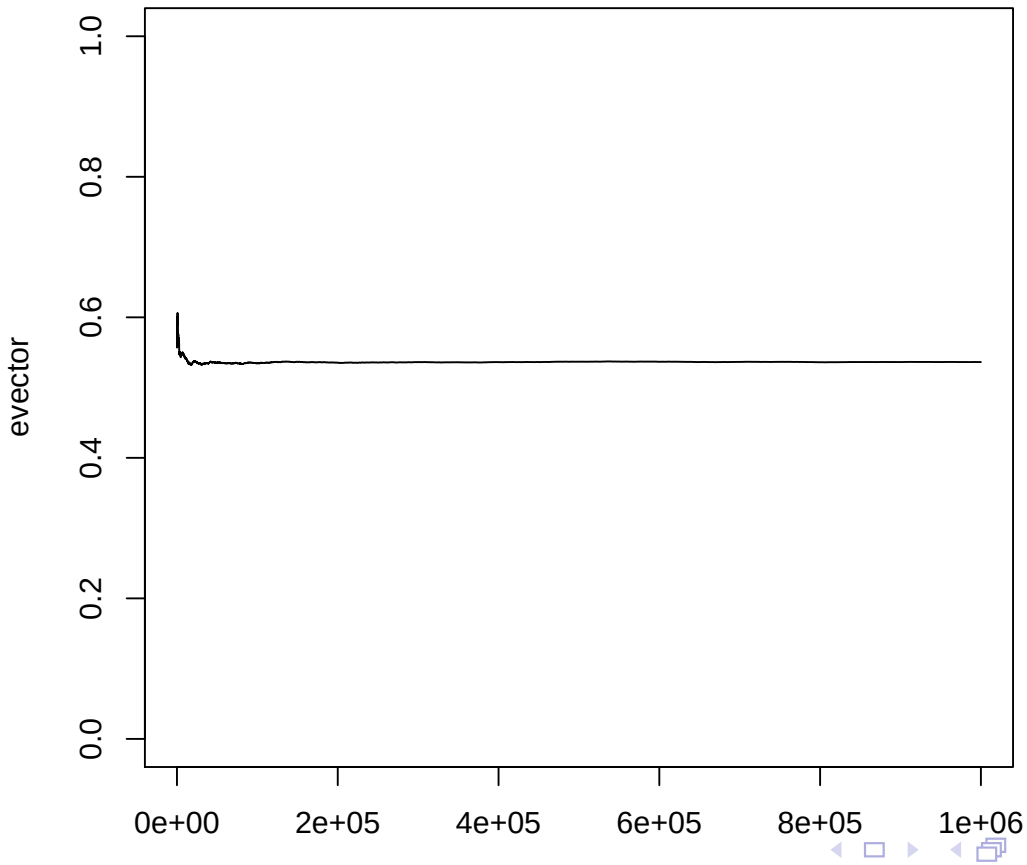
Adjust a and b by $\pm 1/i$, once every 100 (say) iterations, by:

- Decrease a if too few acceptances; increase it if too many.
- Decrease b if fewer acceptances when $|x|$ large; increase it if fewer when $|x|$ small.

Heterogenous scaling: how does it perform?

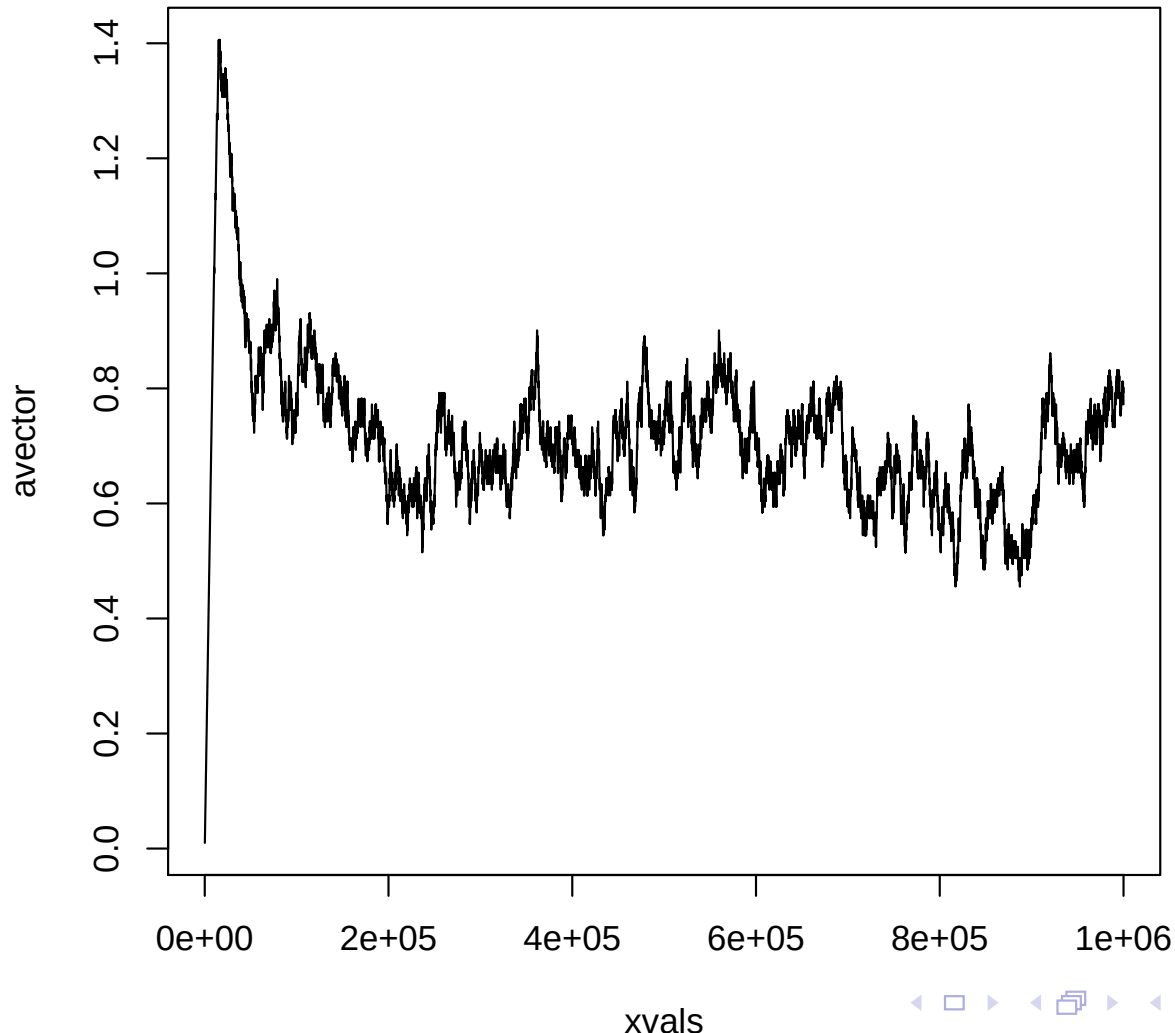
Example: $\pi(\cdot) = N(0, 1)$. We are interested in estimating $\pi\big(\log(1 + |x|)\big)$

Estimate converges quickly to true value (0.534822):



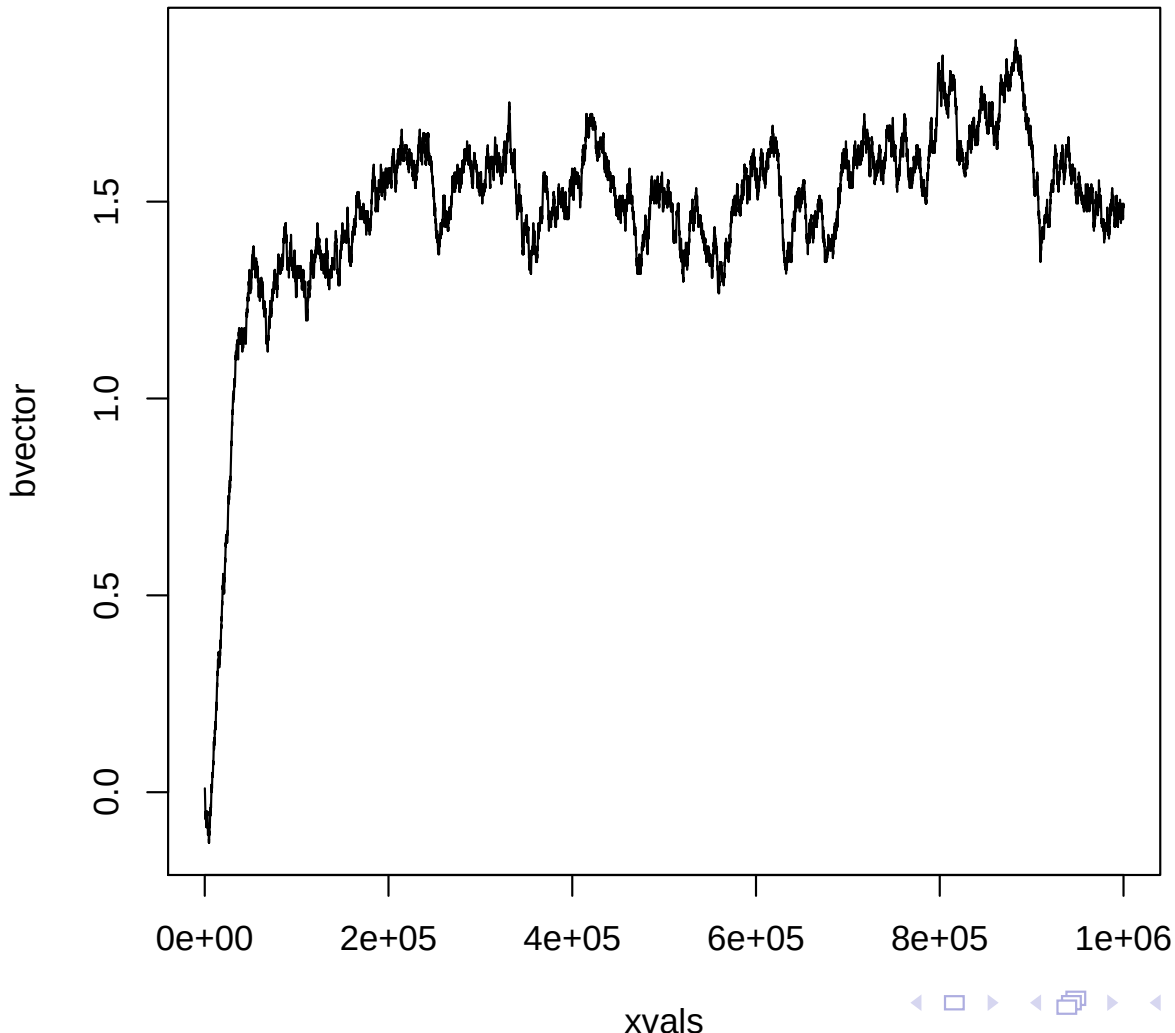
One-Dimensional Adaption: Parameter 'a'

a moves to near 0.7, but keeps oscillating:



One-Dimensional Adaption: Parameter 'b'

b moves to near 1.5, but keeps oscillating:



One-Dimensional Adaption: Comparisons

Algorithm	Acceptance Rate	ACT	Avr Sq Dist
$\sigma^2 = \exp(-5)$	0.973	55.69	0.006
$\sigma^2 = \exp(-1)$	0.813	8.95	0.234
$\sigma^2 = 1$	0.704	4.67	0.450
$\sigma^2 = \exp(5)$	0.237	7.22	0.305
$\sigma^2 = (2.38)^2$	0.445	2.68	0.748
$\sigma_x^2 = e^{0.7}\,(1 + \lvert x \rvert)^{1.5}$	0.458	2.55	0.779
Adaptive (as above)	0.457	2.61	0.774

Conclusion: heterogenous adaptation is much better than arbitrarily-chosen RWM, appears slightly better than wisely-chosen RWM, and nearly as good as an ideally-chosen variable-σ_x^2 scheme.

The story is much more clear-cut where heavy tailed targets.

Some conclusions

- Good progress has been made towards finding simple usable conditions for adaptation, **at least for standard algorithms**. Further progress is subject of joint work with Krys Latuszynski and Jeff Rosenthal.
- Algorithm families with different small sets and drift conditions are dangerous!
- In practice, Adaptive MCMC is **very easy**. Current implentation in some forms of BUGS.
- More complex adaptive schemes often require individual convergence results (eg for adaptive Gibbs samplers and its variants in work with Krys and Jeff, and work by Atchade and Liu on the **Wang Landau** algorithm.