Topics in Retrospective simulation

Gareth Roberts

Department of Statistics, University of Warwick

FoCSML, 2025

Part 1: Introduction and Building Random Variables

Acknowledgements

A collection of modern topics in Stochastic Simulation and Computational Bayesian Statistics connected through simple ideas of retrospective simulation.

This is mostly joint collaborations with many people: Omiros Papaspiliopoulos, Jeff Rosenthal, Alex Beskos, Krys Latuszynski, Flavio, Goncalves, Dootika Vats, Murray Pollock, Christophe Andrieu, Paul Fearnhead, Kasper Berthelsen, Laird Breyer, Hongsheng Dai, Adam Johansen, Joris Bierkens, Sanket Agrawal

....

Motivations

Demands of modern computational statistics and machine learning problems.

Intractable likelihoods:

- Complex models
- Large data
- missing/latent data

Infinite-dimensional random variables:

- Simulating (and inference for) stochastic processes
- Non-parametric inference

Motivations

Demands of modern computational statistics and machine learning problems.

Intractable likelihoods:

- Complex models
- Large data
- missing/latent data

Infinite-dimensional random variables:

- Simulating (and inference for) stochastic processes
- Non-parametric inference

Aim to produce methods which are “exact” or lead to principled approximations.

Motivations

Demands of modern computational statistics and machine learning problems.

Intractable likelihoods:

- Complex models
- Large data
- missing/latent data

Infinite-dimensional random variables:

- Simulating (and inference for) stochastic processes
- Non-parametric inference

Aim to produce methods which are "exact" or lead to principled approximations.

Focus is mostly (not not exclusively) Bayesian.
Stochastic simulation tools will be key.

Plan for lectures (to be adapted .....)

1. Introduction to retrospective sampling
   Barker and Portkey Barker MCMC
2. Exact simulation for diffusions
   Importance sampling , Rao-Blackwellisation, the CIS algorithm
3. Some topics in perfect simulation
4. Subsampling in Monte Carlo:
   PDMPs
   The SCALE algorithm
5. Fusion algorithms
   Football draws

Warning, Part 3 will probably be omitted but I will give you the slides anyway in case you are interested.

Basic simulation toolbox 1: Simulating discrete events

We want to simulate from an event with probability p.

Basic algorithm: simulate from a $U \sim U(0,1)$ random variable. Then if

$$I = \mathbf{1}(U \leq p) \ ,$$

then I is the indicator of an event of probability p.

Basic simulation toolbox 1: Simulating discrete events

We want to simulate from an event with probability p.

Basic algorithm: simulate from a $U \sim U(0,1)$ random variable. Then if

$$I = \mathbf{1}(U \le p) \; ,$$

then I is the indicator of an event of probability p.

But do we really need to KNOW p to carry this out?

Basic simulation toolbox 2: Rejection sampling

Interested in sampling from a distribution $\mathbf{P}$ on a state space $\mathfrak{X}$:

$$\frac{d\mathbf{P}}{d\mu}(x) \propto f(x)$$

$\mathfrak{X}$ could be simple, complex, high-dimensional, infinite dimensional (eg the trajectory of a stochastic process) ....

Propose from $\mathbf{Q}$ instead such that $\mathbf{P} << \mathbf{Q}$ and

$$\frac{d\mathbf{P}}{d\mathbf{Q}}(x) \leq K, \qquad \mathbf{Q} \text{ almost surely.}$$

Accept draw, x, with probability proportional to

$$\frac{1}{K}\frac{d\mathbf{P}}{d\mathbf{Q}}(x)$$

Accepted draws are from $\mathbf{P}$.

Basic simulation toolbox 2: Rejection sampling

Interested in sampling from a distribution **P** on a state space $\mathfrak{X}$:

$$\frac{d\mathbf{P}}{d\mu}(x) \propto f(x)$$

$\mathfrak{X}$ could be simple, complex, high-dimensional, infinite dimensional (eg the trajectory of a stochastic process) ....

Propose from **Q** instead such that $\mathbf{P} << \mathbf{Q}$ and

$$\frac{d\mathbf{P}}{d\mathbf{Q}}(x) \le K, \qquad \mathbf{Q} \text{ almost surely.}$$

Accept draw, x, with probability proportional to

$$\frac{1}{K}\frac{d\mathbf{P}}{d\mathbf{Q}}(x)$$

Accepted draws are from **P**.

Do we need to know K?

Do we need to be able to calculate f pointwise?

Do we even need to simulate from (the whole of) **Q**?

Basic simulation toolbox 3: Importance Sampling

Interested in estimating $\mathbf{E_P}(h(X))$ by taking draws from $\mathbf{Q}$ instead: $X_1, X_2, \ldots X_n$ and use

$$E = \frac{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i) h(X_i)}{n}$$

Only requires

$$d\mathbf{P} << d\mathbf{Q} \ . \tag{1}$$

Then estimator is unbiased and consistent.

So in principle can be used more generally than rejection sampling. (There's also a ratio version of this which is consistent and does not require normalising constants.)

If we don't know K or normalisation constant for f, then can use instead

$$E' = \frac{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i) h(X_i)}{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i)}$$

which is not unbiased but is consistent.

Basic simulation toolbox 3: Importance Sampling

Interested in estimating $\mathbf{E_P}(h(X))$ by taking draws from $\mathbf{Q}$ instead: $X_1, X_2, \ldots X_n$ and use

$$E = \frac{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i) h(X_i)}{n}$$

Only requires

$$d\mathbf{P} << d\mathbf{Q}\ . \qquad (1)$$

Then estimator is unbiased and consistent.

So in principle can be used more generally than rejection sampling. (There's also a ratio version of this which is consistent and does not require normalising constants.)

If we don't know K or normalisation constant for f, then can use instead

$$E' = \frac{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i) h(X_i)}{\sum_{i=1}^{n} \frac{d\mathbf{P}}{d\mathbf{Q}}(X_i)}$$

which is not unbiased but is consistent.

But can we get away without (1)?

Basic simulation toolbox 4: Markov chain Monte Carlo

Metropolis-Hastings algorithm: constructs a reversible Markov chain with invariant density π.

Given x_n, propose y_{n+1} from a Markov chain with kernel density $q(x, y)$ accepting this proposal with probability $\alpha(x_n, y_{n+1})$ where

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

Acceptance implies that $x_{n+1} = y_{n+1}$ while rejection leads to $x_{n+1} = x_n$.

Basic simulation toolbox 4: Markov chain Monte Carlo

Metropolis-Hastings algorithm: constructs a reversible Markov chain with invariant density π.

Given x_n, propose y_{n+1} from a Markov chain with kernel density $q(x, y)$ accepting this proposal with probability $\alpha(x_n, y_{n+1})$ where

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

Acceptance implies that $x_{n+1} = y_{n+1}$ while rejection leads to $x_{n+1} = x_n$.

Do we need to be able to calculate $\pi(y)$ pointwise at each iteration?

What is retrospective simulation?

It is an attempt to take advantage of the redundancy inherent in modern simulation algorithms (particularly MCMC, rejection sampling) by subverting the traditional order of algorithm steps.

It is (in principle) very simple!

Retrospective simulation is most powerful in infinite dimensional contexts, where its natural competitors are approximate and computationally expensive. In contrast, restrospective methods are often computationally inexpensive and "exact".

Restrospective sampling has natural allies in the simulation game, for example catalytic perfect simulation, and non-centering

Ex 1: The birth of retrospective simulation?

Consider the quiz question on a Children's television programme (set in 1975!):

Who is the *current* manager of Liverpool football club?

1. Bill Shankly
2. Bob Paisley
3. Harold Wilson

N people enter a competition to win a prize, entering their answer on a postcard. The winner is drawn uniformly from those who get the question right (ie most of them). Suppose a proportion $p > 0.5$ get it right.

Algorithm 1

1. Mark each of the N entries, placing the correct postcards into a bucket.
2. Shake the bucket and then pick out one postcard, declaring its author the winner.

Cost of this procedure, $O(N)$.

Algorithm 2

1. Throw all the postcards into the bucket without marking them
2. Draw postcards until a winner is found

Cost of this procedure, $O(p^{-1})$.

Ex 2: Rejection sampling

Let f be a density of interest, and g be a density from which we can simulate. f/g bounded by K say.

1. Sample X from g.
2. Compute $p(X) = f(X)/(Kg(X))$.
3. Simulate $U \sim U(0,1)$.
4. Accept X if $p(X) > U$. Otherwise return to 1.

Blue steps are often unnecessary!

Retrospective rejection sampling

We want to carry out the simulation on a space $\mathfrak{X}$ which is typically complex, high- or infinite-dimensional.

Firstly, identify a function $h : [0,1] \times \mathfrak{X} \rightarrow \mathfrak{Y}$ where $\mathfrak{Y}$ is a much simpler space.

We want h to act as a random projection in the sense that for each $v \in [0,1]$ we can identify a set $A(v) \subset \mathfrak{Y}$ such that if $V \sim U(0,1)$ then

$$\mathbf{P}_V\{h(V,X) \in A(V)\} = p(X)$$

Then we can construct a retrospective rejection sampling algorithm which operates in $\mathfrak{Y}$ instead.

Retrospective rejection sampling algorithm

1. Sample $V \sim U(0,1)$.
2. Simulate $h(X, V)$.
3. If $h(X, V) \in A(V)$ the accept. Otherwise return to 1.
4. Fill in missing bits of X from distribution of $X|h(X, V)$ as required.

Retrospective rejection sampling algorithm

1. Sample $V \sim U(0,1)$.
2. Simulate $h(X, V)$.
3. If $h(X, V) \in A(V)$ the accept. Otherwise return to 1.
4. Fill in missing bits of X from distribution of $X|h(X, V)$ as required.

Does this mathematical abstraction serve any purpose?

Can it ever be useful?

Won't step 4 often be impossible?

At Warwick, in the last 15-20 years, we have used these techniques for many Bayesian computational problems, particularly retrospective rejection sampling:

- intractable likelihood problems
- exact simulation of diffusions, jump diffusions, others ...
- Bayesian inference for stochastic processes
- Infinite dimensional MCMC, eg for Bayesian nonparametric models

.... thanks to many collaborators, especially Omiros Papaspiliopoulos and Alex Beskos.
Some of these methods will be presented in the course.

Ex3: Simulating from intractable probabilities

Even simpler example: example from undergraduate simulation class 101:

Suppose we wish to simulate from an event I of probability p.

If p is known, then set

$$I = \mathbf{1}[U \leq p]$$

for a uniform $[0,1]$ random variable U.

Then I has probability p.

But what if p is unknown?

Retrospective discrete simulation

Suppose we can generate a random variable P which is an unbiased estimator of p with $0 \leq P \leq 1, \quad a.s.$, ie $\mathbf{E}(P) = p$

Now set

$$\hat{I} = \mathbf{1}[U \leq P]$$

Then $\hat{I}$ also has probability p.

Retrospective discrete simulation

Suppose we can generate a random variable P which is an unbiased estimator of p with $0 \leq P \leq 1, \quad a.s.$, ie $\mathbf{E}(P) = p$

Now set

$$\hat{I} = \mathbf{1}[U \leq P]$$

Then $\hat{I}$ also has probability p.

Amazing that sometime simulation of a random variable is often EASIER than evaluating a probability!

Slight generalisation

Probabilities $p_1, \dots p_k$ with $\sum_i p_i = 1$.

Random variables $R_1, \dots R_k$ such that there exists a constant M with

- $\mathbf{E}(R_i) = Mp_i, \quad \forall i$
- $\sum_i R_i = M, \quad a.s.$

Slight generalisation

Probabilities $p_1, \dots p_k$ with $\sum_i p_i = 1$.

Random variables $R_1, \dots R_k$ such that there exists a constant M with

- $\mathbf{E}(R_i) = Mp_i, \quad \forall i$
- $\sum_i R_i = M, \quad a.s.$

Then define X to be equal to i if

$$\sum_{j=1}^{i-1} R_j < UM \leq \sum_{j=1}^{i} R_j$$

for $U \sim U(0,1)$, then X has probabilities $\{p_i\}$.

Random discrete rational simulation

If the R_is are guaranteed to be integer-valued, we can do this by a ball draw:

Put R_i balls of label i into a bucket and draw one out at random.

Very simple, but does it have uses .... ?

Random discrete rational simulation

If the R_is are guaranteed to be integer-valued, we can do this by a ball draw:

Put R_i balls of label i into a bucket and draw one out at random.

Very simple, but does it have uses .... ?

We shall see an important example in the last part of the course.

Ex. 3: The alternating series method

Devroye (1986)
Let $p = a_0 - a_1 + a_2 - a_3 + a_4 - \ldots$, where $\{a_i\}$ is a decreasing sequence. To simulate an event of probability p, the retrospective method is as follows.
Use partial sums as upper and lower bounds for p:

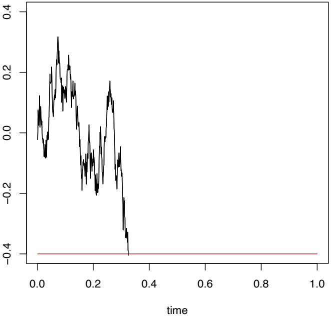
$$p_i^+ = \sum_{j=0}^{2i} a_j (-1)^j;$$

$$p_i^- = \sum_{j=0}^{2i-1} a_j (-1)^j;$$

1. Simulate $U \sim U(0,1)$.
2. Find i with both p_i^+ and p_i^- are either above or below U
3. When values are less than U, event is true, otherwise false.

Example: Simulation of BM hitting times

Let B_t be standard Brownian motion. Let $\tau_a = \inf\{t;\ B_t = a\}$

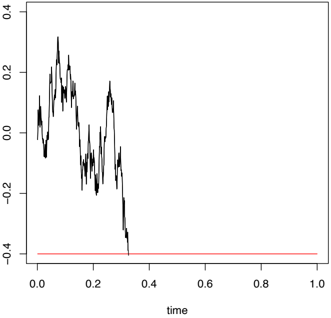


The distribution of τ_a is readily available analytically:

$$\mathbf{P}(\tau_a > t) = 2\Phi\left(\frac{-|a|}{t^{1/2}}\right)$$

Example: Simulation of BM hitting times

Let B_t be standard Brownian motion. Let $\tau_a = \inf\{t;\ B_t = a\}$



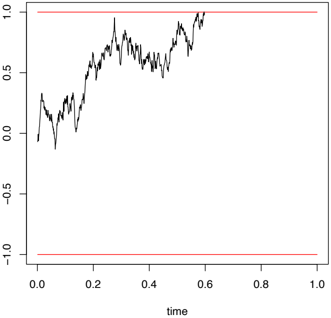
The distribution of τ_a is readily available analytically:

$$\mathbf{P}(\tau_a > t) = 2\Phi\left(\frac{-|a|}{t^{1/2}}\right)$$

However: consider two-sided hitting time,

$\tau_{a,-b} = \inf\{t;\ B_t = a \text{ or } -b\}$. Harder.

Two-sided boundaries



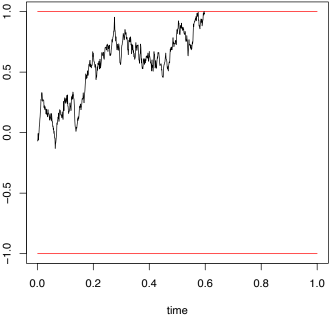
However by the reflection principle there exists an expansion

$$\mathbf{P}(\tau_{a,-b} \le t) = a_0 - a_1 + a_2 \dots$$

Use Union Intersection formula.

$a_0 = \mathbf{P}(U) + \mathbf{P}(L), \quad a_1 = \mathbf{P}(UL) + P(LU)$ etc

Two-sided boundaries



However by the reflection principle there exists an expansion

$$\mathbf{P}(\tau_{a,-b} \leq t) = a_0 - a_1 + a_2 \ldots$$

Use Union Intersection formula.

$a_0 = \mathbf{P}(U) + \mathbf{P}(L), \quad a_1 = \mathbf{P}(UL) + P(LU)$ etc

So we can apply the alternating series method.

Ex 4: Simulating from unnormalised probabilities

We have $p_1, p_2, \ldots$ is a sequence of positive numbers with $p_i \leq q_i$ and $\sum_{i=j+1}^{\infty} q_i = G(j) < \infty$.
We would like to simulate from the discrete distribution with probabilities proportional to $\{p_i\}$.

Think Bayesian analysis with qs coming from prior.

Why not use the inverse CDF method?

1. Calculate $s = \sum_{i=1}^{\infty} p_i$
2. Simulate $U \sim U(0, 1)$.
3. Set $X = \inf\{j;\ \sum_{i=1}^{j} p_j / s \geq U\}$.

We don't know s but we have upper and lower bounds from assumptions.

Retrospective inverse CDF method

$$s_j^- = \sum_{i=1}^{j} p_i$$

$$s_j^+ = \sum_{i=1}^{j} p_i + G(j)$$

Clearly

$$s_j^- \le s_{j+1}^- \le s \le s_{j+1}^+ \le s_j^+$$

$$P_i^{+j} = \sum_{k=1}^{j} \frac{p_k}{s_j^-}$$ upper bound on cumulative probability

$$P_i^{-j} = \sum_{k=1}^{j} \frac{p_k}{s_j^+}$$ lower bound on cumulative probability

Retrospective inverse CDF method

$$s_j^- = \sum_{i=1}^{j} p_i$$

$$s_j^+ = \sum_{i=1}^{j} p_i + G(j)$$

Clearly

$$s_j^- \leq s_{j+1}^- \leq s \leq s_{j+1}^+ \leq s_j^+$$

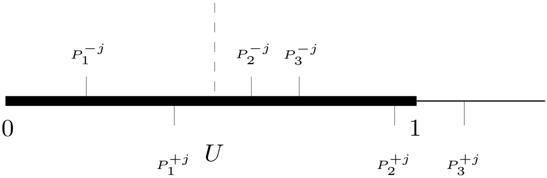
$$P_i^{+j} = \sum_{k=1}^{j} \frac{p_k}{s_j^-}$$ upper bound on cumulative probability

$$P_i^{-j} = \sum_{k=1}^{j} \frac{p_k}{s_j^+}$$ lower bound on cumulative probability

$$X^{+j}(U) = \inf\{j;\ P_i^{+j} \geq U\}$$ lower bound on X

$$X^{-j}(U) = \inf\{j;\ P_i^{-j} \geq U\}.$$ lower bound on X

1. Simulate $U \sim U(0,1)$.
2. Calculate $X^{-j}(U)$ and $X^{+j}(U)$, $j = 1, 2, \ldots$ until $X^{-j}(U) = X^{+j}(U)$. Set X to be this common value.



Here $X^{+j} = X^{-j} = X = 2$.

Ex. 5: Retrospective MCMC

Many opportunities.

Peeking forward at future observations ....

Eg simulate from $\pi(\theta, X)$ with θ 'simple' and X 'complex'.

Consider Gibbs sampler which alternates between updating $\theta|X$ and $X|\theta$. The latter step is harder than the former.

However by suitable construction of random map $X \mapsto \theta$ (eg by catalytic field coupler, Breyer + R, 2001) we can often avoid having to calculate 'all' of X.

See example(s) in Part 3

Ex. 6: Coupling from the past

Propp and Wilson (1996).

Here the naive sampler starts at time $-\infty$ from all possible states. It then records the chain value at time 0.

CFTP starts at time 0 and proceeds backwards till the chain value at time 0 is inevitable.

See example(s) in Part 3

Ex. 7: Poisson thinning

Important for Parts 2 and 4.

How to simulate from a Poisson process of rate $\lambda(s)$, $0 \le s \le 1$ say?

Suppose $\lambda(s) \le \Lambda$ for all $0 \le s \le 1$.

Simulate a PP of rate Λ on $[0,1]$ giving points $Y_i, 1 \le i \le N$.

Then accept each point with probability

$$p_i = \frac{\lambda(Y_i)}{\Lambda}$$

The accepted points are a Poisson process of rate $\lambda(s)$.

Ex. 7: Poisson thinning

Important for Parts 2 and 4.

How to simulate from a Poisson process of rate $\lambda(s)$, $0 \leq s \leq 1$ say?

Suppose $\lambda(s) \leq \Lambda$ for all $0 \leq s \leq 1$.

Simulate a PP of rate Λ on $[0, 1]$ giving points $Y_i, 1 \leq i \leq N$.

Then accept each point with probability

$$p_i = \frac{\lambda(Y_i)}{\Lambda}$$

The accepted points are a Poisson process of rate $\lambda(s)$.

Idea readily generalises and localises.

Bernoulli factories and Barker MCMC

The general *Bernoulli Factory* problem originated from a classical probability question posed by Keane and O'Brien (1994). (Some earlier papers had asked related questions.)

Given coins which can generate events of probability p, how can we construct an algorithm to simulate from events of probability $h(p)$.

Originally studied in the case $f(p) = 2p$, but we shall be interested in more general Bernoulli factories.

The work presented here is recent and can be found in the following papers:

Vats, Goncalves, Latuszynski, and R (2022, Biometrika)

Agrawal, Vats, Latuszynski and R (2023, Advances in Applied Probability)

Intractable targets

Consider a Bayesian model for parameter θ:

$$\underbrace{\pi(\theta|y)}_{\text{Posterior}} \propto \underbrace{f(y|\theta)}_{\text{Likelihood}} \underbrace{\pi(\theta)}_{\text{Prior}} := \tilde{\pi}(\theta|y)\,.$$

The posterior is often complicated enough that it is only known up to the unnormalized $\tilde{\pi}(\theta|y)$.

Markov chain Monte Carlo (MCMC) algorithms may be used to sample from $\pi(\theta|y)$.

Accept-Reject based MCMC

An accept-reject MCMC algorithm $(k+1)$ update:

1. Generate $\theta^* \sim q(\theta^*|\theta_k)$
2. Set $\theta_{k+1} = \theta^*$ with probability $\alpha(\theta_k, \theta^*)$.
3. Else, $\theta_{k+1} = \theta_k$.

Of course $\alpha(\theta, \theta^*)$ needs to chosen to satisfy invariance (more on this later).

Accept-Reject based MCMC

An accept-reject MCMC algorithm $(k+1)$ update:

1. Generate $\theta^* \sim q(\theta^*|\theta_k)$
2. Set $\theta_{k+1} = \theta^*$ with probability $\alpha(\theta_k, \theta^*)$.
3. Else, $\theta_{k+1} = \theta_k$.

Of course $\alpha(\theta, \theta^*)$ needs to chosen to satisfy invariance (more on this later).

If $\alpha(\theta, \theta^*)$ can be evaluated, then obtaining an event with prob. $\alpha(\theta, \theta^*)$ is by:

$$\text{Get } U \sim U(0,1) \text{ and check is } U \leq \alpha(\theta, \theta^*)$$

Metropolis-Hastings (MH)

A popular acceptance probability used is the Metropolis-Hastings acceptance probability:

$$\alpha_{MH}(\theta, \theta^*) = \min\left\{1, \frac{\pi(\theta^*|y)\, q(\theta|\theta^*)}{\pi(\theta|y)\, q(\theta^*|\theta)}\right\} = \min\left\{1, \frac{\tilde{\pi}(\theta^*|y)\, q(\theta|\theta^*)}{\tilde{\pi}(\theta|y)\, q(\theta^*|\theta)}\right\}$$

Metropolis-Hastings (MH)

A popular acceptance probability used is the Metropolis-Hastings acceptance probability:

$$\alpha_{MH}(\theta, \theta^*) = \min\left\{1, \frac{\pi(\theta^*|y)\, q(\theta|\theta^*)}{\pi(\theta|y)\, q(\theta^*|\theta)}\right\} = \min\left\{1, \frac{\tilde{\pi}(\theta^*|y)\, q(\theta|\theta^*)}{\tilde{\pi}(\theta|y)\, q(\theta^*|\theta)}\right\}$$

Of course, if $\tilde{\pi}(\theta|y)$ is known, then MH can be implemented easily.

Intractable posteriors

Consider problems that yield targets that *cannot* be evaluated. This may be for example, because

$$\pi(\theta|y) = \int_{\eta} g(\theta, \eta|y) d\eta \,.$$

This problem arises in

- Priors on constrained spaces
- Missing data - imputation
- Bayesian inference for diffusions

Intractable posteriors

Consider problems that yield targets that *cannot* be evaluated. This may be for example, because

$$\pi(\theta|y) = \int_{\eta} g(\theta, \eta|y) d\eta \, .$$

This problem arises in

- Priors on constrained spaces
- Missing data - imputation
- Bayesian inference for diffusions

Here,

$$\alpha_{MH}(\theta, \theta^*) = \min \left\{ 1, \frac{\pi(\theta^*|y)\, q(\theta_k|\theta^*)}{\pi(\theta|y)\, q(\theta^*|\theta_k)} \right\}$$

cannot be evaluated.

Barker's algorithm

Barker (1965) proposed the acceptance function:

$$\alpha_B(\theta, \theta^*) = \frac{\pi(\theta^*|y)\, q(\theta|\theta^*)}{\pi(\theta|y)\, q(\theta^*|\theta) + \pi(\theta^*|y)\, q(\theta|\theta^*)}$$

Barker's algorithm is not very popular due to Peskun's ordering result.

Peskun Ordering (Peskun (1973))

Let $\bar{X}_h = n^{-1}\sum_t h(X_t)$ be a Monte Carlo estimator for a function h. Let P_B and P_{MH} be Barker's and MH Markov kernels. Then

Peskun Ordering (Peskun (1973))

Let $\bar{X}_h = n^{-1} \sum_t h(X_t)$ be a Monte Carlo estimator for a function h. Let P_B and P_{MH} be Barker's and MH Markov kernels. Then

$$\mathrm{var}_\pi(P_{MH}, h) \leq \mathrm{var}_\pi(P_B, h) \leq 2\,\mathrm{var}_\pi(P_{MH}, h) + \mathrm{Var}_\pi(h)$$

where $\mathrm{var}_\pi(P, h) = \lim_{n\to\infty} n\mathrm{Var}(\bar{X}_h)$ is the asymptotic variance when X_t is produced from P.

So although Barker's is more inefficient, it is not too much so.

Barker's for intractable posteriors

But Barker's still doesn't solve our problem since $\pi(\theta|y)$ still appears in the function:

$$\alpha_B(\theta, \theta^*) = \frac{\pi(\theta^*|y)\, q(\theta|\theta^*)}{\pi(\theta|y)\, q(\theta^*|\theta) + \pi(\theta^*|y)\, q(\theta|\theta^*)}$$

Barker's for intractable posteriors

But Barker's still doesn't solve our problem since $\pi(\theta|y)$ still appears in the function:

$$\alpha_B(\theta,\theta^*) = \frac{\pi(\theta^*|y)\,q(\theta|\theta^*)}{\pi(\theta|y)\,q(\theta^*|\theta) + \pi(\theta^*|y)\,q(\theta|\theta^*)}$$

To the rescue: Bernoulli factory!

Bernoulli factory

A Bernoulli factory is an algorithm that generates a Bernoulli event of probability $h(p)$ for some function $h(\cdot)$, using Bernoulli(p) events.

Bernoulli factory

A Bernoulli factory is an algorithm that generates a Bernoulli event of probability $h(p)$ for some function $h(\cdot)$, using Bernoulli(p) events.

Want a Bernoulli factory to get events of prob. $\alpha_{\mathsf{B}}(\theta, \theta^*)$ without evaluating it.

Bernoulli factory

A Bernoulli factory is an algorithm that generates a Bernoulli event of probability $h(p)$ for some function $h(\cdot)$, using Bernoulli(p) events.

Want a Bernoulli factory to get events of prob. $\alpha_{\mathrm{B}}(\theta, \theta^*)$ without evaluating it.

Goncalves, Latuszynski and R (2017) proposed the following.

Suppose we can, find c_θ

$$\pi(\theta|y)q(\theta^*|\theta) \leq c_\theta\,.$$

Then set $\pi(\theta|y)q(\theta^*|\theta) = c_\theta p_\theta$ where

$$p_\theta = \pi(\theta|y)q(\theta^*|\theta)/c_\theta \leq 1,$$

Bernoulli factory

A Bernoulli factory is an algorithm that generates a Bernoulli event of probability $h(p)$ for some function $h(\cdot)$, using Bernoulli(p) events.

Want a Bernoulli factory to get events of prob. $\alpha_{\mathrm{B}}(\theta, \theta^*)$ without evaluating it.

Goncalves, Latuszynski and R (2017) proposed the following.

Suppose we can, find c_θ

$$\pi(\theta|y)q(\theta^*|\theta) \leq c_\theta\,.$$

Then set $\pi(\theta|y)q(\theta^*|\theta) = c_\theta p_\theta$ where

$$p_\theta = \pi(\theta|y)q(\theta^*|\theta)/c_\theta \leq 1,$$

Then to generate events with probability

$$\frac{\pi(\theta^*|y)q(\theta^*|\theta)}{\pi(\theta|y)q(\theta|\theta^*) + \pi(\theta^*|y)q(\theta^*|\theta)} = \frac{c_{\theta^*} p_{\theta^*}}{c_\theta p_\theta + c_{\theta^*} p_{\theta^*}}$$

we propose a *two-coin* algorithm.

Two-coin algorithm

```
1. Draw $C_1 \sim \text{Bern}\left(\frac{c_{\theta^*}}{c_\theta + c_{\theta^*}}\right)$
2. If $C_1 = 1$, then
    2.1 Draw $C_2 \sim \text{Bern}(p_{\theta^*})$
    2.2 If $C_2 = 1$, then output 1
    2.3 If $C_2 = 0$, then goto Step 1
3. If $C_1 = 0$, then
    3.1 Draw $C_2 \sim \text{Bern}(p_\theta)$
    3.2 If $C_2 = 1$, then output 0
    3.3 If $C_2 = 0$, then go to Step 1
```

The above algorithm outputs 1 w.p. $\alpha_B(\theta, \theta^*)$.

Two-coin algorithm

```
1. Draw $C_1 \sim \text{Bern}\left(\frac{c_{\theta^*}}{c_\theta + c_{\theta^*}}\right)$
2. If $C_1 = 1$, then
    2.1 Draw $C_2 \sim \text{Bern}(p_{\theta^*})$
    2.2 If $C_2 = 1$, then output 1
    2.3 If $C_2 = 0$, then goto Step 1
3. If $C_1 = 0$, then
    3.1 Draw $C_2 \sim \text{Bern}(p_\theta)$
    3.2 If $C_2 = 1$, then output 0
    3.3 If $C_2 = 0$, then go to Step 1
```

The above algorithm outputs 1 w.p. $\alpha_B(\theta, \theta^*)$. But how do we sample Bern(p_θ)?

Two-coin algorithm by retrospective sampling

To sample Bern(p_θ), we use retrospective simulation. Note that

$$p_\theta = \frac{\pi(\theta|y)q(\theta^*|\theta)}{c_\theta} = \frac{q(\theta^*|\theta)\int g(\theta,\eta|y)d\eta}{c_\theta}$$
$$= \frac{q(\theta^*|\theta)}{c_\theta}\int \frac{g(\theta,\eta|y)}{h(\eta)}h(\eta)d\eta$$

for some density h. Then draw $Z \sim h$

$$P_\theta = \frac{q(\theta^*|\theta)g(\theta,Z|y)}{c_{\theta h(Z)}} \quad \text{and } \mathrm{E}\left(P_\theta\right) = p_\theta\,.$$

(careful - need h so that $P_\theta < 1$.)
So if $C_2 \sim$ Bern(M_θ), then

$$\mathbf{P}(C_2 = 1) = \mathrm{E}\left(I(C_2 = 1)\right) = \mathrm{E}\left(\mathrm{E}\left(I(C_2 = 1)|M_\theta\right)\right) = p_\theta\,.$$

So $C_2 \sim$ Bern(p_θ)

Two-coin algorithm

```
1. Draw $C_1 \sim \text{Bern}\left(\frac{c_{\theta^*}}{c_\theta + c_{\theta^*}}\right)$
2. If $C_1 = 1$, then
    2.1 Draw $C_2 \sim \text{Bern}(p_{\theta^*})$
    2.2 If $C_2 = 1$, then output 1
    2.3 If $C_2 = 0$, then goto Step 1
3. If $C_1 = 0$, then
    3.1 Draw $C_2 \sim \text{Bern}(p_\theta)$
    3.2 If $C_2 = 1$, then output 0
    3.3 If $C_2 = 0$, then go to Step 1
```

Algorithm restarts often if p_θ is small and $c_{\theta^*} >> c_\theta$ or p_{θ^*} are small and $c_{\theta^*} << c_\theta$. That is, if we propose unlikely values in the Barker's algorithm, algorithm gets stuck in a loop.

Can choose to increase one of the cs to make them comparable in size, but then maybe both p_θ and p_{θ^*} could be small.

Two-coin algorithm

```
1. Draw $C_1 \sim \text{Bern}\left(\frac{c_{\theta^*}}{c_\theta + c_{\theta^*}}\right)$
2. If $C_1 = 1$, then
    2.1 Draw $C_2 \sim \text{Bern}(p_{\theta^*})$
    2.2 If $C_2 = 1$, then output 1
    2.3 If $C_2 = 0$, then goto Step 1
3. If $C_1 = 0$, then
    3.1 Draw $C_2 \sim \text{Bern}(p_\theta)$
    3.2 If $C_2 = 1$, then output 0
    3.3 If $C_2 = 0$, then go to Step 1
```

Algorithm restarts often if p_θ is small and $c_{\theta^*} >> c_\theta$ or p_{θ^*} are small and $c_{\theta^*} << c_\theta$. That is, if we propose unlikely values in the Barker's algorithm, algorithm gets stuck in a loop.

Can choose to increase one of the cs to make them comparable in size, but then maybe both p_θ and p_{θ^*} could be small.

Not robust!

Portkey Barker's algorithm

We need a way of getting out of trouble (exiting the loop) without violating stationarity of the resulting Markov chain.

(Name inspired by Harry Potter!)

Portkey Barker's algorithm

Here, the acceptance probability need not be the ratio of the target densities.

Portkey Barker's algorithm

Here, the acceptance probability need not be the ratio of the target densities. We propose the *Portkey Barker's* algorithm for $d(\theta, \theta^*) \geq 0$.

$$\alpha_P(\theta, \theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + d(\theta, \theta^*)}$$

Portkey Barker's algorithm

Here, the acceptance probability need not be the ratio of the target densities. We propose the *Portkey Barker's* algorithm for $d(\theta, \theta^*) \geq 0$.

$$\alpha_P(\theta, \theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + d(\theta, \theta^*)}$$

Theorem

If $d(\theta, \theta^*) = d(\theta^*, \theta)$, then $\alpha_P(\theta, \theta^*)$ yields a π-invariant Markov chain.

Portkey Barker's algorithm

Here, the acceptance probability need not be the ratio of the target densities. We propose the *Portkey Barker's* algorithm for $d(\theta, \theta^*) \geq 0$.

$$\alpha_P(\theta, \theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + d(\theta, \theta^*)}$$

Theorem

If $d(\theta, \theta^*) = d(\theta^*, \theta)$, then $\alpha_P(\theta, \theta^*)$ yields a π-invariant Markov chain.

We consider, for $\beta > 0$,

$$\alpha_\beta(\theta, \theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + \frac{1-\beta}{\beta}(c_{\theta^*} + c_\theta)}$$

Portkey Barker's algorithm

$$\alpha_\beta(\theta,\theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + \frac{1-\beta}{\beta}(c_{\theta^*} + c_\theta)}$$

Ideally, choose $\beta \approx 1$ so as to remain close to the Barker's algorithm. Because:

Portkey Barker's algorithm

$$\alpha_\beta(\theta, \theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + \frac{1-\beta}{\beta}(c_{\theta^*} + c_\theta)}$$

Ideally, choose $\beta \approx 1$ so as to remain close to the Barker's algorithm. Because:

Theorem

For $\beta > 0$,

$$var_\pi(h, P_B) \leq \beta\, var_\pi(h, P_\beta) + (\beta - 1) Var_\pi(h)\,.$$

Portkey Barker's algorithm

$$\alpha_\beta(\theta,\theta^*) = \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta) + \pi(\theta^*|y)q(\theta|\theta^*) + \frac{1-\beta}{\beta}(c_{\theta^*} + c_\theta)}$$

Ideally, choose $\beta \approx 1$ so as to remain close to the Barker's algorithm. Because:

Theorem

For $\beta > 0$,

$$var_\pi(h, P_B) \le \beta\, var_\pi(h, P_\beta) + (\beta - 1) Var_\pi(h)\,.$$

and if there exists $\gamma > 0$ such that $p_{\theta^*} > \gamma$ and $p_\theta > \gamma$, then

$$var_\pi(h, P_\beta) \le \left(1 + \frac{1-\beta}{\gamma\beta}\right) var_\pi(h, P_B) + \frac{1-\beta}{\gamma\beta} Var_\pi(h)\,.$$

Then why use Portkey Barker's?

Portkey Two-coin algorithm

```
1. Draw $S \sim \text{Bern}(\beta)$                    ($S$ is the portkey)
2. If $S = 0$, output 0.
3. If $S = 1$,
    3.1 Draw $C_1 \sim \text{Bern}\left(\frac{c_{\theta^*}}{c_\theta + c_{\theta^*}}\right)$
    3.2 If $C_1 = 1$, then
        3.2.1 Draw $C_2 \sim \text{Bern}(p_{\theta^*})$
        3.2.2 If $C_2 = 1$, then output 1
        3.2.3 If $C_2 = 0$, then goto Step 1
    3.3 If $C_1 = 0$, then
        3.3.1 Draw $C_2 \sim \text{Bern}(p_\theta)$
        3.3.2 If $C_2 = 1$, then output 0
        3.3.3 If $C_2 = 0$, then go to Step 1
```

Flipped Portkey's

Notice that if we divide Portkey Barker's throughout by

$$\pi(\theta^*|y)q(\theta|\theta^*)\pi(\theta|y)q(\theta^*|\theta)$$

then,

$$\begin{aligned}\alpha_P(\theta,\theta^*) &= \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta)+\pi(\theta^*|y)q(\theta|\theta^*)+d(\theta,\theta^*)} \\ &= \frac{(\pi(\theta|y)q(\theta^*|\theta))^{-1}}{(\pi(\theta|y)q(\theta^*|\theta))^{-1}+(\pi(\theta^*|y)q(\theta|\theta^*))^{-1}+d'(\theta,\theta^*)}\end{aligned}$$

Flipped Portkey's

Notice that if we divide Portkey Barker's throughout by

$$\pi(\theta^*|y)q(\theta|\theta^*)\pi(\theta|y)q(\theta^*|\theta)$$

then,

$$\begin{aligned}\alpha_P(\theta,\theta^*) &= \frac{\pi(\theta^*|y)q(\theta|\theta^*)}{\pi(\theta|y)q(\theta^*|\theta)+\pi(\theta^*|y)q(\theta|\theta^*)+d(\theta,\theta^*)} \\ &= \frac{(\pi(\theta|y)q(\theta^*|\theta))^{-1}}{(\pi(\theta|y)q(\theta^*|\theta))^{-1}+(\pi(\theta^*|y)q(\theta|\theta^*))^{-1}+d'(\theta,\theta^*)}\end{aligned}$$

So if we can *lower bound* $\pi(\theta|y)q(\theta^*|\theta)$, we can implement a similar Portkey two-coin algorithm.

Application: Bayesian Correlation Estimation

Suppose

$$y_1, \ldots, y_n | R \overset{iid}{\sim} N(0, R)$$

where R is a $p \times p$ correlation matrix.

Application: Bayesian Correlation Estimation

Suppose

$$y_1, \ldots, y_n | R \overset{iid}{\sim} N(0, R)$$

where R is a $p \times p$ correlation matrix.

Let S_p^+ be the set of $p \times p$ correlation matrices. Liechty, Liechty and Muller (Bimetrika, 2009) set priors:

$$f(R \mid \mu, \sigma^2) = L(\mu, \sigma^2) \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}\{R \in S_p^+\}, \text{ s.t.}$$

$$L^{-1}(\mu, \sigma^2) = \int_{R \in S_p^+} \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} dr_{ij}$$

Further, $\mu \sim N(0, \tau^2)$ and $\sigma^2 \sim IG(a_0, b_0)$ are chosen. Interest is in the posterior distribution for (R, μ, σ^2).

MCMC steps

Let $l = p(p-1)/2$. To implement a component-wise algorithm:

$$f(r_{ij} \mid r_{-ij}, \mu, \sigma^2) \propto |R|^{-n/2} \exp\left\{-\frac{\mathrm{tr}(R^{-1}Y^TY)}{2} - \frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \mathbb{I}_{\{l_{ij} \leq r_{ij} \leq u_{ij}}$$

We use Metropolis-Hastings update with a Gaussian proposal for each r_{ij}.

MCMC steps

Let $l = p(p-1)/2$. To implement a component-wise algorithm:

$$f(r_{ij} \mid r_{-ij}, \mu, \sigma^2) \propto |R|^{-n/2} \exp\left\{ -\frac{\mathrm{tr}(R^{-1}Y^TY)}{2} - \frac{(r_{ij}-\mu)^2}{2\sigma^2} \right\} \mathbb{I}_{\{l_{ij} \le r_{ij} \le u_{ij}}$$

We use Metropolis-Hastings update with a Gaussian proposal for each r_{ij}.

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{ -\frac{(r_{ij}-\mu)^2}{2\sigma^2} \right\} \exp\left\{ -\frac{\mu^2}{2\tau^2} \right\},$$

$$f(\sigma^2 \mid R, \mu) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{ -\frac{(r_{ij}-\mu)^2}{2\sigma^2} \right\} \left(\frac{1}{\sigma^2} \right)^{a_0 + l/2 + 1} \exp\left\{ -\frac{b_0}{\sigma^2} \right\}$$

Running Metropolis steps for the conditional updates of μ and σ^2 is not possible.

Liechty Liechtly and Muller (2009) use an approximate inference shadow prior approach.

Application: Flipped Portkey Barker's

Let's focus on the μ update:

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\}$$

Recall,

$$L^{-1}(\mu, \sigma^2) = \int_{R \in S_p^+} \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} dr_{ij}$$

Application: Flipped Portkey Barker's

Let's focus on the μ update:

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\}$$

Recall,

$$L^{-1}(\mu, \sigma^2) = \int_{R \in S_p^+} \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} dr_{ij}$$

Obtaining an unbiased estimate of L^{-1} is simple and

Application: Flipped Portkey Barker's

Let's focus on the μ update:

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\}$$

Recall,

$$L^{-1}(\mu, \sigma^2) = \int_{R \in S_p^+} \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} dr_{ij}$$

Obtaining an unbiased estimate of L^{-1} is simple and

$$L^{-1}(\mu, \sigma^2) \leq \left[\Phi\left(\sigma^{-1}(1-\mu)\right) - \Phi\left(\sigma^{-1}(-1-\mu)\right)\right]^{p(p-1)/2} := \tilde{c}_\mu$$

which gives us a lower bound for $f(\mu \mid R, \sigma^2)$. So we can use flipped portkey Barker's algorithm.

Application: Flipped Portkey Barker's

Let's focus on the μ update:

$$f(\mu \mid R, \sigma^2) \propto L(\mu, \sigma^2) \prod_{i<j} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\}$$

Recall,

$$L^{-1}(\mu, \sigma^2) = \int_{R \in S_p^+} \prod_{i<j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(r_{ij}-\mu)^2}{2\sigma^2}\right\} dr_{ij}$$

Obtaining an unbiased estimate of L^{-1} is simple and

$$L^{-1}(\mu, \sigma^2) \leq \left[\Phi\left(\sigma^{-1}(1-\mu)\right) - \Phi\left(\sigma^{-1}(-1-\mu)\right)\right]^{p(p-1)/2} := \tilde{c}_\mu$$

which gives us a lower bound for $f(\mu \mid R, \sigma^2)$. So we can use flipped portkey Barker's algorithm.

We study the correlation of the closing prices of the four major European stocks from 1991-1998.

Number of loops

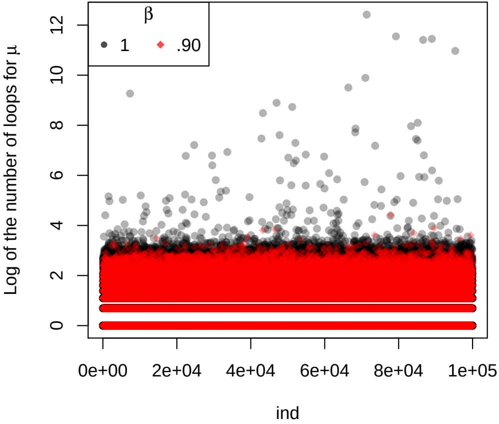


Figure: Log of the ratio of the Bernoulli factory loops for one run of length $1e5$.

Example: ACF plots

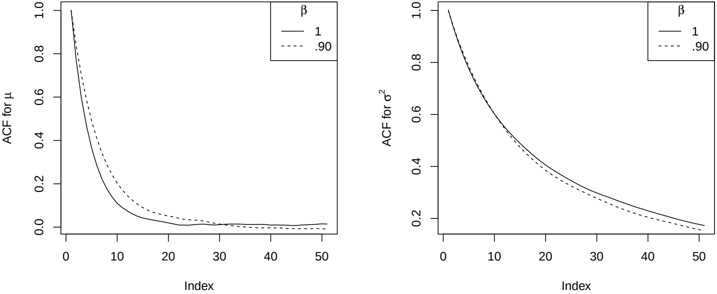


Figure: Autocorrelation plot for one run of length 1e5.

Example: Efficiency

Table: Averaged results from 10 replications of length $1e4$

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS$/s$	9.63* (-)	14.83 (0.279)
Mean loops μ	218.43 (148.89)	2.99 (0.010)
Mean loops σ^2	3.21 (0.02)	2.49 (0.010)
Max loops μ	2084195 (1491777)	34 (2.94)
Max loops σ^2	38 (1.13)	27 (1.51)

Example: Efficiency

Table: Averaged results from 10 replications of length 1e4

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS$/s$	9.63* (-)	14.83 (0.279)
Mean loops μ	218.43 (148.89)	2.99 (0.010)
Mean loops σ^2	3.21 (0.02)	2.49 (0.010)
Max loops μ	2084195 (1491777)	34 (2.94)
Max loops σ^2	38 (1.13)	27 (1.51)

Could only do 10 replications as $\beta = 1$ original would get stuck in large loops!

Example: Efficiency

Table: Averaged results from 10 replications of length 1e4

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS/s	9.63* (-)	14.83 (0.279)
Mean loops μ	218.43 (148.89)	2.99 (0.010)
Mean loops σ^2	3.21 (0.02)	2.49 (0.010)
Max loops μ	2084195 (1491777)	34 (2.94)
Max loops σ^2	38 (1.13)	27 (1.51)

Could only do 10 replications as $\beta = 1$ original would get stuck in large loops!

However to get 2-coin to even work this well, we had to use really small proposals (much smaller than "optimal" Barker) ....

Some conclusions

- Bernoulli factory MCMC methods can be an attractive to data augmentation in a wide range of bayesian inference problems.
- Barker has natural advantages due to the two-coin algorithm.
- But Barker is not computationally robust.
- Portkey Barker solves the problem at the cost of a very minor loss in Markov chain efficiency.
- Efficiency and scaling for Barker and Portkey Barker can use theory developed in Agrawal, S., Vats, D., Łatusyński, K., Roberts, G. O. (2023) .

Optimal Scaling Beyond Metropolis

Consider the setup of R Gelman and Gilks (1996)

Optimal Scaling Beyond Metropolis

Consider the setup of R Gelman and Gilks (1996)
Consider a sequence of target distributions $\{\boldsymbol{\pi}_d\}$

$$\pi_d(\boldsymbol{x}^d) = \prod_{i=1}^{d} f(x_i^d), \qquad \boldsymbol{x}^d = (x_1^d, \ldots, x_d^d)^T \in \mathbb{R}^d.$$

where f is positive and in C^2 (the class of all real-valued functions with continuous second order derivatives) Further, f'/f is Lipschitz and the following moment conditions hold,

$$\mathbb{E}_f\left[\left(\frac{f'(X)}{f(X)}\right)^8\right] < \infty, \qquad \mathbb{E}_f\left[\left(\frac{f''(X)}{f(X)}\right)^4\right] < \infty.$$

Denote:

$$J = \mathbb{E}_f\left[\left(\frac{f'(X)}{f(X)}\right)^2\right].$$

Optimal Scaling Beyond Metropolis

Consider the Gaussian proposal distributions $\{Q_d(\boldsymbol{x}^d, \cdot)\} = N(\boldsymbol{x}^d, \sigma_d^2)$ where for some constant $l \in \mathbb{R}^+$,

$$\sigma_d^2 = l^2/(d-1)\,.$$

Optimal Scaling Beyond Metropolis

Consider the Gaussian proposal distributions $\{Q_d(\boldsymbol{x}^d, \cdot)\} = N(\boldsymbol{x}^d, \sigma_d^2)$ where for some constant $l \in \mathbb{R}^+$,

$$\sigma_d^2 = l^2/(d-1)\,.$$

We accept draws from the proposal with probability α where α is such that there exists a *balancing* function, $g_\alpha : [0,\infty) \to [0,1]$, such that,

$$\textit{(Practicality)} \qquad \alpha(x,y) = g_\alpha\left(\frac{\pi(y)}{\pi(x)}\right), \quad x, y \in \mathcal{X},$$

$$\textit{(Invariance)} \qquad g_\alpha(z) = z g_\alpha\left(\frac{1}{z}\right), \quad 0 \le z < \infty,$$

$$\textit{(Weak)} \qquad g_\alpha(e^z), z \in \mathbb{R} \text{ is Lipschitz continuous.}$$

Optimal Scaling Beyond Metropolis

Consider the Gaussian proposal distributions $\{Q_d(\boldsymbol{x}^d, \cdot)\} = N(\boldsymbol{x}^d, \sigma_d^2)$ where for some constant $l \in \mathbb{R}^+$,

$$\sigma_d^2 = l^2/(d-1)\,.$$

We accept draws from the proposal with probability α where α is such that there exists a *balancing* function, $g_\alpha : [0, \infty) \to [0, 1]$, such that,

$$\textit{(Practicality)} \qquad \alpha(x, y) = g_\alpha\left(\frac{\pi(y)}{\pi(x)}\right), \quad x, y \in \mathcal{X},$$

$$\textit{(Invariance)} \qquad g_\alpha(z) = z g_\alpha\left(\frac{1}{z}\right), \quad 0 \le z < \infty,$$

$$\textit{(Weak)} \qquad g_\alpha(e^z), z \in \mathbb{R} \text{ is Lipschitz continuous.}$$

R Gelman and Gilks (1996) (and many others) prove weak convergence and obtain optimal scaling for $\alpha = \alpha_{MH}$. We extend their proof to this general class of acceptances.

Optimal Scaling Beyond Metropolis: Result

We speed up the process by a factor of d

$$\boldsymbol{X}^d_{[dt]} = (X^d_{[dt],1}, X^d_{[dt],2}, \dots, X^d_{[dt],d})^T; \quad t > 0.$$

Define a new 1-dimensional process $U^d_t = X^d_{[dt],1}$.

Optimal Scaling Beyond Metropolis: Result

We speed up the process by a factor of d

$$\boldsymbol{X}^d_{[dt]} = (X^d_{[dt],1}, X^d_{[dt],2}, \ldots, X^d_{[dt],d})^T; \quad t > 0.$$

Define a new 1-dimensional process $U^d_t = X^d_{[dt],1}$. Then, $U^d \Rightarrow U$, where U satisfies the Langevin stochastic differential equation,

$$dU_t = (h_\alpha(l))^{1/2} dB_t + h_\alpha(l) \frac{f'(U_t)}{2f(U_t)} dt,$$

with $h_\alpha(l) = l^2 M_\alpha(l)$, where,

Optimal Scaling Beyond Metropolis: Result

We speed up the process by a factor of d

$$\boldsymbol{X}^d_{[dt]} = (X^d_{[dt],1}, X^d_{[dt],2}, \ldots, X^d_{[dt],d})^T; \quad t > 0.$$

Define a new 1-dimensional process $U^d_t = X^d_{[dt],1}$. Then, $U^d \Rightarrow U$, where U satisfies the Langevin stochastic differential equation,

$$dU_t = (h_\alpha(l))^{1/2} dB_t + h_\alpha(l) \frac{f'(U_t)}{2f(U_t)} dt,$$

with $h_\alpha(l) = l^2 M_\alpha(l)$, where,

$$M_\alpha(l) = \int_{\mathbb{R}} \underbrace{g_\alpha(e^b)}_{\text{Coming from } \alpha} \frac{1}{\sqrt{2\pi l^2 J}} \exp\left\{ \frac{-(b + l^2 J/2)^2}{2l^2 J} \right\} db. \quad (2)$$

In previous proofs, $\alpha = \alpha_{MH}$ and $M_\alpha(l)$ is obtained in closed form.

Optimal Scaling Beyond Metropolis: Result

We speed up the process by a factor of d

$$\boldsymbol{X}^d_{[dt]} = (X^d_{[dt],1}, X^d_{[dt],2}, \ldots, X^d_{[dt],d})^T; \quad t > 0.$$

Define a new 1-dimensional process $U^d_t = X^d_{[dt],1}$. Then, $U^d \Rightarrow U$, where U satisfies the Langevin stochastic differential equation,

$$dU_t = (h_\alpha(l))^{1/2} dB_t + h_\alpha(l) \frac{f'(U_t)}{2f(U_t)} dt,$$

with $h_\alpha(l) = l^2 M_\alpha(l)$, where,

$$M_\alpha(l) = \int_{\mathbb{R}} \underbrace{g_\alpha(e^b)}_{\text{Coming from } \alpha} \frac{1}{\sqrt{2\pi l^2 J}} \exp\left\{ \frac{-(b + l^2 J/2)^2}{2l^2 J} \right\} db. \quad (2)$$

In previous proofs, $\alpha = \alpha_{MH}$ and $M_\alpha(l)$ is obtained in closed form. We keep the integral intact and using the invariance property of α get exact weak convergence.

Barker's magic number?

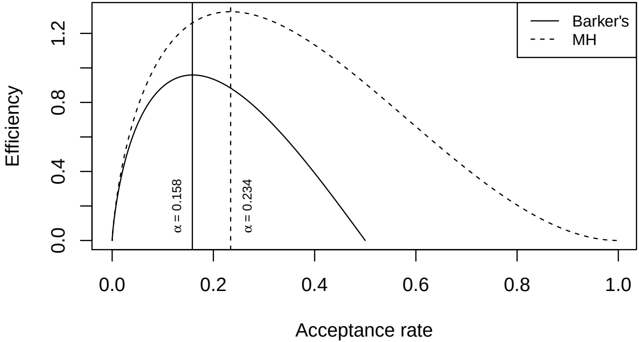
For Barker's α_B, the optimal acceptance probability is 0.158 and the optimal proposal variance is $l_*^2/(d-1)$ where

$$l_* = \frac{2.46}{\sqrt{J}} \qquad \text{compare to } \frac{2.38}{\sqrt{J}} \text{ for MH}$$

Barker's magic number?

For Barker's α_B, the optimal acceptance probability is 0.158 and the optimal proposal variance is $l_*^2/(d-1)$ where

$$l_* = \frac{2.46}{\sqrt{J}} \qquad \text{compare to } \frac{2.38}{\sqrt{J}} \text{ for MH}$$



Back to Bernoulli and Barker!

Unfortunately, optimal scaling for Portkey Barker's are not obtained. However, since Portkey Barker's $\approx$ Barker's, we can use the Barker scaling as a good approximation.

Back to Bernoulli and Barker!

Unfortunately, optimal scaling for Portkey Barker's are not obtained. However, since Portkey Barker's ≈ Barker's, we can use the Barker scaling as a good approximation.

Table: Averaged results from 10 replications using optimal scaling for Barker and Portkey Barker ($\beta = 0.9$).

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS/s	9.63* (-)	14.83 (0.279)

Back to Bernoulli and Barker!

Unfortunately, optimal scaling for Portkey Barker's are not obtained. However, since Portkey Barker's ≈ Barker's, we can use the Barker scaling as a good approximation.

Table: Averaged results from 10 replications using optimal scaling for Barker and Portkey Barker ($\beta = 0.9$).

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS/s	9.63* (-)	14.83 (0.279)

Tuning to about 15.8% here:
$\beta = .90$: 10^4 samples in 40s with estimated ESS $= 514$

Back to Bernoulli and Barker!

Unfortunately, optimal scaling for Portkey Barker's are not obtained. However, since Portkey Barker's $\approx$ Barker's, we can use the Barker scaling as a good approximation.

Table: Averaged results from 10 replications using optimal scaling for Barker and Portkey Barker ($\beta = 0.9$).

β	1	.90
ESS	542 (13.50)	496 (9.00)
ESS/s	9.63* (-)	14.83 (0.279)

Tuning to about 15.8% here:
$\beta = .90$: 10^4 samples in 40s with estimated ESS $= 514$
For Barker, the number in * is not to be trusted:
$\beta = 1$: $< 10^3$ samples in 24hrs and simulation was forcibly stopped!

Main references

- Vats, D., Gonçalves, F., Łatusyński, K., Roberts, G. O., Efficient Bernoulli Factory MCMC for intractable posteriors, Biometrika, 2021+

Main references

- Vats, D., Gonçalves, F., Łatusyński, K., Roberts, G. O., Efficient Bernoulli Factory MCMC for intractable posteriors, Biometrika, 2021+

  Advantages
  - Markovian dynamics are mildly altered for $\beta \approx 1$
  - Exact MCMC
  - Significantly more robust

  Disadvantages
  - Loss of statistical efficiency from MH algorithms
  - Finding the bounds c_θ may be challenging.

Main references

- Vats, D., Gonçalves, F., Łatusyński, K., Roberts, G. O., Efficient Bernoulli Factory MCMC for intractable posteriors, Biometrika, 2021+

  Advantages
  - Markovian dynamics are mildly altered for $\beta \approx 1$
  - Exact MCMC
  - Significantly more robust

  Disadvantages
  - Loss of statistical efficiency from MH algorithms
  - Finding the bounds c_θ may be challenging.

- Agrawal, S., Vats, D., Łatusyński, K., Roberts, G. O., Optimal Scaling of MCMC Beyond Metropolis, 2023.
  - Results for Lazy MH and other acceptance
  - Can be extended to other optimal scaling results