Some recent developments in scaling of Metropolis-Hastings algorithms

Gareth Roberts
Lancaster University

Joint work with Andrew Gelman, Wally Gilks, Jeff Rosenthal, Ole Christensen, Pete Neal, John Yuen

CRiSM workshop, Warwick, August 2006

Diffusions and MCMC

- Diffusions as limits of MCMC algorithms
- Diffusions as motivation for the construction of MCMC algorithms
- MCMC for inference for diffusions

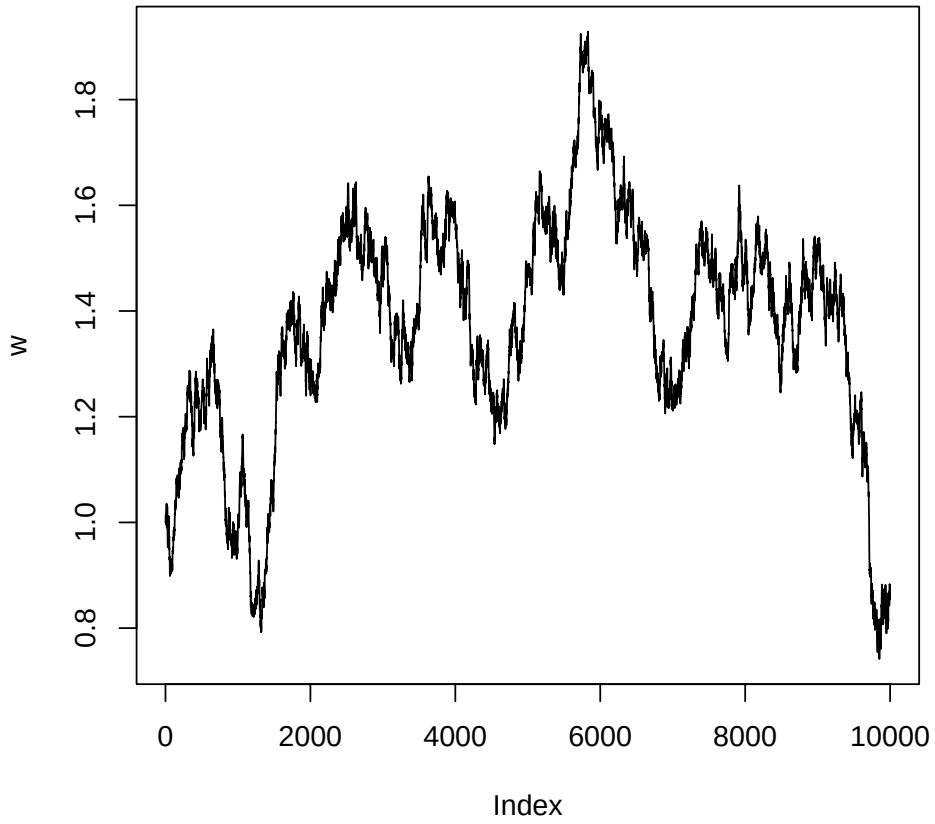


Figure 1:

Why 'limits' of MCMC algorithms?

- Useful for understanding algorithms
- Useful for comparing different algorithms
- Can be used to guide implementation

A very loose classification of some common algorithms:

	Local	Global
Vanilla	Random walk Metropolis	Independence sampler
Problem specific	Langevin algorithms	Gibbs sampler/ IID

Diffusion limit results exist for most of these algorithms

Scaling problems arise for local algorithms.

Metropolis-Hastings algorithm

Given a target density $\pi(\cdot)$ that we wish to sample from, and a Markov chain transition kernel density $q(\cdot, \cdot)$, we construct a Markov chain as follows. Given X_n, generate Y_{n+1} from $q(X_n, \cdot)$. Now set $X_{n+1} = Y_{n+1}$ with probability

$$\alpha(X_n, Y_{n+1}) = 1 \wedge \frac{\pi(Y_{n+1})q(Y_{n+1}, X_n)}{\pi(X_n)q(X_n, Y_{n+1})} \ .$$

Otherwise set $X_{n+1} = X_n$.

Symmetric Random Walk Metropolis algorithm

$$q(\mathbf{x}, \mathbf{y}) = q(|\mathbf{y} - \mathbf{x}|)$$

The acceptance probability simplifies to

$$\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}$$

For example $q \sim MVN_d(\mathbf{x}, \sigma^2 I_d)$.

Algorithm is geometrically ergodic "most of the time" when the tails of the target density are no heavier than exponential.

MALA (Metropolis adjusted Langevin)

(for example Besag, 1994, R and Tweedie, 1996)

Since the SDE

$$d\mathbf{X}_t = d\mathbf{B}_t + \nabla \log \pi(\mathbf{X}_t)\ dt/2$$

has stationary distribution π, why not use a proposal distribution based on a discrete approximation (the Euler approximation) of this?

$$q(\mathbf{x}, \cdot) \sim MVN_d(\mathbf{x} + \sigma^2 \nabla \log \pi(\mathbf{x})/2, \sigma^2 I_d)$$

say.

A broader class of Langevin diffusions exist with stationary distribution π. Alternative discretisations exist too. (Andrew Stuart, Jochen Voss talks.)

The Goldilocks dilemma

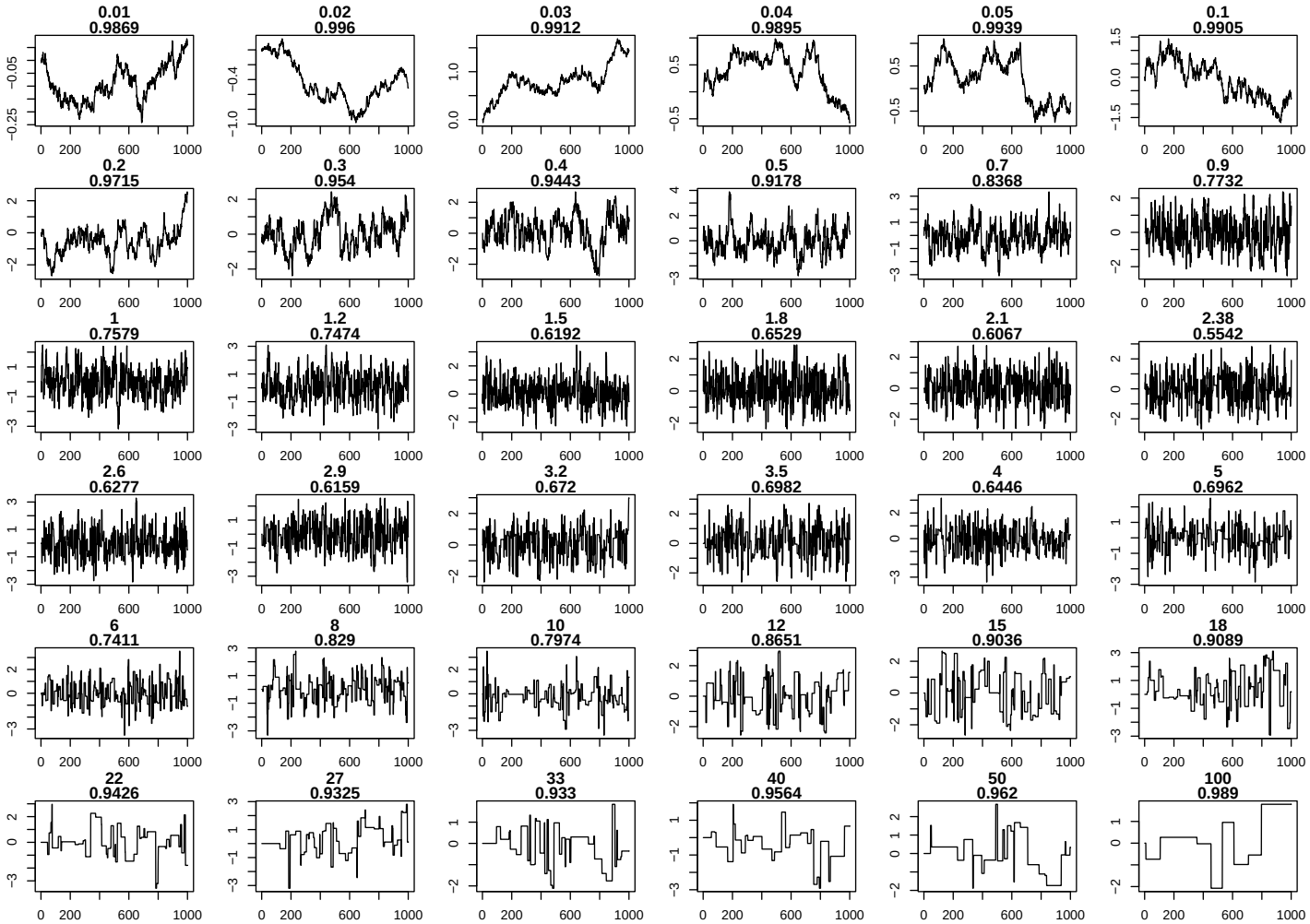


Figure 2:

Scaling problems and diffusion limits

Choosing σ in the above algorithms to optimise efficiency. For 'appropriate choices' the d-dimensional algorithm has a limit which is a diffusion. The faster the diffusion the better!

- How should σ_d depend on d for large d?
- What does this tell us about the efficiency of the algorithm?
- Can we optimise σ_d in some sensible way?
- Can we characterise optimal (or close to optimal) values of σ_d in terms of observable properties of the Markov chain?

For RWM and MALA (and some other local algorithms) and for some simple classses of target distributions, a solution to the above can be obtained by considering a diffusion limit (for high dimensional problems).

Metropolis-within-Gibbs

At each iteration, choose $d \times c_d$ components at random, and update these components according to a Metropolis algorithm which preseves the conditional distribution of those co-ordinates given the rest. The remaining $d(1 - c_d)$ components stay unchanged.

This is not really a generalisation of the Metropolis algorithm.

How should be jointly choose (c_d, σ^2) to optimise the Markov chain?

What is "efficiency"?

Let X be a Markov chain. Then for a π-integrable function f, efficiency can be described by

$$\lim_{n\to\infty} n\text{Var}\left(\frac{\sum_{i=1}^{n} g(X_i)}{n}\right) \ .$$

In general relative efficiency between two possible Markov chains varies depending on what function of interest g is being considered. As $d \to \infty$ the dependence on g disappears, at least in cases where we have a diffusion limit as we will see....

"Efficiency" for diffusions

Consider two Langevin diffusions, both with stationary distribution π.

$$dX_t^i = h_i^{1/2} dB_t + h_i \nabla \log \pi(X_t^i)/2, \quad i = 1, 2,$$

with $h_1 < h_2$.

X^2 is a "speeded-up" version of X^1.

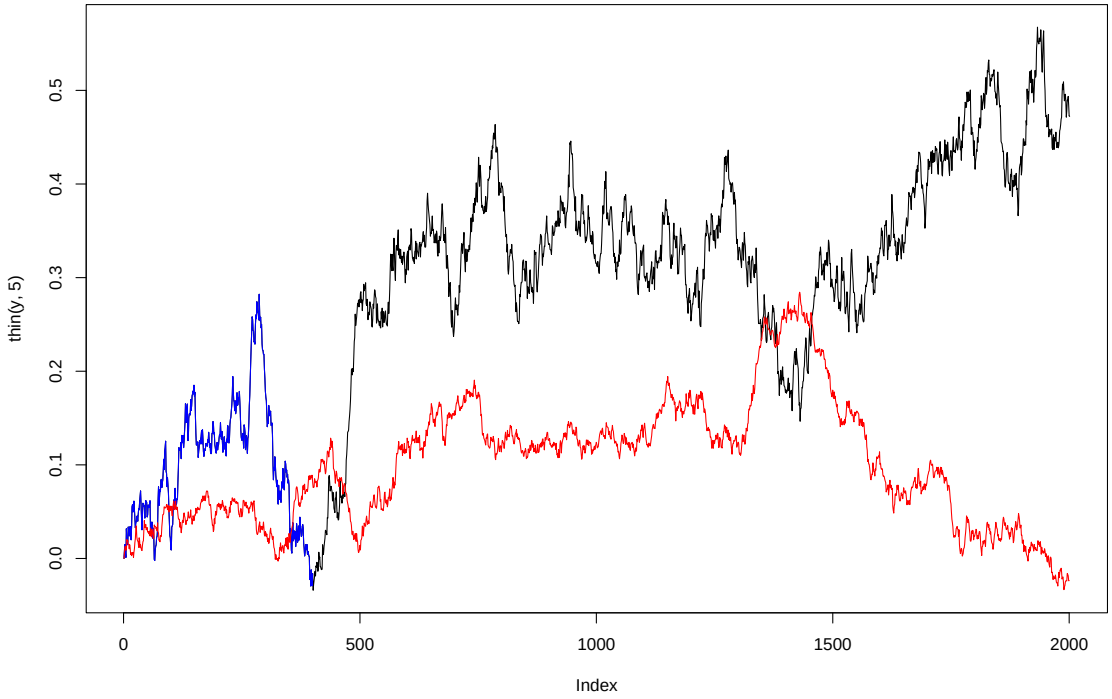


Figure 3:

A diffusion limit

Consider the Metropolis case.

Suppose $\pi \sim \prod_{i=1}^{d} f(x_i)$, $q(\mathbf{x}, \cdot) \sim N(\mathbf{x}, \sigma_d^2 I_d)$, $\mathbf{X}_0 \sim \pi$.

Set $\sigma_d^2 = \ell^2/d$. Consider

$$Z_t^d = X_{[td]}^{(1)} \ . \qquad \text{Speed up time by factor } d$$

Z^d is **not** a Markov chain, however in the limit as d goes to ∞, it is Markov:

$$Z_d \Rightarrow Z$$

where Z satisfies the SDE,

$$dZ_t = h(\ell)^{1/2} dB_t + \frac{h(\ell) \nabla \log f(Z_t)}{2} dt \ ,$$

for some function $h(\ell)$.

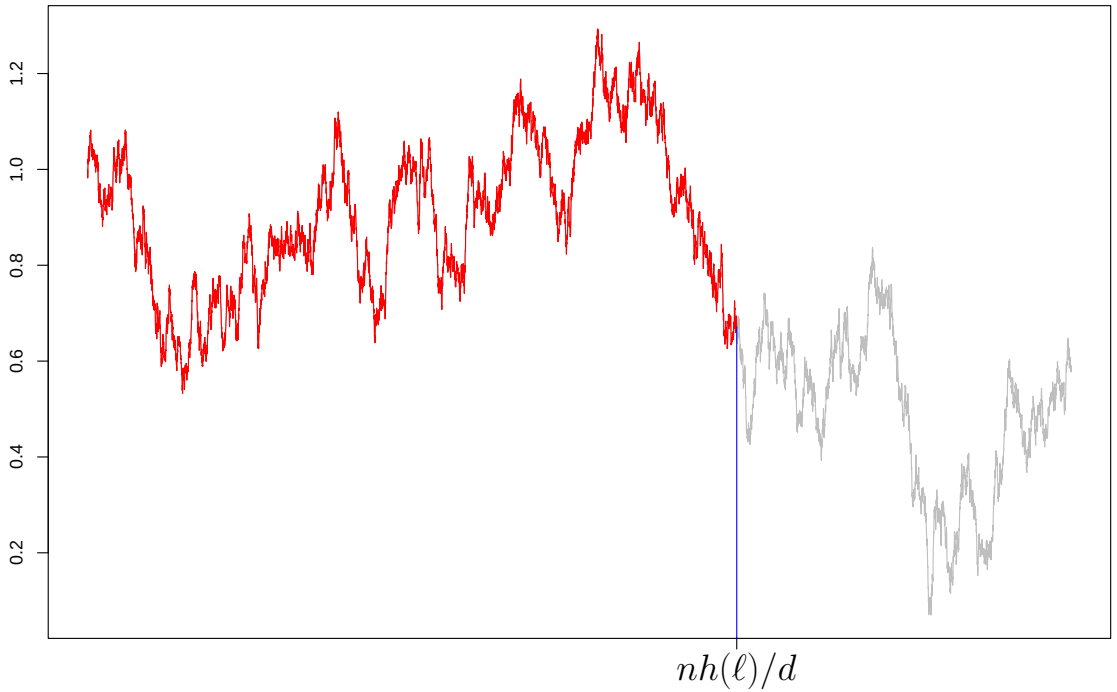


Figure 4: How much diffusion path do we get for our n iterations?

$$h(\ell) = \ell^2 \times 2\Phi\left(-\frac{\sqrt{I}\ell}{2}\right),$$

and $I = E_f[((\log f(X))')^2]$. So

$$h(\ell) = \ell^2 \times A(\ell) \ ,$$

where $A(\ell)$ is the limiting overall acceptance rate of the algorithm, ie the proportion of proposed Metropolis moves ultimately accepted. So

$$h(\ell) = \frac{4}{I}\left(\Phi^{-1}(A(\ell))\right)^2 A(\ell) \ ,$$

and so the maximisation problem can be written entirely in terms of the algorithm's acceptance rate.

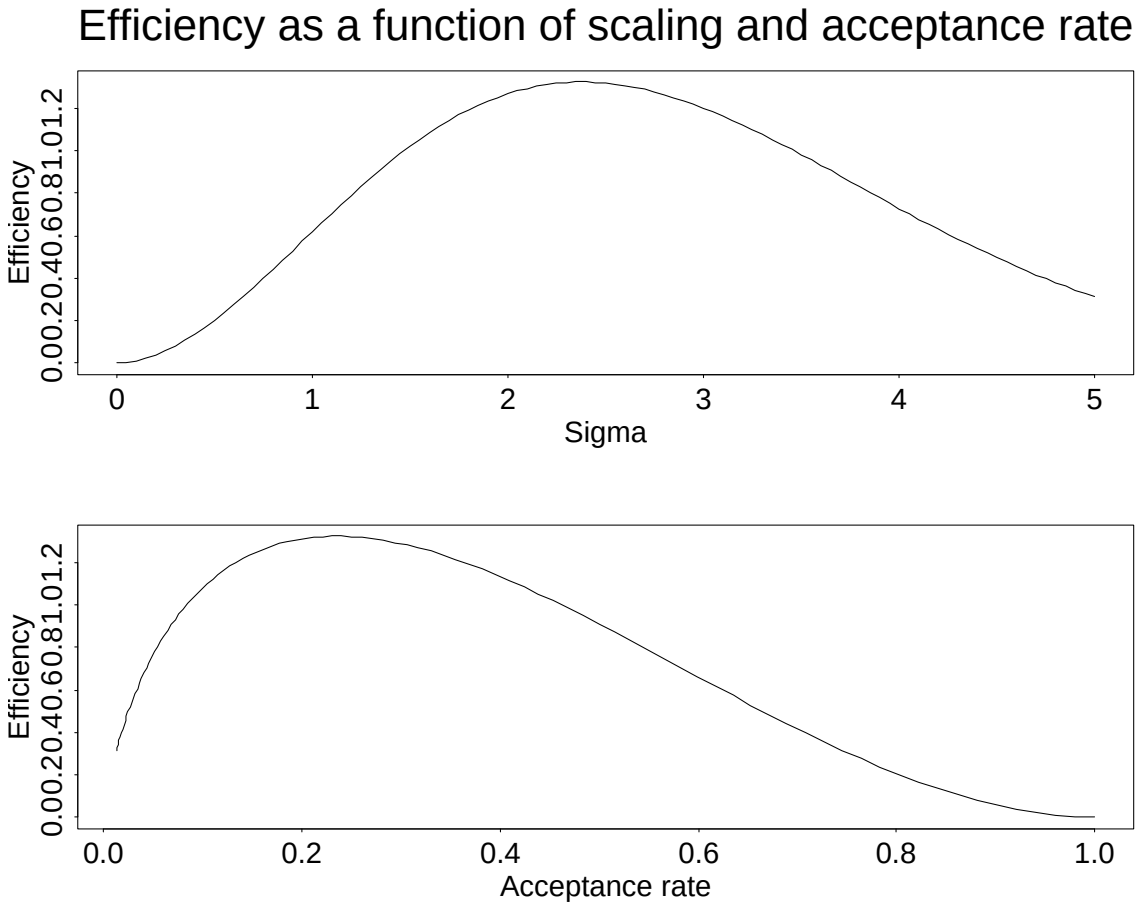

Efficiency as a function of scaling and acceptance rate
Efficiency
0.00.20.40.60.81.01.2
0
1
2
3
4
5
Sigma
Efficiency
0.00.20.40.60.81.01.2
0.0
0.2
0.4
0.6
0.8
1.0
Acceptance rate

Figure 5:

When can we 'solve' the scaling problem?

We need a sequence of target densities π_d which are sufficiently regular as $d \to \infty$ in order that meaningful (and optimisable) limiting distributions exist.

Examples include

1. $\pi \sim \prod_{i=1}^{d} f(x_i)$.
2. $\pi \sim \prod_{i=1}^{d} f(c_i x_i)$, $q(\mathbf{x}, \cdot) \sim N(\mathbf{x}, \sigma_d^2 I_d)$. for some inverse scales c_i. See Mylene Bedard's talk later! Also talks by Jochen Voss and Andrew Stuart
3. Elliptically symmetric target densities. See Chris Sherlock's poster later!
4. The components form a homogeneous Markov chain.
5. π is a Gibbs random field with finite range interactions.
6. Purely discrete product form distributions.

Some questions

- Most results need smoothness conditions on the target. What happens for discontinuous densities?
- Results for 'Metropolis within Gibbs' and 'Langevin within Gibbs'
- What happens to algorithms started out in the tails?
- What happens if we use heavy-tailed proposals?
- What about multivariate scaling problems? See Jeff Rosenthal's talk
- What about scaling in different ways in different parts of the space. See Jeff Rosenthal's talk

Discontinuous target densities

Suppose $\pi \sim \prod_{i=1}^{d} f(x_i)$, with

$$f(x) = \begin{cases} \exp(g(x)), & 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where $g \in C^1[0,1]$.

$q(\mathbf{x}, \cdot) \sim \prod_{i=1}^{d} U(x_i - \sigma_d, x_i + \sigma_d)$, $\mathbf{X}_0 \sim \pi$.

Set $\sigma_d^2 = \ell^2/d^2$. Consider

$$Z_t^d = X_{[td^2]}^{(1)} \, . \qquad \text{Speed up time by factor } d^2$$

$$Z_d \Rightarrow Z$$

where Z satisfies the reflected Langevin SDE on $[0,1]$,

$$dZ_t = h(\ell)^{1/2} dB_t + \frac{h(\ell) \nabla \log f(Z_t)}{2} dt \ ,$$

with

$$h(\ell) = \frac{2\ell^2}{3} \exp\left(-\frac{f^* \ell}{2}\right)$$

and $f^* = \lim_{x \downarrow 0} \left(\frac{f(x) + f(1-x)}{2} \right)$

Partial dimensional updating

At each iteration, choose $d \times c_d$ components at random, and update according to the conditional distribution of those co-ordinates given the rest.

Can we maximise (σ, c_d)?

Suppose $\pi \sim \prod_{i=1}^{d} f(x_i)$.

Set $c_d \sigma_d^2 = \ell^2/d$, $c_d \to c$ as $d \to \infty$. Consider

$$Z_t^d = X_{[td]}^{(1)} \ . \qquad \text{Speed up time}$$

Z^d is **not** a Markov chain, however in the limit as d goes to ∞, it is Markov:

$$Z_d \Rightarrow Z$$

where Z satisfies the SDE,

$$dZ_t = h(\ell)^{1/2} dB_t + \frac{h(\ell) \nabla \log f(Z_t)}{2} dt \ ,$$

for the same function $h(\ell)$ for all c.

So we can do as well just updating a proportion of our components.

Therefore taking into account computing time, full dimensional updating can never be better than strategies which update smaller-dimensional components.

Behaviour in high dimensions

Let T_d be the 'mixing time' for a problem in d dimensions.

- **Random Walk Metropolis** In the best case scenario, for large d need to take $\sigma_d^2 = O(d^{-1})$.

  $$T_d = O(d)$$

  for all choices of $0 < c \leq 1$.

- **Langevin algorithms**

  For large d we need to take $\sigma_d^2 = O((c_d d)^{-1/3})$.

  $$T_d = O(c_d^{-2/3} d^{1/3})$$

So it is typically optimal to update large proportions of components in a Langevin algorithm, even after taking into account computing cost considerations.

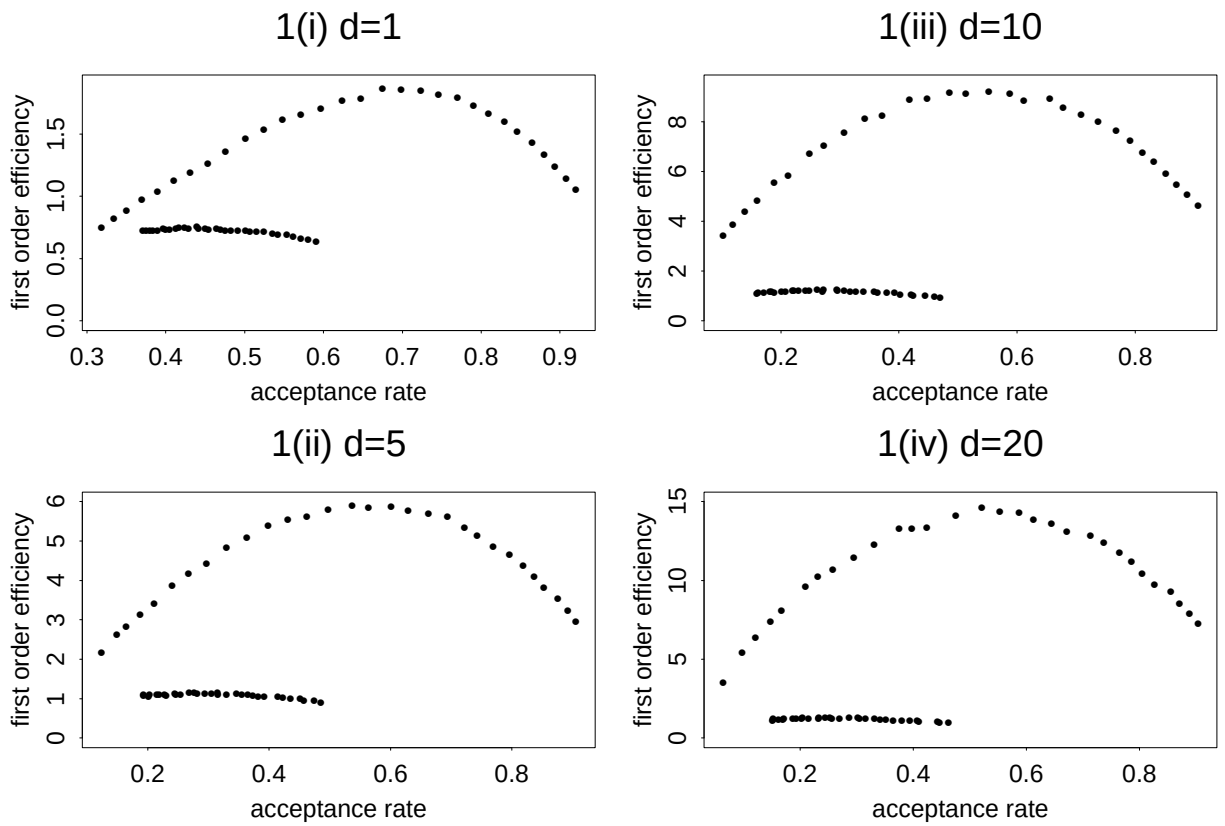


Figure 6: A comparison of Metropolis and Langevin algorithms in terms of efficiency.

Dependence and partial updating

Dependence in target densities makes mixing worse for any partial updating algorithm.

However dependence also affects full-dimensional updating.

Which does it affect most?

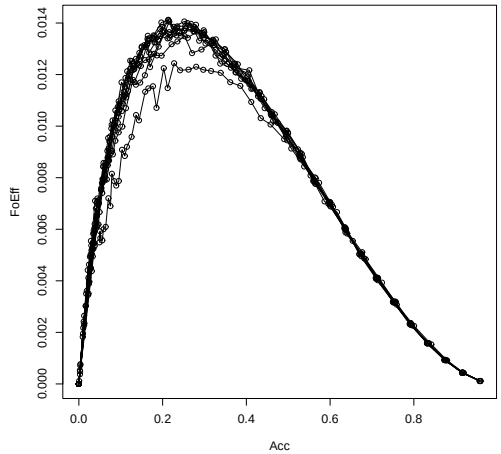


Figure 7: Efficiency of RWM-within-Gibbs as a function of overall acceptance rates for $c = 0.1, 0.2, \ldots, 1$ with $\pi \sim t_{50}(\mathbf{0}, \Sigma_{0.5})$.

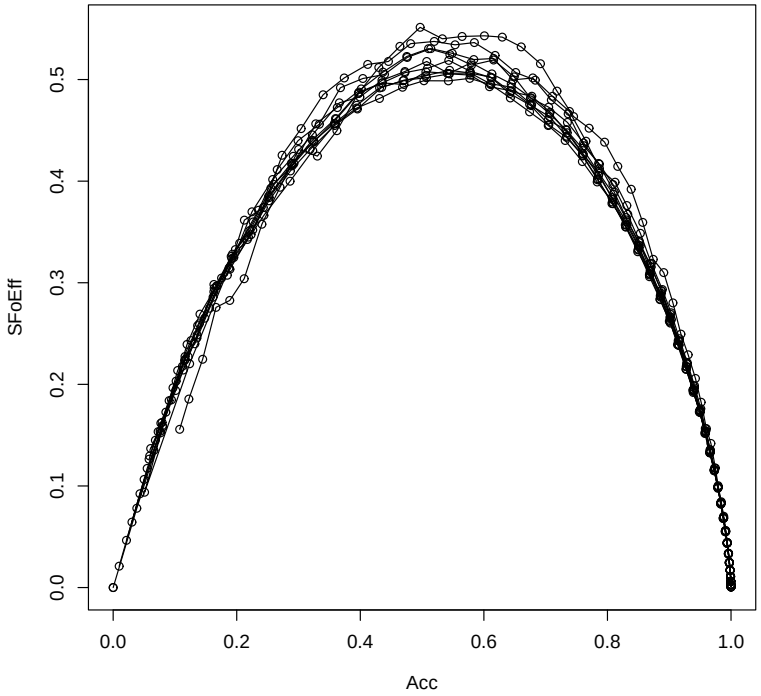


Figure 8: Normalised efficiency of MALA-within-Gibbs, $c^{-\frac{2}{3}}\mathbf{E}[(X_{t+1}^1 - X_t^1)^2]$, as a function of overall acceptance rates for $c = 0.1, 0.2, \ldots, 1$ with $\pi \sim N(\mathbf{0}, \Sigma_0)$.

Non-stationary initial distribution

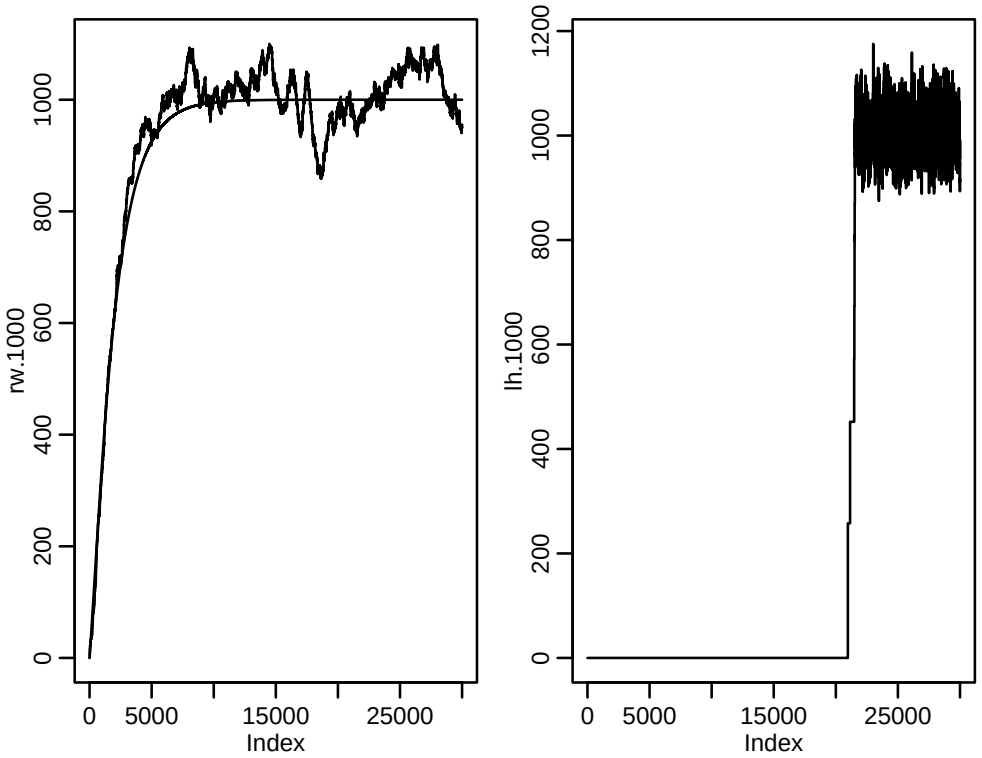


Figure 9:

Gaussian example

Set $\pi \sim MVN_d(\mathbf{0}, I_d)$. Suppose we apply 'optimally scaled' RWM.

Consider $W_t^d = |\mathbf{X}_{[td]}|^2/\mathrm{d}$

Theorem When $W_0^d = w_0 \neq 1$, then as $d \to \infty$, we have $W^d \Rightarrow f$, where f is a deterministic function satisfying $f(0) = w_0$ and

$$f'(t) \;=\; a_\ell(f(t))$$

with function $a_\ell(\cdot)$ which can be explicitly calculated.

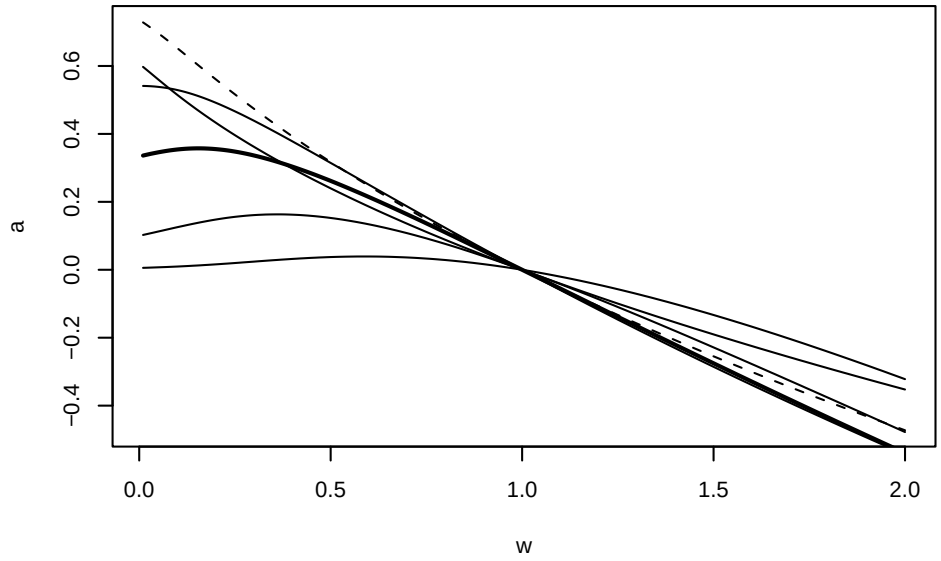


Figure 10: Deterministic convergence speed, $a_\ell(\cdot)$

Langevin case

Using the 'optimal' scaling it gets stuck...

Though using the scaling $\sigma_d^2 = \ell^2/d^{1/2}$, we get a similar deterministic limit result.

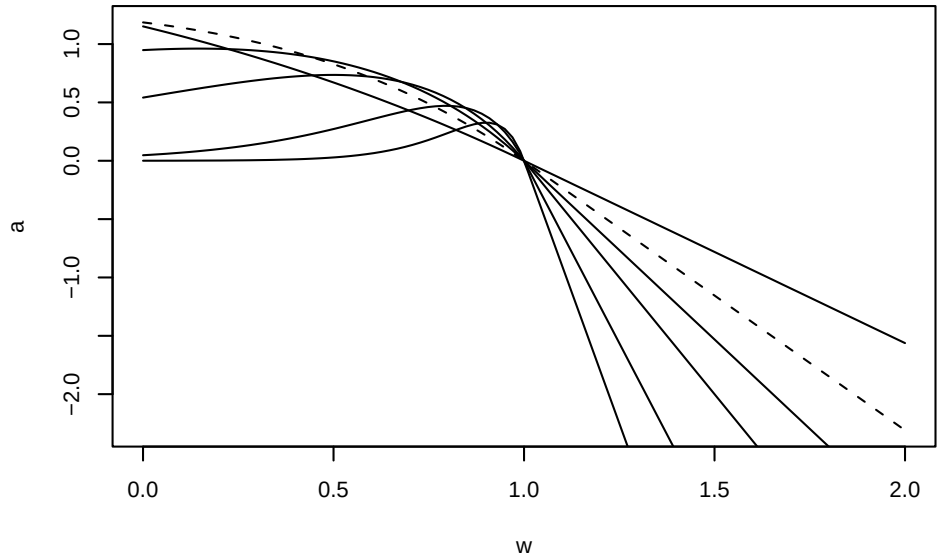


Figure 11: Deterministic convergence speed, $a_\ell(\cdot)$, the Langevin case.

A Point Process Example

From Møller, Syversveen and Waagepetersen (1998 Sc. J. Stat.) Locations of 126 Scots pine saplings in a Finnish forest

Observed point pattern modelled as a Poisson point process X with intensity

$$\Lambda(s) = \exp(Y(s)),$$

where $Y(\cdot) = \{Y(s) \mid s \in \mathbf{R}^2\}$ is a Gaussian process with mean $\mathrm{E}[Y(s)] = \mu$ and covariance

$$\mathrm{Cov}(Y(s), Y(s')) = \sigma^2 \exp(-\|s - s'\|/\beta).$$

The latent Gaussian process is discretised on a 64×64 regular grid.

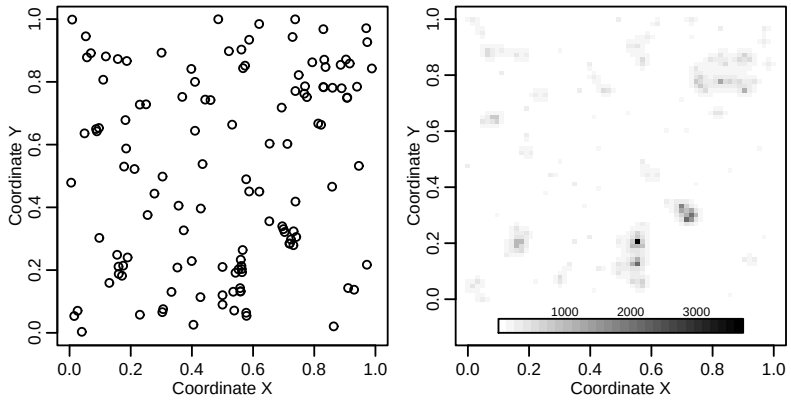


Figure 12: Scotish pine saplings. Left : locations of trees. Right : the estimated intensity $\mathbf{E}[\Lambda(s) \mid x]$.

Updating latent Gaussian field requires MALA updates.

Compare the performance of the algorithm for three different starting values. The starting values expressed in terms of Y (which have to be transformed to starting values for Γ) are

I : $Y_{i,j} = \mu$ for $i, j = 1, \ldots, 64$.

II : a random starting value, simulated from the prior $Y \sim \mathrm{N}(\mu, \Sigma)$.

III : a starting value near the posterior mode. Let $Y_{i,j}$ solve the equation $0 = x_{i,j} - \exp(Y_{i,j}) - (Y_{i,j} - \beta)/\sigma^2$.

In all three cases we use the scaling $\hat{\ell}^2/(4096)^{1/3} = 0.16$ where $\hat{\ell} = 1.6$ is derived using 'optimal scaling' criteria.

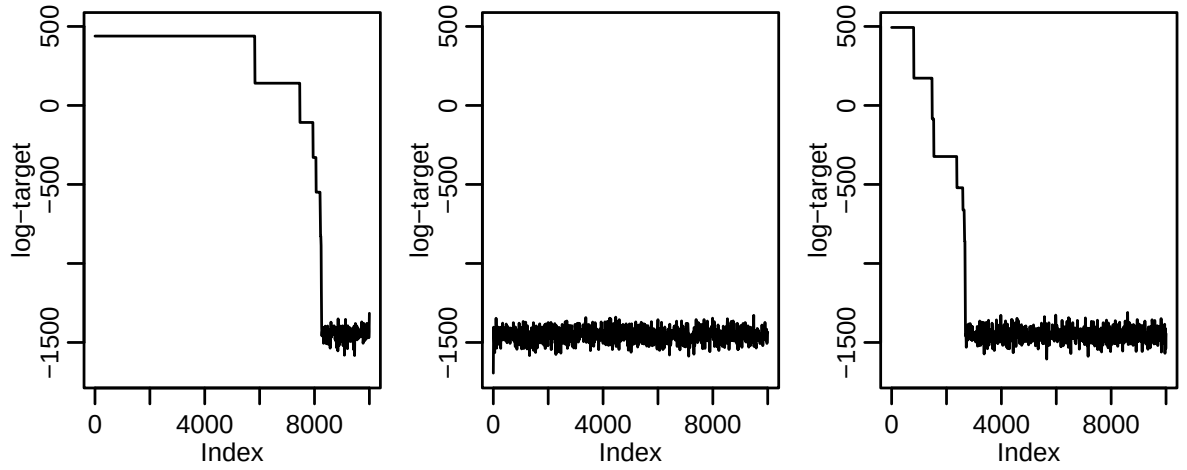


Figure 13: Scots pine saplings. Traceplots $\log(\gamma \mid x)$ when using the scaling 0.16. Left : starting value I. Middle : starting value II. Right : starting value III.

Now using the scaling $\hat{\ell}^2/(4096)^{1/2} = 0.034$. The acceptance rate for all algorithms was around 95%.

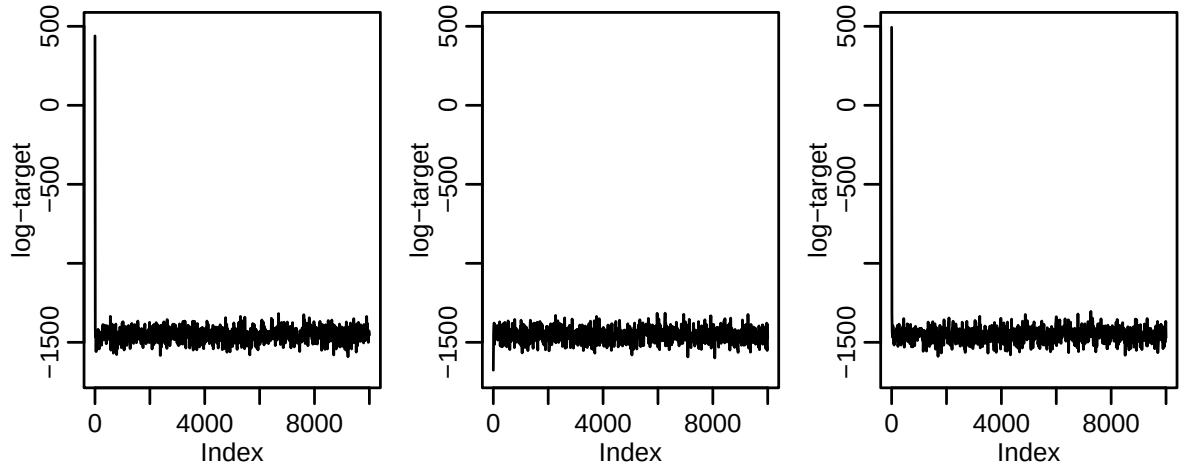
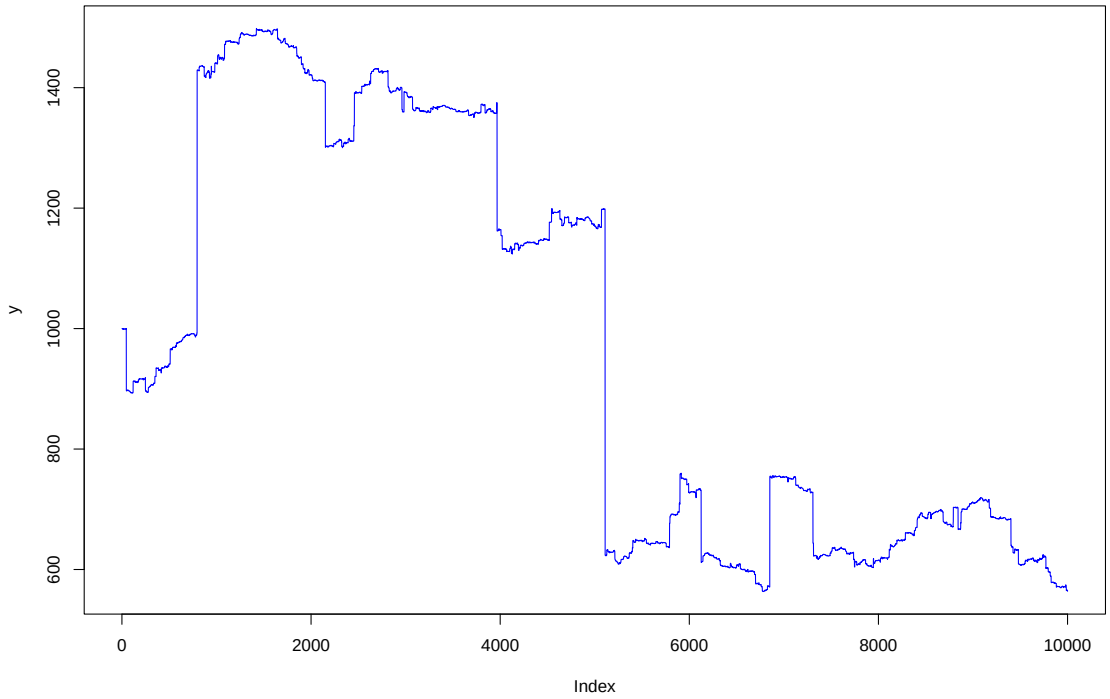
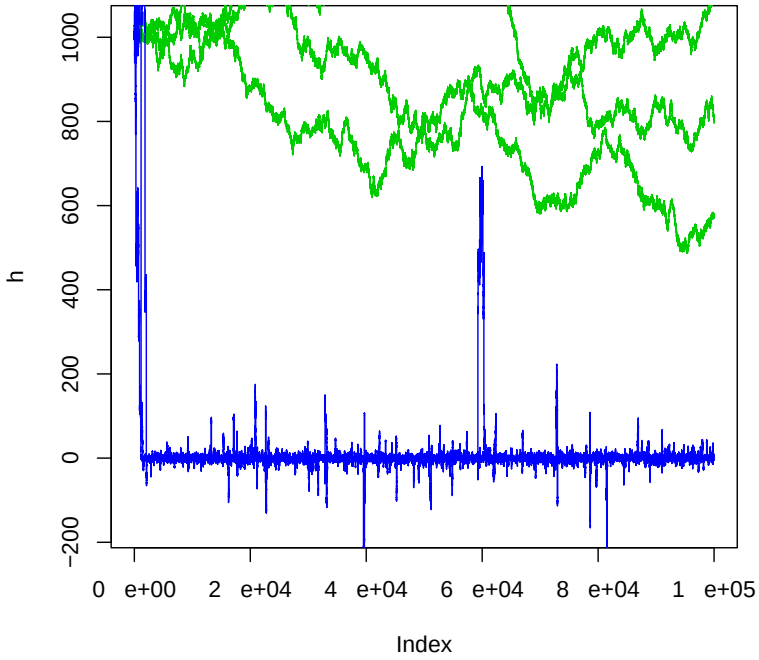


Figure 14: Scots pine saplings. Traceplots $\log(\gamma \mid x)$ when using the scaling 0.034. Left : starting value I. Middle : starting value II. Right : starting value III.

Heavy-tailed proposals

If proposal variance is infinite, all the above theory fails and diffusion limits cannot exist!





To fix ideas, consider RWM, and replace independent Gaussian proposals in each direction by independent Cauchy proposals in each direction.

Evidence from other results that heavy-tailed proposals improve mixing (eg Jarner and R, 2003, 2006, talk by Gersende Fort).

Discontinuous targets, heavy-tailed proposals

Suppose $\pi \sim \text{Unif}(0,1)^d$.

$q(\mathbf{x}, \cdot) \sim \text{Cauchy}(\mathbf{x}, \sigma_d^2 I_d)$, $\mathbf{X}_0 \sim \pi$.

Set $\sigma_d^2 = \ell^2/d\log d$. Consider

$$Z_t^d = X^{(1)}_{[td\log(d)]} \ . \quad \text{Speed up time by factor } d\log d$$

$$Z_d \Rightarrow \text{ a scaled truncated Cauchy process}$$

with an associated explicit optimal scaling problem.

Here, light-tailed proposals are $O(d^2)$ while Cauchy proposals are $O(d\log d))$.

Final comments

Smarter Langevin methods exist and can solve some of the Langevin mixing problems. See talks by Jochen Voss and Andrew Stuart

Do we really want our algorithms to 'look like diffusions'?

Inevitably much of the practical importance of this work lies to problems which lie beyond the nice classes of problems for which clean diffusion limits exist and for which the scaling problem can be rigorously solved.

Jeff Rosenthal will talk about the use of this theory in adaptive MCMC methods.