Optimising and Adapting the Metropolis Algorithm

Jeffrey S. Rosenthal

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

23 June 2010
Newton Institute, Cambridge, U.K.
**Motivation**

Given some complicated, high-dimensional density function $\pi : \mathcal{X} \to [0, \infty)$, for some $\mathcal{X} \subseteq \mathbb{R}^d$ 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 :

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

Calculus? Numerical integration?

Impossible! Typical $\pi$ is something like ...
Typical $\pi$: Variance Components Model

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

with, say, $K = 19$, $d = 22$.

High-dimensional! Complicated! What to do?
Estimation from sampling: Monte Carlo

Can try to sample from $\pi$, i.e. generate i.i.d.

$$X_1, X_2, \ldots, X_M \sim \pi$$

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

Then can estimate by e.g.

$$E_{\pi}(h) \approx \frac{1}{M} \sum_{i=1}^{M} h(X_i).$$

Good. But how to sample? Often infeasible!

Instead …
Markov chain Monte Carlo (MCMC)

Define a Markov chain $X_0, X_1, X_2, \ldots$, such that for large $n$, $P(X_n \in A) \approx \int_A \pi(x) \, dx$.

(Just approximate ... and not i.i.d.)

Still, hopefully for $M \gg B \gg 1$,

$$E_\pi(h) \approx \frac{1}{M - B} \sum_{i=B+1}^{M} h(X_i).$$

But how to define a simple Markov chain such that

$$P(X_n \in A) \rightarrow \int_A \pi(x) \, dx$$

(4/22)
The Metropolis Algorithm

\( \pi = \text{target density (important! complicated! high-dim!)} \)

Goal: obtain samples from \( \pi \).

The algorithm: for \( n = 1, 2, 3, \ldots, \)

• \( Y_n := X_{n-1} + Z_n, \) where \( Z_n \sim Q \) (i.i.d., symmetric)

• \( \alpha := \min \left( 1, \frac{\pi(Y_n)}{\pi(X_{n-1})} \right) \)

• with probability \( \alpha \), \( X_n := Y_n \) (“accept”)

• else, with probability \( 1 - \alpha \), \( X_n := X_{n-1} \) (“reject”)

Assuming “irreducibility”, have \( \mathbb{P}(X_n \in A) \rightarrow \pi(A) \).

Good!

(5/22)
Example #1: Java applet

\( \pi(\cdot) \) simple distribution on \( \mathcal{X} = \{1, 2, 3, 4, 5, 6\} \).
[Take \( \pi(x) = 0 \) for \( x \notin \mathcal{X} \).]

\( Q(\cdot) = \text{Uniform}\{-1, 1\}. \ [\text{APPLET}] \)

Works.

But what if \( Q(\cdot) = \text{Uniform}\{-2, -1, 1, 2\} \).

Or, \( Q(\cdot) = \text{Uniform}\{-\gamma, -\gamma + 1, \ldots, -1, 1, 2, \ldots, \gamma\} \).

Which \( \gamma \) is best? ("optimise")

Good \( \gamma \) is between the two extremes, i.e. acceptance rate should be far from 0 and far from 1.
("Goldilocks Principle")

(6/22)
Example #2: $N(0,1)$

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

- $\sigma = 0.1$?
  - too small!
  - A.R. = 0.962

- $\sigma = 25$?
  - too big!
  - A.R. = 0.052

- $\sigma = 2.38$?
  - (better!)
  - A.R. = 0.441

What about higher dimensions? (need smaller $\sigma$ ...)
How to make theoretical progress?

Consider diffusion limits!

**Analogy**: 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.

**Theorem** [Roberts, Gelman, Gilks, AAP 1994] :

If \( \{X_n\} \) is a Metropolis algorithm in high dimension \( d \), with \( Q(\cdot) = N(0, \frac{\ell^2}{d} I_d) \), and \( Z_t = d^{-1/2} X^{(1)}_{dt} \), then under “certain conditions” on \( \pi(\cdot) \), the process \( \{Z_t\} \) converges to a diffusion.
More precisely, as $d \to \infty$, $Z_t = d^{-1/2} X_{dt}^{(1)}$ converges to a Langevin diffusion which satisfies:

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

where

$$\text{speed} = h(\ell) = 2 \ell^2 \Phi(-C_\pi \ell/2)$$

and

$$\text{acceptance rate} \equiv A(\ell) = 2 \Phi(-C_\pi \ell/2).$$

(Here $C_\pi$ depends on $\pi(\cdot)$, and $\Phi(x) = \int_{-\infty}^x \frac{e^{-u^2/2}}{\sqrt{2\pi}} \, du$.)

**Key point**: algorithm’s speed $h(\ell)$ is explicitly related to its asymptotic acceptance rate $A(\ell)$. 

(9/22)
Lots of information here!

- The speed $h(\ell)$ is related to the acceptance rate $A(\ell)$.
- To optimise the algorithm, we should maximize $h(\ell)$.
- The maximization is easy: $\ell_{opt} = 2.38/C_{\pi}$.
- Then we can compute that: $A(\ell_{opt}) = 0.234$.

So, for $Q(\cdot) = N(0, \sigma^2 I_d)$, it is **optimal** to choose

$$\sigma^2 = \frac{\ell_{opt}^2}{d} = \frac{(2.38)^2}{(C_{\pi})^2 d},$$

which leads to an acceptance rate of 0.234.

Clear, simple rule – good!

(Also shows algorithm’s running time is $O(d)$.)
What are these “conditions” on $\pi$?

Original result: $\pi(x) = \prod_{i=1}^{d} f(x_i)$ for fixed $f$ (i.i.d.).
Very restrictive, artificial condition.

Some generalizations (Bédard, AAP 2007):
$\pi(x) = \prod_{i=1}^{d} \theta_i(d) f(\theta_i(d) x_i)$, where certain $\{\theta_i(d)\}$ repeat more and more as $d \rightarrow \infty$. More flexible! (Also, for certain other cases, 0.234 is no longer optimal: Bédard, SPA 2008.)

Anyway, 0.234 is often nearly optimal, even if the theorem conditions are not satisfied. (“robust”)

But does acceptance rate tell us everything?
Example #3 : $\pi = N(0, \Sigma)$ in dimension 20

First try : $Q(\cdot) = N(0, I_{20})$ (acc rate = 0.006)

Horrible : $\Sigma_{11} = 24.54$, $E(X_1^2) = 1.50$. 
Second try: $Q(\cdot) = N\left(0, (0.0001)^2 I_{20}\right)$ (acc=0.892)

Also horrible: $\Sigma_{11} = 24.54$, $E(X_1^2) = 0.0053$. 
Third try: \( Q(\cdot) = N\left(0, (0.02)^2 I_{20}\right) \) (acc=0.234)

Still poor: \( \Sigma_{11} = 24.54 \), \( E(X_1^2) = 3.63 \).
Fourth try: $Q(\cdot) = N\left(0, \left[(2.38)^2/20\right] \Sigma\right)$ (acc=0.263)

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

(15/22)
Optimal Proposal Covariance

Theorem [Roberts and R., Stat Sci 2001]:
Under certain conditions on $\pi(\cdot)$, the optimal Metropolis algorithm Gaussian proposal distribution as $d \to \infty$ is

$$Q(\cdot) = N\left(0, \left(\frac{(2.38)^2}{d}\right) \Sigma\right).$$

(Not $N(0, \sigma^2 I_d)$ . . .) Furthermore, with this choice, the asymptotic acceptance rate is again 0.234.

And, optimal / nearly optimal for many other high-dimensional densities, too.

But this only helps if $\Sigma$ is known!

What if it isn’t??
How to use this result if $\Sigma$ is unknown?

Use adaptive MCMC! (Haario et al., Bernoulli 2001)

- Replace $\Sigma$ by the empirical estimator $\Sigma_n$.
- Hope that for large $n$, we have $\Sigma_n \approx \Sigma$.
- Then $N\left(0, ((2.38)^2/d)\Sigma_n\right) \approx N\left(0, ((2.38)^2/d)\Sigma\right)$.
- So, use this proposal instead!

Are we allowed to do this?? (Subtle, because the process is no longer Markovian.)

- In examples, it usually works well ... (next page)
- But not always ... [APPLET]
Good adaptation in dimension 200...
Is Adaptive MCMC Valid??

**Theorem** [Roberts and R., J Appl Prob 2007]: Yes, any adaptive MCMC converges asymptotically to $\pi(\cdot)$, assuming:

1. “Diminishing Adaptation”: Adaption chosen so that
   \[
   \lim_{n \to \infty} \sup_{x \in \mathcal{X}} \sup_{A \subseteq \mathcal{X}} |P_{\Gamma_{n+1}}(x, A) - P_{\Gamma_n}(x, A)| = 0 \text{ (in prob.)}
   \]

2. “Containment”: Times to stationary from $X_n$, if we fix $\gamma = \Gamma_n$, remain bounded in probability as $n \to \infty$. [Technical condition. Satisfied e.g. under compactness and continuity.]

Meanwhile, in applications, adaption often leads to significant speed-ups, even in hundreds of dimensions (Roberts and R., JCGS 2009; Richardson, Bottolo, R., Valencia 2010).
**Another application: Simulated Tempering**

Simulated Tempering: replace $\pi$ by a family $\{\pi^{\beta_i}\}_{i=1}^m$, with $0 \leq \beta_m < \beta_{m-1} < \ldots < \beta_0 = 1$.

Here $\pi^{\beta_m}$ is the “hot” distribution (easily sampled). And $\pi^{\beta_0} = \pi$ is the “cold” distribution (the distribution of interest, but hard to sample).

Hope the algorithm can move efficiently between the different $\pi^{\beta_i}$, so it can “benefit” from $\pi^{\beta_m}$ to efficiently explore $\pi^{\beta_0}$.

**Question**: how to choose the values $\beta_i$?

Often chosen to be “geometric”: $\beta_i = a^i$ for $0 < a < 1$.

**Theorem** [Atchadé, Roberts, R., Stat & Comput 2010]: optimal to choose $\{\beta_i\}$ so that the asymptotic acceptance rate for moves $\beta_i \mapsto \beta_{i \pm 1}$ is 0.234. (Not necessarily geometric!)
Langevin Algorithms

If possible, it’s more efficient to use a non-symmetric proposal distribution, inspired by Langevin diffusions:

\[ Y_n = X_{n-1} + \sigma Z_n + \frac{\sigma^2}{2} \nabla \log \pi(X_{n-1}). \]

**Theorem** [Roberts and R., JRSSB 1997] :
Optimal choice is now \[ \sigma = \ell d^{-1/6} \] (not \( \sigma = \ell d^{-1/2} \)), and \( A(\ell_{opt}) = 0.574 \) (not \( A(\ell_{opt}) = 0.234 \)).

In this case, the algorithm’s running time is \( O(d^{1/3}) \), not \( O(d) \), with optimal acceptance rate 0.574, not 0.234.
Summary

• The Metropolis algorithm is very important.
• The optimisation of the algorithm can be crucial.
• Want acceptance rate far from 0, far from 1.
• Various theorems tell us how to optimise under certain conditions: $0.234, N\left(0, \frac{(2.38)^2 \Sigma}{d}\right)$, etc.
  • Even if some information is unknown (e.g., $\Sigma$), can still adapt towards the optimal choice; valid if the adaption satisfies “Diminishing Adaptation” and “Containment”.
  • Can lead to tremendous speed-up in high dimensions.
• Application to computing rare tail probabilities of $\pi(\cdot)$.

http://probability.ca/jeff/