STSCI 4780
Bayesian computation:
Markov Chain Monte Carlo, 2

Tom Loredo, CCAPS & SDS, Cornell University

Notation focusing on computational tasks

$$
\begin{aligned}
p(\theta|D, M) &= \frac{p(\theta|M)p(D|\theta, M)}{p(D|M)} \\
&= \frac{\pi(\theta)\mathcal{L}(\theta)}{Z} = \frac{q(\theta)}{Z}
\end{aligned}
$$

- $M =$ model specification
- D specifies observed data
- $\theta =$ model parameters
- $\pi(\theta) =$ prior pdf for θ
- $\mathcal{L}(\theta) =$ likelihood for θ (likelihood function)
- $q(\theta) = \pi(\theta)\mathcal{L}(\theta) =$ "quasiposterior"
- $Z = p(D|M) =$ (marginal) likelihood for the model

Line of argument

- Bayesian computation largely amounts to computing expectations (up to a factor $1/Z$):

$$\int \mathrm{d}^m\theta \; g(\theta)\,\pi(\theta)\,\mathcal{L}(\theta) \;\; = \;\; \int \mathrm{d}^m\theta \; g(\theta)\,q(\theta)$$

- Intuition: Approximate the *expectation* (integral over θ) with a *sample mean* (N-term sum, whose distribution involves an N-D integral over N copies of θ)

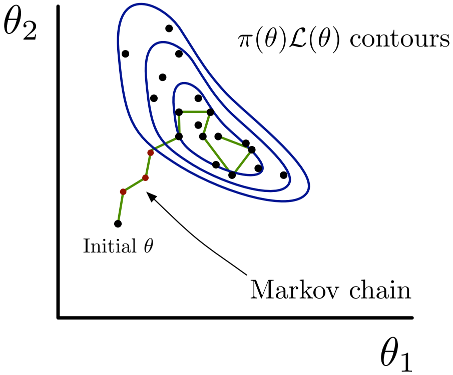
$$\int \mathrm{d}\theta \; g(\theta)p(\theta) \approx m \equiv \frac{1}{N}\sum_{\theta_i \sim p(\theta)} g(\theta_i)$$

- Justification: $\mathbb{E}(m) = \mu$ for $\{\theta_i\}$ from *any* joint distribution with *identical marginals* equal to $p(\theta)$, and $\mu \equiv \int \mathrm{d}\theta\, g(\theta)\, p(\theta)$

- Simplest option: *IID Monte Carlo*, for which we can demonstrate *convergence*
    - Weak LLN shows probability for large error between m and μ is bounded by σ^2/N (for any N)
    - CLT shows the distribution for the error becomes Gaussian at large N, with $\sigma_m = \sigma/\sqrt{N}$
- IID sampling from $p(\theta)$ is feasible in 1-D, but increasingly challenging as dimension increases
  E.g.: Accept/reject will be inefficient unless we can find an envelope that resembles $p(\theta)$ — *hard!*

- Since $\mathbb{E}(m) = \mu$ holds for any joint dist'n *with identical marginals matching $p(\theta)$*, consider a simple dependent class: *stationary/homogeneous Markov chains*
  - *Given a transition matrix T*, under weak conditions there is an *equillibrium distribution*
  - Weak convergence result: Even if we don't start with draws from p_{eq}, eventually the marginals converge to p_{eq}
  - $\Rightarrow$ *Markov chains can give us dependent draws with identical marginals* (eventually/approximately!)
  - Today we reverse the viewpoint: How can we design a transition so a Markov chain will *have the posterior as its equillibrium dist'n*?
    I.e.: *Given $p_{\text{eq}} = p(\theta)$*, find a T that has this eq. dist'n

Goal: Posterior sampling via Markov chain Monte Carlo



Reversibility/Detailed Balance

A sufficient (but not necessary) condition for there to be an equilibrium distribution is the *detailed balance* or *reversibility* condition

$$\begin{array}{rcl} p_{\mathrm{eq}}(x)T(y|x) & = & p_{\mathrm{eq}}(y)T(x|y) \qquad \textit{or} \\ \dfrac{T(y|x)}{T(x|y)} & = & \dfrac{p_{\mathrm{eq}}(y)}{p_{\mathrm{eq}}(x)} \end{array}$$

If we set $p_{\mathrm{eq}} = q/Z$, and we build a transition distribution that is reversible for this choice, then *the equilibrim distribution will be the posterior distribution*

Note that this condition is nontrivial only for $y \neq x$; DB doesn't directly constrain the probability for staying in the same place (but it must be < 1!)

DB implies the eigenvalue condition

The detailed balance condition is:

$$p_{\mathrm{eq}}(x)\,T(y|x) \;=\; p_{\mathrm{eq}}(y)\,T(x|y)$$

Sum both sides over x:

$$\begin{aligned}
\sum_x p_{\mathrm{eq}}(x)\,T(y|x) &= \sum_x p_{\mathrm{eq}}(y)\,T(x|y) \\
&= p_{\mathrm{eq}}(y) \sum_x T(x|y) \\
&= p_{\mathrm{eq}}(y)
\end{aligned}$$

In matrix notation:

$$T \cdot \vec{p}_{\mathrm{eq}} = \vec{p}_{\mathrm{eq}}$$

This says $\vec{p}_{\mathrm{eq}}$ is an eigenvector of T with eigenvalue 1

Designing Reversible Transitions

Set $p_{\mathrm{eq}}(x) = q(x)/Z$; how can we build a $T(y|x)$ with this as its EQ dist'n? I.e., how can we build a T satisfying

$$
\begin{aligned}
\frac{q(x)}{Z} T(y|x) &= \frac{q(y)}{Z} T(x|y) \\
\Rightarrow \quad q(x) T(y|x) &= q(y) T(x|y)
\end{aligned}
$$

Note that Z dropped out—*to build a Markov chain targeting the posterior, we don't need the normalization constant*

This was also true of accept/reject, since only the *relative* areas under the target and proposal matter

The irrelevance of Z also implies that we can't directly use MCMC to compute the value of Z (e.g., for model comparison)—we'll need other approaches

Goal: Build a conditional distribution $T(\cdot|\cdot)$ satisfying

$$q(x)T(y|x) \;\; = \;\; q(y)T(x|y)$$

Steal an idea from **accept/reject**: Start with a *proposal* or candidate distribution, $k(y|x)$

We won't keep all proposals, so *k is not by itself the transition dist'n T*

We'll devise an accept/reject criterion that uses k to build a reversible $T(y|x)$ with equilibrium distribution q/Z

Basic idea: Using an arbitrary $k(y|x)$ as T will not guarantee reversibility. E.g., from a particular x, the transition rate to a particular y *might* be too large:

$$q(x)k(y|x) > q(y)k(x|y)$$

When this is true, we should use rejections to reduce the rate to y

Hopefully this will also prevent too-small rates (we can't *over*accept!)

Acceptance probability: Accept y with probability $\alpha(y|x)$; reject it with probability $1 - \alpha(y|x)$ and stay at x:

$$T(y|x) = k(y|x)\alpha(y|x) + [1 - \alpha(y|x)]\delta_{y,x}$$

The detailed balance condition is a nontrivial requirement for $y \neq x$ transitions, for which $\delta_{y,x} = 0$; it implies a condition for α:

$$q(x)k(y|x)\alpha(y|x) = q(y)k(x|y)\alpha(x|y)$$

For pairs of states where $q(x)k(y|x) > q(y)k(x|y)$, we want $\alpha(y|x)$ to suppress $x \to y$ transitions, but we want $\alpha(x|y)$ to maximize $y \to x$ transitions

Try setting $\alpha(x|y) = 1$; then the detailed balance condition becomes:

$$\alpha(y|x) = \frac{q(y)k(x|y)}{q(x)k(y|x)} \quad \text{when the RHS} < 1$$

If instead $q(x)k(y|x) < q(y)k(x|y)$, the situation is reversed: we adopt $\alpha(y|x) = 1$, and let $\alpha(x|y)$ suppress $y \to x$ transitions

Gather the cases defining $\alpha(y|x)$:

$$\alpha(y|x) = \begin{cases} \frac{q(y)k(x|y)}{q(x)k(y|x)} & \text{if } q(x)k(y|x) > q(y)k(x|y) \\ 1 & \text{otherwise} \end{cases}$$

(Note that swapping the values of x and y shows that $\alpha(x|y)$ is appropriately assigned in the opposite case)

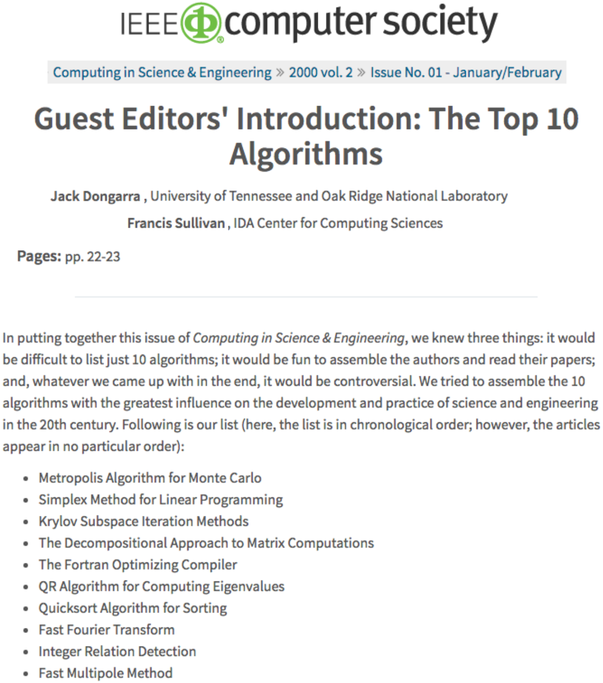
Equivalently:

$$\alpha(y|x) = \min\left[\frac{q(y)k(x|y)}{q(x)k(y|x)}, 1\right]$$

Metropolis-Hastings algorithm

Given a target quasi-distribution $q(x)$ (it need not be normalized):

1. Specify a proposal distribution $k(y|x)$ (make sure it is irreducible and aperiodic).
2. Choose a starting point x; set $t = 0$ and $S_t = x$
3. Increment t
4. Propose a new state $y \sim k(y|x)$
5. If $q(x)k(y|x) < q(y)k(x|y)$, set $S_t = y$; goto (3)
6. Draw a uniform random number u
7. If $u < \frac{q(y)k(x|y)}{q(x)k(y|x)}$, set $S_t = y$; *else set $S_t = x$*; goto (3)



IEEE computer society

Computing in Science & Engineering » 2000 vol. 2 » Issue No. 01 - January/February

Guest Editors' Introduction: The Top 10 Algorithms

Jack Dongarra , University of Tennessee and Oak Ridge National Laboratory

Francis Sullivan , IDA Center for Computing Sciences

Pages: pp. 22-23

In putting together this issue of *Computing in Science & Engineering*, we knew three things: it would be difficult to list just 10 algorithms; it would be fun to assemble the authors and read their papers; and, whatever we came up with in the end, it would be controversial. We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century. Following is our list (here, the list is in chronological order; however, the articles appear in no particular order):

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

Proposal distributions

The art of MCMC is in *specifying the proposal distribution $k(y|x)$*

We want:

- New proposals to be accepted, so there is movement
- Movement to be significant, so we explore efficiently

These desiderata compete! Recall:

$$\alpha(y|x) = \begin{cases} \frac{q(y)k(x|y)}{q(x)k(y|x)} & \text{if } q(x)k(y|x) > q(y)k(x|y) \\ 1 & \text{otherwise} \end{cases}$$

We can get a high acceptance rate by proposing small steps ($k(y|x)$ high only for y near x), so that $q(y) \approx q(x)$ — but then each step will be small

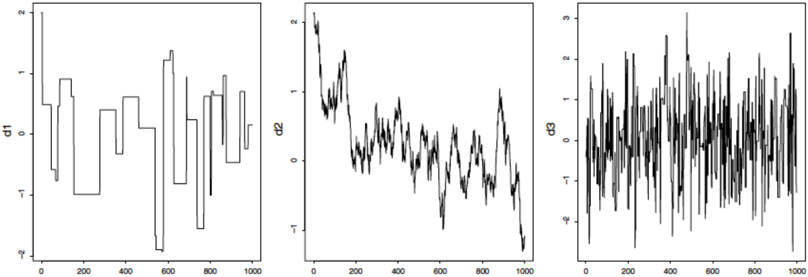
We can propose large steps — but then $q(y)$ and $q(x)$ are likely to be very different, and if the data are informative, large steps will likely take us away from high-density regions, so large steps are unlikely to be accepted

Optimal acceptance rate

Rate too small
chain gets stuck

Rate too large
chain moves too slowly

Optimal rate
Efficient exploration,
“Good mixing”



(a) Proposal variance too large

(b) Proposal variance too small

(c) Proposal variance approximately optimised

Roberts & Rosenthal (2001)

Metropolis algorithm

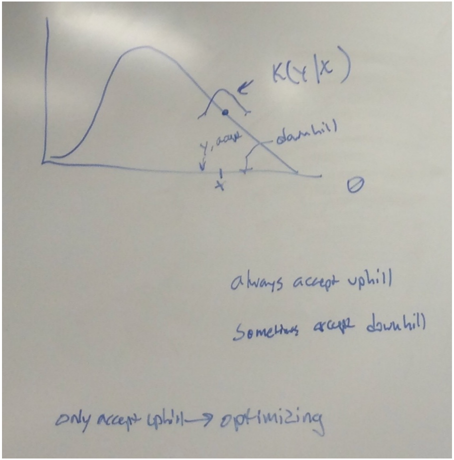
Suppose a symmetric proposal is adopted:

$$k(y|x) = k(x|y)$$

The acceptance probability simplifies:

$$\alpha(y|x) = \min\left[\frac{q(y)}{q(x)}, 1\right]$$

This special case of MH is the *Metropolis algorithm* (Hastings generalized it later)



always accept uphill

Sometimes accept downhill

only accept uphill → optimizing

(There is an optimization algorithm that resembles the Metropolis algorithm in allowing downhill steps (that might enable escape from a local extremum): *simulated annealing*. To make a Metropolis-like algorithm optimize, it slowly raises the target function to an increasing power as it iterates, so the distribution focuses on the peak. See Simulated Annealing at Wolfram MathWorld.)

Random walk Metropolis (RWM, MRW)

Propose an *increment*, z, from the current location, *not dependent on the current location*, so $y = x + z$ with a specified PDF $K(z)$

This corresponds to

$$k(y|x) = K(y - x)$$

The proposals would give rise to a *random walk* if they were all accepted; the M-H rule modifies them to be a kind of directed random walk (steps in some directions are preferred)

Most commonly, a *symmetric* proposal is adopted:

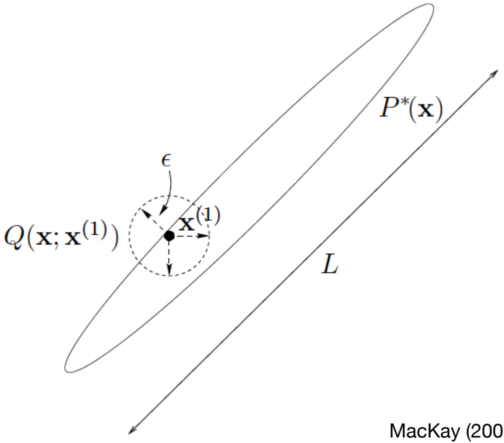
$$k(y|x) = K(|y - x|)$$

The acceptance probability simplifies:

$$\alpha(y|x) = \min\left[\frac{q(y)}{q(x)}, 1\right]$$

Key issues: shape and scale (in all directions) of $K(z)$

RWM in 2-D



MacKay (2003)

Small step size → good acceptance rate, but slow exploration

Independent Metropolis (IM)

Propose a new point independently of the current location:

$$k(y|x) = K(y)$$

The acceptance probability is now

$$\alpha(y|x) = \min\left[\frac{q(y)K(x)}{q(x)K(y)}, 1\right]$$

Note if $K(\cdot) \propto q(\cdot)$, proposals are from the target and are always accepted

Good acceptance requires $K(\cdot)$ to resemble the posterior → IM is typically only useful in low-D problems where we can construct a good $K(\cdot)$ (e.g., MVN at the mode)

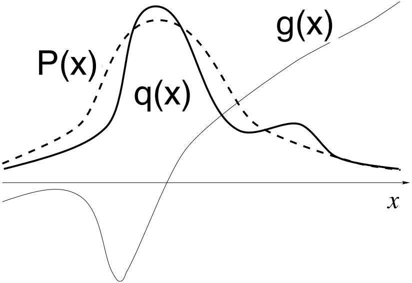
This looks like accept-reject, but K isn't an envelope function, and*repeating the current point* when we reject the proposed point makes it a Markov chain Monte Carlo algorithm (vs. IID Monte Carlo)

It more closely resembles *importance sampling* (an IID Monte Carlo method)

Importance sampling

$$\int \mathrm{d}\theta\; g(\theta)q(\theta) = \int \mathrm{d}\theta\; g(\theta)\frac{q(\theta)}{P(\theta)}P(\theta) \approx \frac{1}{n}\sum_{\theta_i \sim P(\theta)} g(\theta_i)\frac{q(\theta_i)}{P(\theta_i)}$$

Choose P to make variance small. (Not easy!)



Can be used for both model comparison (marginal likelihood calculation), and parameter estimation.

Adaptive importance sampling: Build the importance sampler on-the-fly (e.g., `VEGAS`, `miser` in *Numerical Recipes*); annealing adaptive importance sampling (TL$^+$ 2012; Liu 2014)...

Historical aside: Leonhard Euler

Swiss mathematician/physicist, 1707–1783

- Possibly most prolific author in any field; collected works $>$ 80 vol.
- Number theory, analysis, mathematical physics. . .
- Notation: $f(x)$, $\sin x$, $\cos x$, e, $\sum$
- $e^{\pi i} + 1 = 0$
- Used Fourier series, Bessel functions, Laplace transforms—*before F, B & L were born*

- Polymath: knew Virgil's *Aenid* by heart, well-versed in medicine, botany, geography
- "The Shakespeare of mathematics"
- Total blindness for last 17 yr, but his output *increased*

Euler on averaging

In the course of a study of the orbits of Jupiter and Saturn, requiring estimating 7 parameters using 75 observations:

> *"By the combination of two or more equations, the errors of the observations and of the calculations can multiply themselves."*
> — *Euler, 1749*

Tobias Mayer (German lunar astronomer), and later, Laplace, would advocate averaging, based on practical experience of seeing errors tending to cancel, & Bayesian reasoning