MATH 559

Bayesian Theory and Methods

Dr David A. Stephens

Department of Mathematics & Statistics
Room 1225, Burnside Hall

david.stephens@mcgill.ca

Part 2

Bayesian Computation

Monte Carlo Methods

Monte Carlo: basic principles

Suppose $X_1, \ldots, X_n, \ldots$ are a sequence of random variables. Then as $n \longrightarrow \infty$

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \xrightarrow[\text{p}]{\text{a.s.}} \mathbb{E}[g(X)]$$

and

$$a_n\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i) - b_n\right) \xrightarrow{d} \mathcal{N}(\mu, \sigma^2)$$

for suitable choices of the sequences $\{a_n\}$ and $\{b_n\}$, under mild conditions on the joint distribution of the rvs.

Monte Carlo: basic principles

For probability density $f(x)$, we consider approximating

$$\mathbb{E}_f[g(X)] = \int g(x)f(x)\,dx \qquad \text{by} \qquad \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

where $x_1, \ldots, x_N \sim f$ are an i.i.d sample, *provided* the expectation exists.

We need to establish

- the accuracy of the approximation,
- how the samples from $f(x)$ are obtained.

Does this ever go wrong ?

Monte Carlo: basic principles

Consider computing

$$\int_0^1 \frac{1}{x} \sin(2\pi/x)\, dx$$

by sampling $X_i \sim \mathit{Uniform}(0,1)$, and then computing

$$\frac{1}{N} \sum_{i=1}^{N} \frac{1}{X_i} \sin(2\pi/X_i)$$

Monte Carlo: basic principles

The integral can be computed as

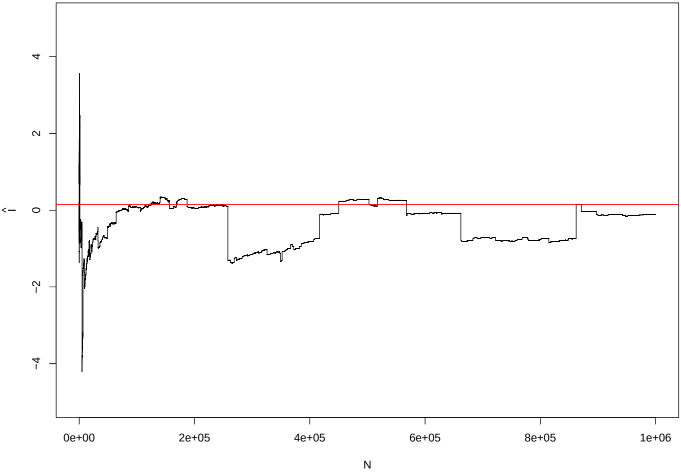
$$\begin{aligned}\int_0^1 \frac{1}{x}\sin(2\pi/x)\,dx &= \int_1^\infty \frac{\sin(2\pi t)}{t}\,dt \\ &= \int_0^\infty \frac{\sin(t)}{t}\,dt - \int_0^{2\pi} \frac{\sin(t)}{t}\,dt \\ &= \mathrm{Si}(\infty) - \mathrm{Si}(2\pi)\end{aligned}$$

where Si(.) is a special function (the *sine integral*).

We have that $\mathrm{Si}(\infty) = \pi/2 \bumpeq 0.1526$.

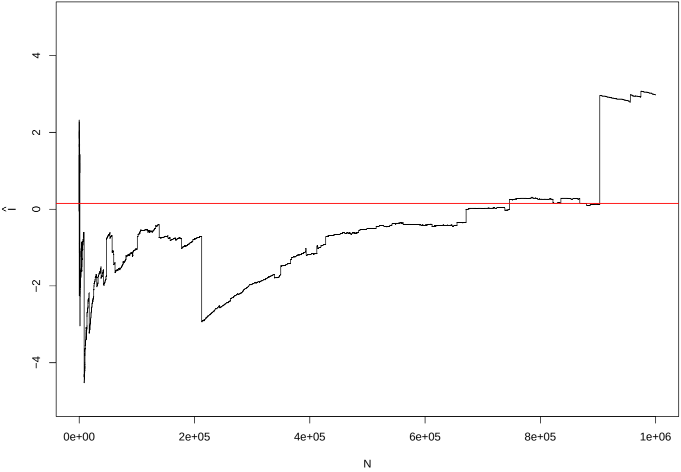
Monte Carlo: basic principles

Run 1:



Monte Carlo: basic principles

Run 2:



Monte Carlo: basic principles

Occasional large values of

$$Y_i = \frac{1}{X_i} \sin(2\pi/X_i)$$

cause the sample average to not converge as N gets large.

A sufficient condition for strong convergence is

$$\int |g(x)| f(x)\, dx < \infty$$

which does *not* hold here.

Monte Carlo Estimation: Statistical Properties

In situations where the Monte Carlo estimator

$$\widehat{I}_N(g) = \frac{1}{N}\sum_{i=1}^{N} g(X_i) \xrightarrow[\text{p}]{\text{a.s.}} \mathbb{E}[g(X)] = \mu(g)$$

say, *and* a central limit theorem applies, we have that

$$\sqrt{N}(\widehat{I}_N(g) - \mu(g)) \xrightarrow{d} \mathcal{N}(0, V(g))$$

where

$$V(g) = \mathrm{Var}[g(X)] = \int (g(x) - \mu(g))^2 \, f(x)\, dx$$

The Monte Carlo estimator exhibits $o_P(\sqrt{N})$ convergence.

Monte Carlo Estimation

Motivated by the deterministic approximation

$$\int g(x)f(x)\,dx \bumpeq \sum_{j=0}^{k} w_j g(x_j)$$

where weights are determined by $f(x)$, to minimize the error in the approximation, it should be advantageous to choose design points $x_0, \ldots, x_k$ where g is largest.

In a Monte Carlo setting, it seems clear that the estimator will converge more quickly and have lower variance for finite N when $f(x)$ generates points in regions where $g(x)$ is large in magnitude.

Monte Carlo Estimation

Example:

See `knitr 4`

Importance sampling

If f_0 is a pdf with support including the support of f, then

$$\int g(x)f(x)\,dx = \int g(x)f(x)\frac{f_0(x)}{f_0(x)}\,dx = \int \frac{g(x)f(x)}{f_0(x)}f_0(x)\,dx.$$

and so

$$\mathbb{E}_f[g(X)] = \mathbb{E}_{f_0}\left[\frac{g(X)f(X)}{f_0(X)}\right].$$

An estimator of the expecat

$$\widehat{I}_N^{(f_0)}(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{g(X_i)f(X_i)}{f_0(X_i)}$$

where $X_1, \ldots, X_N \sim f_0(.)$.

Importance sampling

$\widehat{I}_N^{(f_0)}$ is termed the *importance sampling* estimator, and f_0 is termed the *importance sampling density*.

Note that

$$\widehat{I}_N^{(f_0)}(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{f_0(X_i)}g(X_i) = \frac{1}{N}\sum_{i=1}^{N} w_0(X_i)g(X_i)$$

say, where

$$w_0(X_i) = \frac{f(X_i)}{f_0(X_i)}$$

is the *importance sampling weight*.

Importance sampling

Note that

$$\mathbb{E}_{f_0}\left[\frac{f(X)}{f_0(X)}\right] = \int f(x)\, dx = 1$$

so

$$\mathbb{E}_{f_0}\left[\frac{1}{N}\sum_{i=1}^{N} w_0(X_i)\right] = 1$$

although for any realization

$$\frac{1}{N}\sum_{i=1}^{N} w_0(x_i) \neq 1$$

in general.

Optimal Importance Sampling

We now seek guidelines for choosing f_0 optimally. Note first that the variance of $\widehat{I}_N^{(f_0)}(g)$ is finite if and only if

$$\frac{g(X)f(X)}{f_0(X)}$$

has finite variance, that is, if and only if

$$\mathbb{E}_{f_0}\left[\left\{\frac{g(X)f(X)}{f_0(X)}\right\}^2\right] = \int_{-\infty}^{\infty}\left\{\frac{g(x)f(x)}{f_0(x)}\right\}^2 f_0(x)\,dx$$

is finite. The *optimal* choice is then $f_0(x) \propto |g(x)|f(x)$.

Optimal Importance Sampling

If

$$\frac{f(x)}{f_0(x)}$$

is *unbounded* on the support of f, the variance of $\widehat{I}_N^{(f_0)}(g)$ is *not finite*. Therefore, for f with unbounded support, we must ensure that this ratio stays bounded particularly in the *tails*.

Optimal Importance Sampling

Note that, in general, if $X_1, \ldots, X_N, \ldots \sim f_0$, then

$$\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{f_0(X_i)} \xrightarrow{a.s.} \int_{-\infty}^{\infty} \frac{f(x)}{f_0(x)} f_0(x)\, dx = 1$$

so therefore

$$\widetilde{I}_N^{(f_0)}(g) = \frac{\sum_{i=1}^{N} \frac{g(X_i) f(X_i)}{f_0(X_i)}}{\sum_{i=1}^{N} \frac{f(X_i)}{f_0(X_i)}} \xrightarrow{a.s.} \mathbb{E}_f[g(X)]$$

also, if the expectation exists. This estimator $\widetilde{I}_N^{(f_0)}(g)$ may have smaller variance than $\widehat{I}_N^{(f_0)}(g)$

Optimal Importance Sampling

It seems appealing to combine this with the optimality result. The estimator

$$\widetilde{I}_N^{(f_0)}(g) = \frac{\sum_{i=1}^{N} \frac{g(X_i)f(X_i)}{f_0(X_i)}}{\sum_{i=1}^{N} \frac{f(X_i)}{f_0(X_i)}} = \frac{\sum_{i=1}^{N} \frac{g(X_i)}{|g(X_i)|}}{\sum_{i=1}^{N} \frac{1}{|g(X_i)|}}$$

is feasible. When $g(.)$ is positive

$$\widetilde{I}_N^{(f_0)}(g) = \frac{N}{\sum_{i=1}^{N} \frac{1}{g(X_i)}}$$

that is, the *harmonic mean* estimator. Unfortunately, this estimator often has poor properties.

Optimal Importance Sampling

However, to get as close to the variance bound as possible, a sensible objective is to choose f_0 so that

$$\frac{|g(x)|f(x)}{f_0(x)}$$

is almost constant, such that the variance is *finite*.

This suggests designing f_0 to have high density whenever the original integrand $|g(x)|f(x)$ is large, subject to the constraint

$$\frac{f(x)}{f_0(x)} < M \qquad \text{or} \qquad \mathbb{E}_{f_0}\left[\frac{f(X)}{f_0(X)}\right] < M$$

for some finite bound M.

Importance Sampling in Higher Dimensions

All of the previous results carry across to the case where the target integral is an integral in dimension higher than one.

- $g(\mathbf{x})$ is a scalar function of vector argument $\mathbf{x}$,
- $f(\mathbf{x})$ and $f_0(\mathbf{x})$ are multivariate densities.

In higher dimensions, in general, many more random samples are needed to obtain sufficient accuracy than in the univariate case.

Random number generation

Monte Carlo sampling requires ready access to random samples from univariate or multivariate distributions.

There are many straightforward techniques available to obtain random samples from standard distributions once a random sample of Uniform random variables is available.

Rejection sampling

To sample from $f(x)$ using samples from from the density $f_0(x)$, where for all x and for some finite M,

$$\frac{f(x)}{f_0(x)} < M \qquad \text{i.e.} \qquad f(x) < Mf_0(x)$$

we may use the following algorithm:

1. generate x from f_0
2. generate u from $Uniform(0,1)$
3. if
$$u \leqslant \frac{f(x)}{Mf_0(x)}$$
accept x as a variate from f; if this inequality is not met, return to 1.

Rejection sampling

Now

$$\begin{aligned}
\Pr[X \text{ is accepted}] &= \Pr\left[U \leqslant \frac{f(X)}{Mf_0(X)}\right] \\
&= \int_{-\infty}^{\infty} \left\{ \int_0^{f(x)/(Mf_0(x))} du \right\} f_0(x)\, dx \\
&= \int_{-\infty}^{\infty} \frac{f(x)}{Mf_0(x)} f_0(x)\, dx \\
&= \frac{1}{M} \int_{-\infty}^{\infty} f(x)\, dx = \frac{1}{M}
\end{aligned}$$

Rejection sampling

For $t \in \mathbb{R}$,

$$\begin{aligned}
\Pr[X \leqslant t | X \text{ is accepted}] &= \frac{\Pr[X \leqslant t, X \text{ is accepted}]}{\Pr[X \text{ is accepted}]} \\
&= M \Pr\left[X \leqslant t, U \leqslant \frac{f(X)}{M f_0(X)}\right] \\
&= M \int_{-\infty}^{t} \left\{ \int_0^{f(x)/(M f_0(x))} du \right\} f_0(x)\, dx \\
&= M \int_{-\infty}^{t} \frac{f(x)}{M f_0(x)} f_0(x)\, dx \\
&= \int_{-\infty}^{t} f(x)\, dx
\end{aligned}$$

that is, the density of accepted points is precisely $f(x)$.

Rejection sampling

This is the *rejection sampling* (or *accept-reject*) algorithm with *proposal density* f_0; it works for arbitrary multivariate distributions.

If $f(x)$ is bounded with support a bounded subset $\mathcal{X}$ of $\mathbb{R}$, then f_0 can be the Uniform density on $\mathcal{X}$, although this is not necessarily the optimal choice. Recall that

$$\Pr[X \text{ is accepted}] = \frac{1}{M}$$

so ideally M should be as small as possible.

Rejection sampling: Efficiency

Neither f nor f_0 need to be normalized for this algorithm to be valid:

- If $f(x) = mg(x)$ and $f_0(x) = m_0 g_0(x)$, then

  $$\frac{f(x)}{f_0(x)} = \frac{m}{m_0}\frac{g(x)}{g_0(x)} < M$$

  or

  $$\frac{g(x)}{g_0(x)} < M' = \frac{mM}{m_0}$$

  is the rejection sampling bound.
- We proceed by bounding $g(x)/g_0(x)$.
- The acceptance probability is now indeterminate, however, by monitoring the empirical acceptance rate, an estimate of m/m_0 can be obtained.

Rejection sampling

Example: Normal mixture

Consider sampling from the normal mixture

$$f(x) = \frac{1}{4}\phi(x+2) + \frac{3}{4}\phi(x-1)$$

where $\phi(.)$ is the standard normal pdf. Consider rejection sampling from this density using

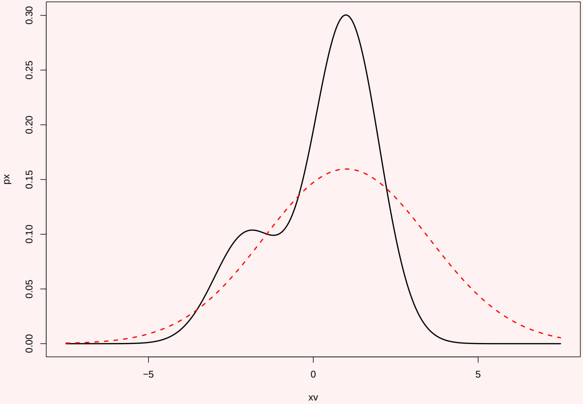
$$f_0(x) = \frac{1}{\sigma}\phi\left(\frac{x-1}{\sigma}\right)$$

for some variance σ^2.

Rejection sampling

Example: Normal mixture

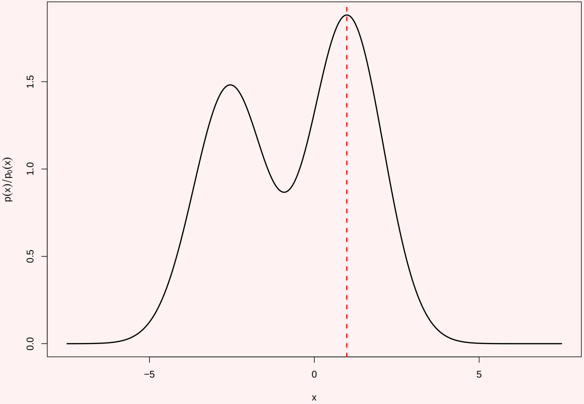
Target $f(x)$ (solid) and $f_0(x)$ (dashed) with $\sigma = 2.5$.



Rejection sampling

Example: Normal mixture

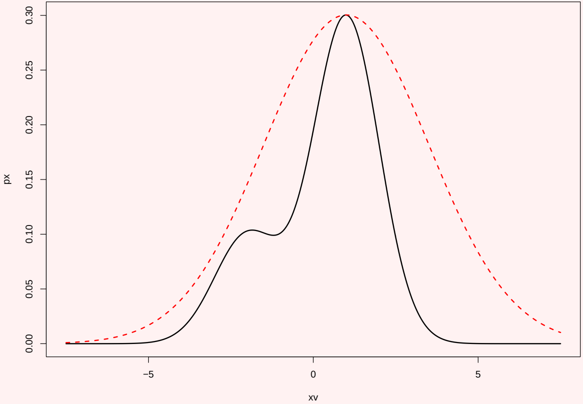
$f(x)/f_0(x)$ maximized at $x = 0.9863$, yielding $M = 1.8821$:



Rejection sampling

Example: Normal mixture

$f(x)$ bounded by $Mf_0(x)$:



Rejection sampling

If x is a variate from f_0, then $f_0(x)$ is the value of the density at that variate, and $Mf_0(x)$ is the scaled version. Consider a vertical slice at x which is the line segment

$$(x, 0) \longrightarrow (x, Mf_0(x))$$

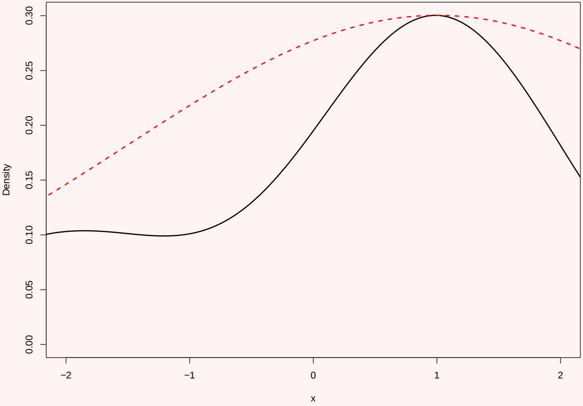
By assumption $f(x) \leqslant Mf_0(x)$. If u is a $\mathit{Uniform}(0, 1)$ variate, then $uMf_0(x)$ is the portion of the vertical slice, and if

$$f(x) \leqslant uMf_0(x)$$

then $f(x)$ is *below* the random point on the line segment, and hence is rejected.

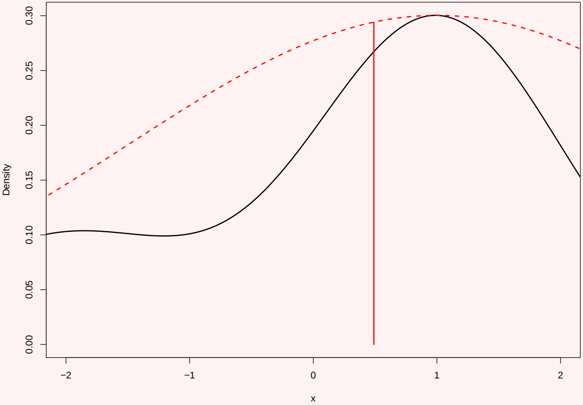
Rejection sampling

Example: Normal mixture



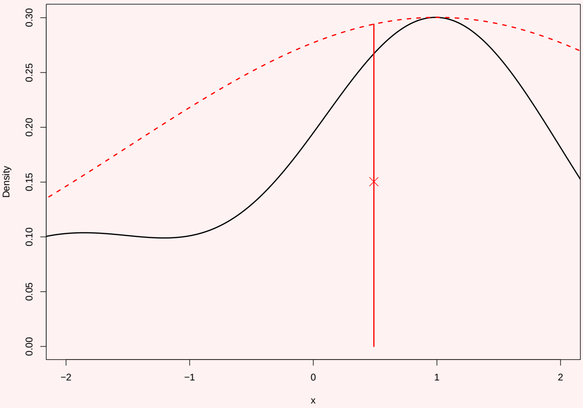
Rejection sampling

Example: Normal mixture



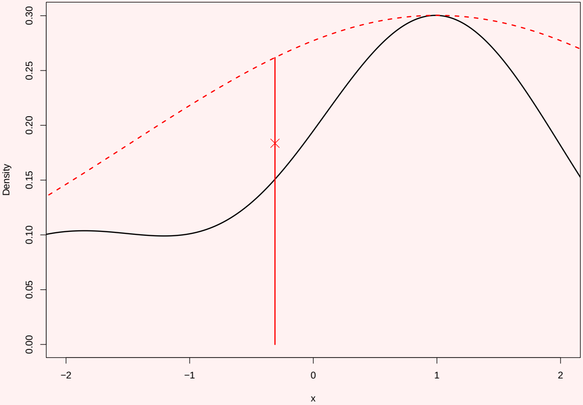
Rejection sampling

Example: Normal mixture



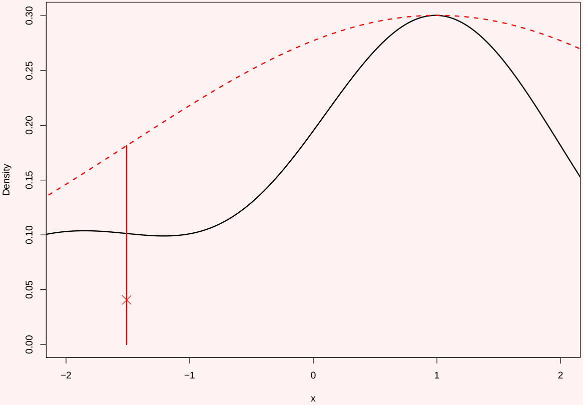
Rejection sampling

Example: Normal mixture



Rejection sampling

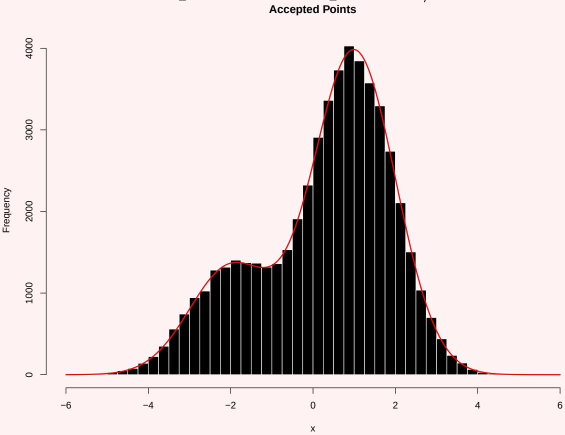
Example: Normal mixture



Rejection sampling

Example: Normal mixture

53078 out of 100000 points accepted ($1/M = 0.5313$).



Rejection sampling and Importance sampling

There is obviously a clear connection between rejection sampling and importance sampling

- both rely on choosing a suitable $f_0(x)$
- the boundedness of the ratio $f(x)/f_0(x)$ is crucial in the construction of the procedure

The difference is that rejection sampling produces i.i.d. samples from $f(x)$, whereas importance sampling approximates numerical integration with respect to $f(x)$.

Sampling Importance Resampling

Sampling Importance Resampling (SIR) can be used to sample (approximately) from density $f(x)$ by *re-weighting* and then *resampling* samples from $f_0(x)$:

1. Generate variates $x_1, \ldots, x_N$ from f_0.
2. Compute renormalized weights $w_1, \ldots, w_N$ given by

$$w_i = \frac{f(x_i)/f_0(x_i)}{\sum_{j=1}^{N} (f(x_j)/f_0(x_j))} \qquad i = 1, \ldots, N.$$

3. Resample y_i from the discrete distribution on $\{x_1, \ldots, x_N\}$ with masses $\{w_1, \ldots, w_N\}$.

Sampling Importance Resampling

Example: Normal mixture

$$
\begin{aligned}
f(x) &= \frac{1}{4}\phi\left(x+2.5\right)+\frac{3}{4}\phi\left(x-1\right)\\
f_0(x) &= \frac{1}{\sigma}\phi\left(x/\sigma\right)
\end{aligned}
$$

for some $\sigma > 0$.

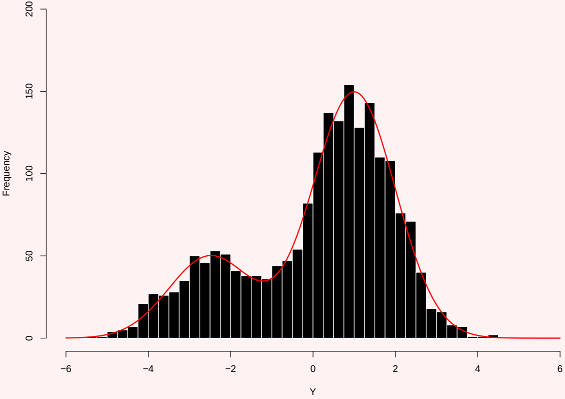
Choosing σ poorly can compromise the SIR algorithm.

- need to ensure there are samples in the tails of $f(x)$.

Sampling Importance Resampling

Example: Normal mixture

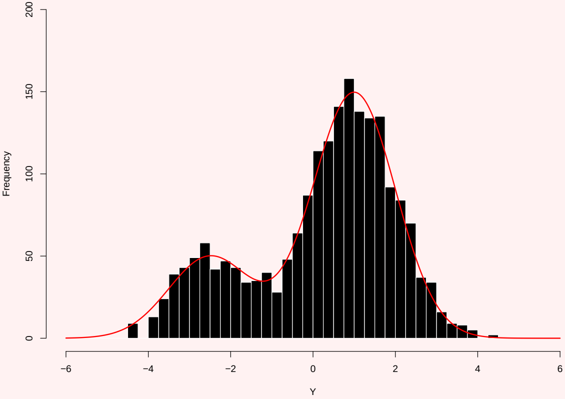
SIR with $\sigma = 2$ ($N = 100000$, 2000 resampled points):



Sampling Importance Resampling

Example: Normal mixture

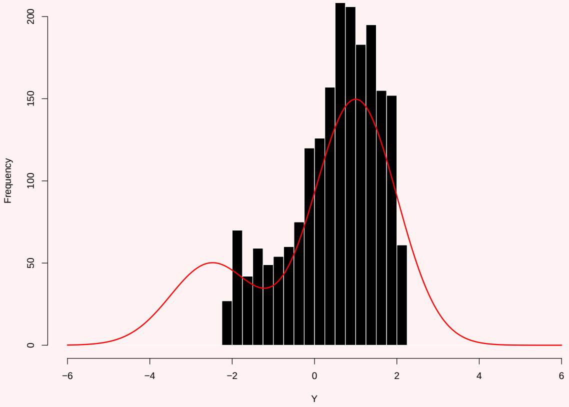
SIR with $\sigma = 1$ ($N = 100000$, 2000 resampled points):



Sampling Importance Resampling

Example: Normal mixture

SIR with $\sigma = 0.5$ ($N = 100000$, 2000 resampled points):



Variance Reduction Methods

Example:

See `knitr 5`

Markov chain Monte Carlo

A brief introduction to Markov chains

A *Markov chain* is a sequence of rvs, $\{X_t\}$, for which

$$\Pr[X_t \in B | X_1 = x_1, \ldots, X_{t-1} = x_{t-1}] = \Pr[X_t \in B | X_{t-1} = x_{t-1}].$$

X_t is *conditionally independent* of $X_0, \ldots, X_{t-2}$ *given* X_{t-1}.

In the simplest case, the $\{X_t\}$ takes values on the *finite* (discrete) state space

$$\mathcal{S}_X = \{s_1, \ldots, s_d\}$$

and the chain is *homogeneous*, that is, the stochastic properties of $\{X_t\}$ do not change with time.

A brief introduction to Markov chains

The Markov chain is characterized by its initial state X_0 or its initial distribution $p^{(0)}$, and its *transition matrix* P, a $d \times d$ *stochastic* matrix whose rows sum to one, such that

$$P_{ij} = \Pr[X_t = s_j | X_{t-1} = s_i]$$

describes the set of one-step ahead conditional probabilities. Denote by

$$p_{ij}(k) = \Pr[X_{t_0+k} = s_j | X_{t_0} = s_i]$$

the k-step ahead probabilities.

A brief introduction to Markov chains

If $P(k)$ is the matrix of k-step ahead probabilities, then

$$P(k) = P^k$$

and then the k-step distribution is

$$p^{(k)} = p^{(0)} P^k$$

A brief introduction to Markov chains

The properties of the chain depend on P. The chain is

- *irreducible* if $p_{ij}(k) > 0$, for all i, j, and at least one k.
- *aperiodic* if all states have period 1: that is, for each i, returns to state i can occur after any number of steps.

  The *period* of state i is defined as the greatest common divisor of the set of possible return times, $\mathcal{R}$,

$$\mathcal{R} = \{r : \Pr[X_r = s_i | X_0 = s_i] > 0\}$$

A brief introduction to Markov chains

- *recurrent* if all states are recurrent, that is, the probability of returning to each state in a finite number of steps is positive. Let $T_i = \inf\{k : X_k = i | X_0 = i\}$. State i is recurrent if and only if

  $$\Pr[T_i = \infty] = 0$$

  and *transient* otherwise. If $\mathbb{E}[T_i] < \infty$, state i is termed *positive recurrent*, otherwise it is termed *null recurrent*.

A brief introduction to Markov chains

A *stationary* or *invariant distribution*, π^*, of a homogeneous Markov chain is the $1 \times d$ vector of probabilities such that

$$\pi^* = \pi^* P$$

that is, for each i,

$$\pi_i^* = \sum_j \pi_j^* P_{ji}$$

A brief introduction to Markov chains

The *equilibrium distribution* of the chain, π, is defined by

$$\pi = \lim_{k \longrightarrow \infty} p^{(k)} = p^{(0)} \lim_{k \longrightarrow \infty} P^k$$

when this limit exists and is independent of $p^{(0)}$. That is, we may compute π as

$$\mathbf{1}\pi = \lim_{k \longrightarrow \infty} P^k$$

if the limit exists. The equilibrium distribution is a stationary distribution.

A brief introduction to Markov chains

An irreducible chain has an equilibrium distribution if and only if all of its states are positive recurrent, in which case π is *unique*, and can be computed as

$$\lim_{k \longrightarrow \infty} P^k = \mathbf{1}\pi.$$

where $\mathbf{1}$ is the $d \times 1$ vector of 1s.

The equilibrium/stationary distribution π can be computed by solving $\pi = \pi P$.

A brief introduction to Markov chains

Key aspects of the stationary distribution are that

(a) As the k-step ahead probability matrix P^k converges to a matrix with d identical rows, the Markov chain can eventually “forget" its initial value X_0.

(b) Realized values of $\{X_t\}$ have statistical properties that exhibit convergence to the stationary distribution, that is, for $i = 1, \ldots, d$,

$$\lim_{n \longrightarrow \infty} \frac{\sum_{k=1}^{n} \mathbb{1}_{\{s_i\}}(X_k)}{n} = \pi_i$$

A brief introduction to Markov chains

A Markov chain is *reversible* if, for every $n \geqslant 1$,

$$X_0, X_1, \ldots, X_{n-1}, X_n$$

and

$$X_n, X_{n-1}, \ldots, X_1, X_0$$

have the same joint distribution.

A brief introduction to Markov chains

It follows that the reverse chain is also Markov and that the individual X_k have the same marginal distribution: for arbitrary state sequence $(l_{k+2}, \ldots, l_n)$,

$$\begin{aligned}
\Pr[X_k = i | X_{k+1} = j, X_{k+2} = l_{k+2}, \ldots, X_n = l_n] \\
&= \frac{\Pr[X_k = i, X_{k+1} = j, X_{k+2} = l_{k+2}, \ldots, X_n = l_n]}{\Pr[X_{k+1} = j, X_{k+2} = l_{k+2}, \ldots, X_n = l_n]} \\
&= \frac{\pi_i P_{ij} P_{j,l_{k+2}} \cdots P_{l_{n-1},l_n}}{\pi_j P_{j,l_{k+2}} \cdots P_{l_{n-1},l_n}} \\
&= \frac{\pi_i P_{ij}}{\pi_j}
\end{aligned}$$

which only depends on i and j.

A brief introduction to Markov chains

A homogeneous Markov chain with stationary distribution π is reversible if

$$\pi_i P_{ij} = \pi_j P_{ji}$$

for all states i and j. This is also termed the *detailed balance* condition.

Note that *if* this equation holds for a specified π, *then* this implies that the P has been specified so as to have stationary distribution π.

Discrete Markov chains

Note that the $\{X_t\}$ are *dependent* random variables, so the standard frequentist asymptotic laws do not directly apply.

However, the *ergodic theorem* applies for irreducible, aperiodic and positive recurrent Markov chains, in particular

$$\frac{1}{N}\sum_{t=1}^{N} g(X_t) \xrightarrow{a.s.} \mathbb{E}_\pi[g(X)]$$

for all bounded functions g, provided

$$\mathbb{E}_\pi[|g(X)|] < \infty$$

Discrete Markov chains

A Central Limit Theorem result also holds under mild regularity conditions, specifically,

$$\sqrt{N}\left(\frac{1}{N}\sum_{t=1}^{N} g(X_t) - \mathbb{E}_\pi[g(X)]\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2(g))$$

where

$$\sigma^2(g) = \mathrm{Var}_\pi[g(X_0)] + 2\sum_{t=1}^{\infty} \mathrm{Cov}_\pi[g(X_0), g(X_t)].$$

Discrete Markov chain: example

Example: $d = 2$

Consider $d = 2$, with

$$P = \left[\begin{array}{cc} 0.3 & 0.7 \\ 0.9 & 0.1 \end{array} \right]$$

Then $\pi = (9/16, 7/16)$. Here

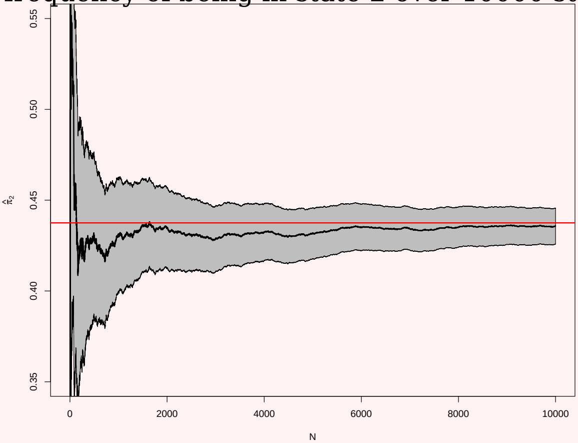
$$\pi_1 P_{12} = \frac{9}{16} \times \frac{7}{10} = \frac{63}{160} \qquad \pi_2 P_{21} = \frac{7}{16} \times \frac{9}{10} = \frac{63}{160}$$

so this chain is reversible.

Discrete Markov chain: example

Example: $d = 2$

Relative frequency of being in state 2 over 10000 steps.



Constructing reversible chains

In the 2 × 2, for a reversible chain, we require

$$\pi_1 P_{12} = (1 - \pi_1) P_{21}$$

or

$$\frac{\pi_1}{(1 - \pi_1)} = \frac{P_{21}}{P_{12}}$$

Constructing reversible chains

Suppose

$$P_{12} = \min\left\{1, \frac{1-\pi_1}{\pi_1}\right\} \qquad P_{21} = \min\left\{1, \frac{\pi_1}{1-\pi_1}\right\}$$

Then

$$\begin{aligned}
\pi_1 P_{12} &= \pi_1 \min\left\{1, \frac{1-\pi_1}{\pi_1}\right\} \\
&= \min\left\{\pi_1, 1-\pi_1\right\} \\
&= \min\left\{1-\pi_1, \pi_1\right\} \\
&= (1-\pi_1)\min\left\{1, \frac{\pi_1}{1-\pi_1}\right\} \\
&= (1-\pi_1)P_{21}
\end{aligned}$$

Constructing reversible chains

The above Markov chain we can think of as acting as follows:

- *If* $X_t = 1$: *propose* setting $X_{t+1} = 2$, but only *accept* this move with probability

  $$\min\left\{1, \frac{1-\pi_1}{\pi_1}\right\}$$

  otherwise set $X_{t+1} = 1$.
- *If* $X_t = 2$: *propose* setting $X_{t+1} = 1$, but only *accept* this move with probability

  $$\min\left\{1, \frac{\pi_1}{1-\pi_1}\right\}$$

  otherwise set $X_{t+1} = 2$.

Constructing reversible chains

A generalization of this approach is as follows:

- If $X_t = 1$: simulate z from $\{1,2\}$ with probabilities (q_{11}, q_{12}). If $z = 2$, set $X_{t+1} = 2$ with probability

  $$\alpha_{12} = \min\left\{1, \frac{\pi_2}{\pi_1}\frac{q_{21}}{q_{12}}\right\}$$

  otherwise set $X_{t+1} = 1$.

- If $X_t = 2$: simulate z from $\{1,2\}$ with probabilities (q_{21}, q_{22}). If $z = 1$, set $X_{t+1} = 1$ with probability

  $$\alpha_{21} = \min\left\{1, \frac{\pi_1}{\pi_2}\frac{q_{12}}{q_{21}}\right\}$$

  otherwise set $X_{t+1} = 2$.

Constructing reversible chains

The transition probabilities are then

$$\begin{aligned} P_{12} &= \Pr[X_{t+1} = 2 | X_t = 1] = q_{12}\alpha_{12} \\ P_{21} &= \Pr[X_{t+1} = 1 | X_t = 2] = q_{21}\alpha_{21} \end{aligned}$$

so therefore

$$\pi_1 P_{12} = \pi_1 q_{12}\alpha_{12} = \pi_1 q_{12} \min\left\{1, \frac{\pi_2}{\pi_1}\frac{q_{21}}{q_{12}}\right\} = \min\{\pi_1 q_{12}, \pi_2 q_{21}\}$$

$$\pi_2 P_{21} = \pi_2 q_{21}\alpha_{21} = \pi_2 q_{21} \min\left\{1, \frac{\pi_1}{\pi_2}\frac{q_{12}}{q_{21}}\right\} = \min\{\pi_2 q_{21}, \pi_1 q_{12}\}$$

and

$$\pi_1 P_{12} = \pi_2 P_{21}.$$

Constructing reversible chains

This allows the generalization to the case

$$\mathcal{S}_X = \{1, 2, \ldots, d, \ldots\}.$$

Let π be an arbitrary discrete distribution, and matrix Q define the *proposal probabilities*

$$[Q]_{ij} = \Pr[Z = j | X_t = i]$$

for all i, j.

Constructing reversible chains

Define the *acceptance probabilities*

$$\alpha_{ij} = \min\left\{1, \frac{\pi_j}{\pi_i}\frac{q_{ji}}{q_{ij}}\right\}$$

and implement the chain as follows: when $X_t = i$

- set $X_{t+1} = Z = j$ with probability α_{ij},
- otherwise, set $X_{t+1} = X_t = i$.

Constructing reversible chains

This Markov chain satisfies detailed balance

$$\pi_i P_{ij} = \pi_j P_{ji} \qquad \text{for all } i, j$$

provided it is irreducible, aperiodic and positive recurrent.

Note that the rows of Q must sum to 1 as

$$\sum_{j=1}^{\infty} \Pr[Z = j | X_t = i] = 1$$

so Q defines a stochastic proposal (or *transition*) matrix.

Constructing reversible chains: example

Example: Poisson distribution

Suppose, for $\lambda > 0$,

$$\pi_i = \frac{e^{-\lambda}\lambda^i}{i!} \qquad i = 0, 1, 2, \ldots.$$

Suppose

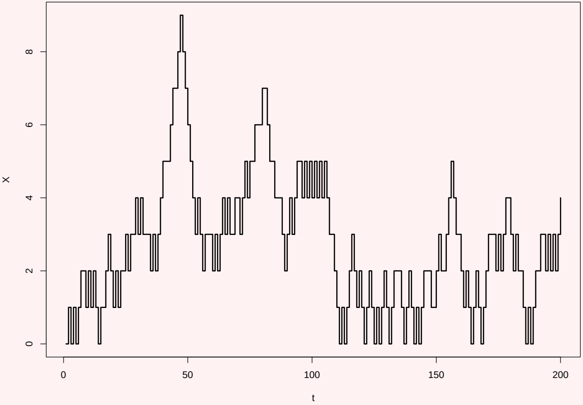
$$q_{ij} = \begin{cases} 1 & i = 0, j = 1 \\ \frac{1}{2} & i \geqslant 1, j = i-1, i+1 \\ 0 & \text{otherwise} \end{cases}$$

Z is proposed uniformly on the finite set $\{x_t - 1, x_t + 1\}$, unless $X_t = 0$, in which case $Z = 1$ is proposed.

Constructing reversible chains: example

Example: Poisson distribution

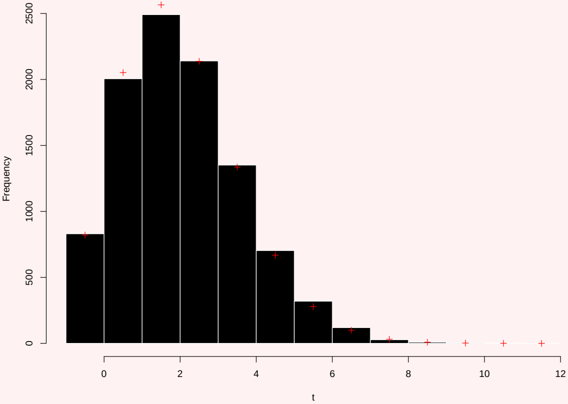
First 200 steps of the chain starting at $X_0 = 0$ with $\lambda = 2.5$.



Constructing reversible chains: example

Example: Poisson distribution

Histogram of states for $N = 10000$ steps (true values +)



Discrete Markov chains: Recap

We can sample from target discrete distribution, π,

- specify the stochastic matrix Q
- initialize the chain by setting X_0
- for each t, if $X_t = i$, use the ith row of Q as a discrete distribution for proposing Z
- If $Z = j$, accept $X_{t+1} = j$ with probability α_{ij}

$$\alpha_{ij} = \min \left\{ 1, \frac{\pi_j}{\pi_i} \frac{q_{ji}}{q_{ij}} \right\}$$

  otherwise set $X_{t+1} = X_t = i$.
- collect the sequence $\{X_t\}$.

Discrete Markov chains: Recap

Example:

See `knitr 6`

Continuous State Space Markov Chains

The theory above extends to *continuous* state spaces, $\mathbb{X}$.

We must specify a *transition kernel*

$$P(x, B) = \int_B P(x, z)\, dz$$

$P(x, B)$ determines the probability of making the transition from current value x into the set $B \subset \mathbb{X}$ in any given step.

Continuous State Space Markov Chains

We retain the *discrete time* nature of the Markov chain, and again consider outcome sequences $\{x_1, x_2, \ldots, x_n, \ldots\}$.

Transitions are implemented using a *transition density*

$$P\left(x, z\right) \equiv P\left(x \to z\right)$$

which specifies a conditional probability density in z, given the current value x, for $x, z \in \mathbb{X}$.

Continuous State Space Markov Chains

By analogy with the discrete case, the stationary distribution π for the continuous state space chain must satisfy

$$\pi(x) = \int P(z, x)\pi(z)\, dz$$

A reversible chain must satisfy detailed balance

$$\pi(x)P(x, z) = \pi(z)P(z, x)$$

for all x and z. Given P, we can in theory solve for π.

In the context of sampling from a *target* probability distributions, we wish to *specify* π, and *then find* a P such that its equilibrium distribution is π.

The Metropolis-Hastings Algorithm

We attempt to mimic the construction of a Markov chain with stationary distribution π used in the discrete case.

Let Q be any proposal (transition) kernel suitable for moving (exhaustively) around $\mathbb{X}$, with associated transition density q such that

$$q\left(z,x\right)=q(z\longrightarrow x)>0$$

for all x,z.

In fact, this can be relaxed to the condition that requires $Q^{n}\left(x,z\right)>0$ for all $x,z\in\mathbb{X}$, separated by n steps in the chain).

The Metropolis-Hastings Algorithm

Then, for $z \neq x$, define

$$P\left(x, z\right) = q\left(x, z\right) \alpha\left(x, z\right)$$

where

$$\alpha\left(x, z\right) = \min\left\{1, \frac{\pi\left(z\right)}{\pi\left(x\right)} \frac{q\left(z, x\right)}{q\left(x, z\right)}\right\}$$

defines an *acceptance probability* for the move from x to z.

The Metropolis-Hastings Algorithm

Under this transition kernel or density P with transition density if the current state of the chain at step n is $X_n = x$, then the next value of the chain is either

- a *new value* $X_{n+1} = z$, generated from the conditional density $q(x, z)$,
- or the *current value* $X_{n+1} = x$.

The value z the proposed or *candidate* state.

Thus, starting from the n^{th} step when $x_n = x$, we have the following algorithm for implementing the continuous state space Markov chain:

The Metropolis-Hastings Algorithm

1. Generate z from conditional density $q(x, \cdot)$ given x;
2. Compute $\alpha(x, z)$;
3. Generate u from $\mathit{Uniform}(0, 1)$
   - if $u \leqslant \alpha(x, z)$, *accept* the move to z and set $X_{n+1} = z$
   - if $u > \alpha(x, z)$, *reject* the move to z and set $X_{n+1} = x$
4. Return to 1 to generate X_{n+2}

and so on.

This is the *Metropolis-Hastings (MH)* algorithm.

The Metropolis Algorithm

The general algorithm above has some special cases of interest. If q is chosen such that

$$q\left(x,z\right)=q\left(z,x\right)$$

so that q is symmetric in its arguments, then

$$\alpha\left(x,z\right)=\min\left\{1,\frac{\pi\left(z\right)}{\pi\left(x\right)}\right\}$$

and the move to z is accepted with certainty if the target probability density at z is higher than at x.

The Metropolis Algorithm

A simple symmetric transition density has

$$Z|X_n = x \sim \mathcal{N}\left(x, \sigma_q^2\right)$$

Choosing σ_q^2 small encourages many small moves.

This is the original Markov chain simulation algorithm, known as the *Metropolis Algorithm*.

Many such “local" moves can be proposed. Note that it is important to respect any parameter constraints in the proposal.

Independence Metropolis-Hastings

The *independence Metropolis-Hastings algorithm* uses

$$q(x,z) = q(z)$$

that is, independent of the current value of the chain. This still defines a Markov chain as

$$p(x,z) = q(z)\,\alpha(x,z)$$

still depends on x through $\alpha(x,z)$. If π can be approximated by a density q (as in rejection sampling), then this method can work well.

Metropolis-Hastings algorithm

Example: Gamma density

Suppose, for $\gamma > 0$

$$\pi(x) = \frac{1}{\Gamma(\gamma)} x^{\gamma-1} e^{-x} \qquad x > 0.$$

Suppose, first that $q(x, z)$ is specified as a *reflected* normal density, that is, we propose z by simulating

$$Y | X_t = x \sim \mathcal{N}(x, \sigma_q^2),$$

and setting $Z = |Y|$.

Metropolis-Hastings algorithm

Example: Gamma density

Note that

$$\Pr[Z \leqslant z | X = x] = \Pr[|Y| \leqslant z | X = x] = \Pr[-z \leqslant Y \leqslant z | X = x]$$

so therefore

$$\Pr[Z \leqslant z | X = x] = \Phi((z - x)/\sigma_q) - \Phi((-z - x)/\sigma_q)$$

and, on differentiation wrt z,

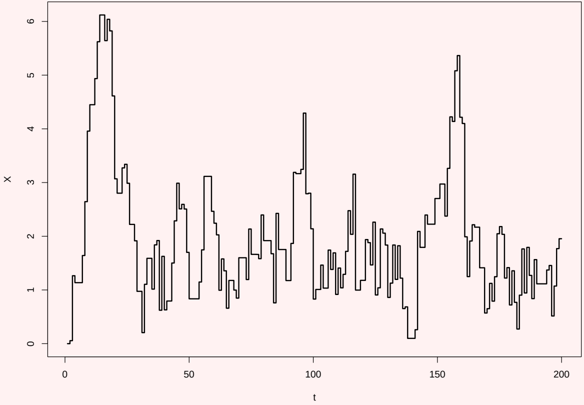
$$q(x, z) = \frac{1}{\sigma_q}(\phi((z - x)/\sigma_q) + \phi((-z - x)/\sigma_q)) = q(z, x)$$

as ϕ is an even function.

Metropolis-Hastings algorithm

Example: Gamma density

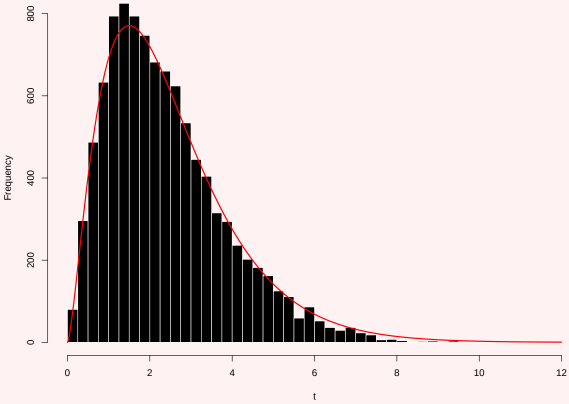
200 steps starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 1$.



Metropolis-Hastings algorithm

Example: Gamma density

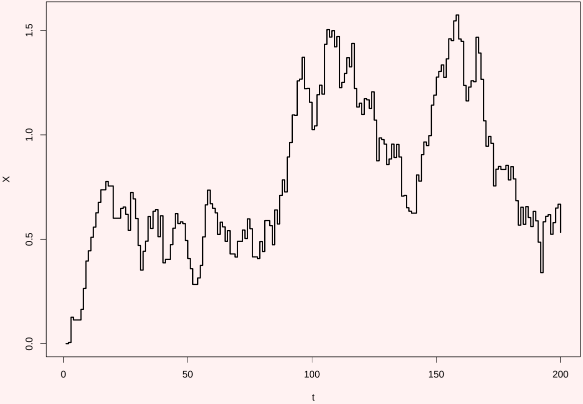
Histogram of states visited over $N = 10000$ steps



Metropolis-Hastings algorithm

Example: Gamma density

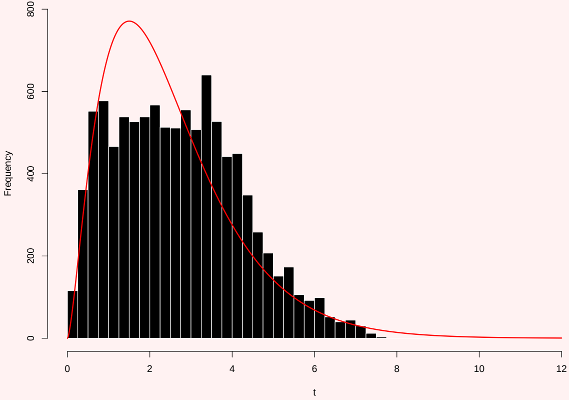
200 steps starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 0.1$.



Metropolis-Hastings algorithm

Example: Gamma density

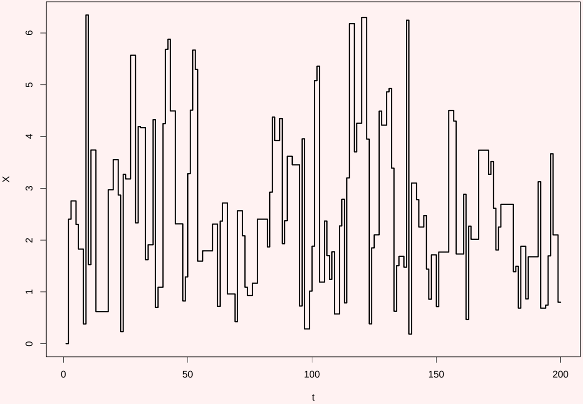
Histogram of states visited over $N = 10000$ steps



Metropolis-Hastings algorithm

Example: Gamma density

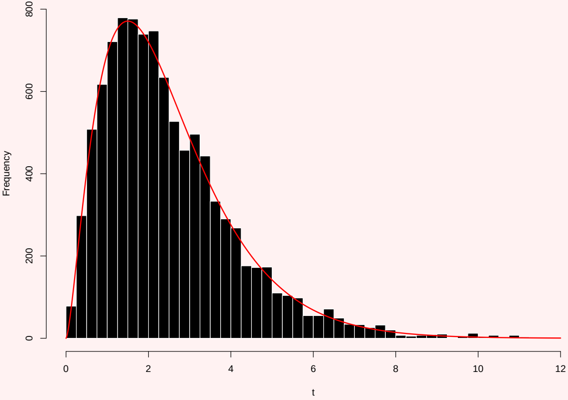
200 steps starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 3$.



Metropolis-Hastings algorithm

Example: Gamma density

Histogram of states visited over $N = 10000$ steps



Metropolis-Hastings algorithm

The states generated by the Markov chain are correlated; we can assess the performance of the Markov chain by examining the sample autocorrelation function

$$r(k) = \left(\frac{N-1}{N-k-1}\right) \frac{\sum\limits_{t=k+1}^{N} (x_t - \overline{x})(x_{t-k} - \overline{x})}{\sum\limits_{t=1}^{N} (x_t - \overline{x})^2}$$

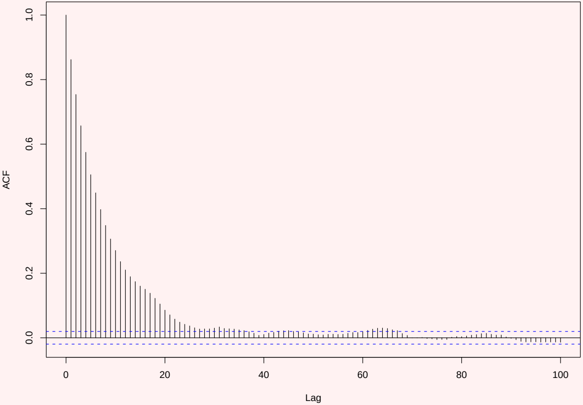
for $k = 0, 1, 2 \ldots$.

A chain with high autocorrelation for large k is typically slow to converge.

Metropolis-Hastings algorithm

Example: Gamma density

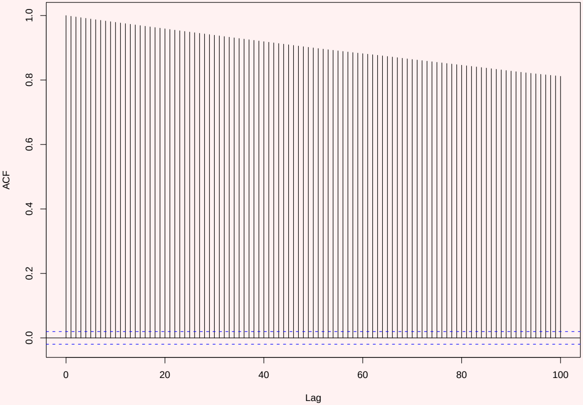
Autocorrelation function for $\sigma_q = 1$.



Metropolis-Hastings algorithm

Example: Gamma density

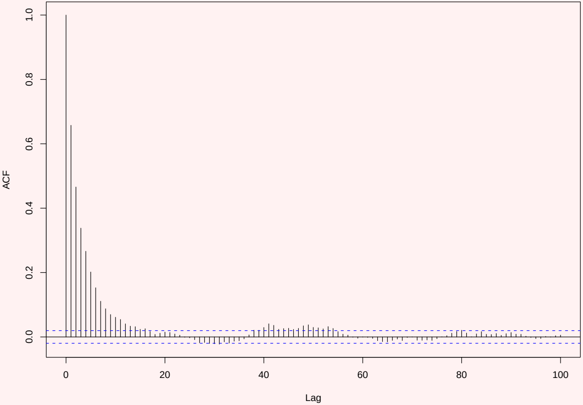
Autocorrelation function for $\sigma_q = 0.1$: inferior performance.



Metropolis-Hastings algorithm

Example: Gamma density

Autocorrelation function for $\sigma_q = 3$: superior performance.



Metropolis-Hastings algorithm

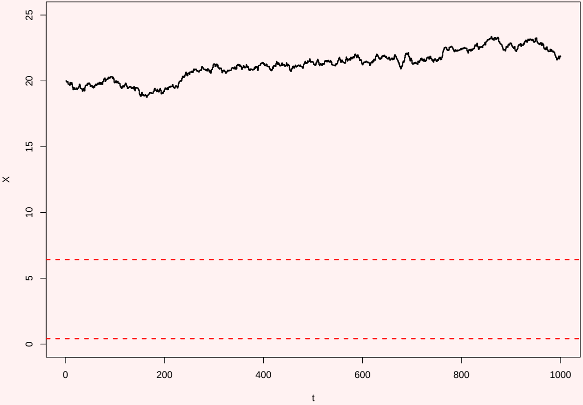
If the chain is started away from the high-probability region of π, then the it can take many steps to return there.

In the following trace plots, the red dashed lines give the 0.025 and 0.975 quantiles of the $\mathit{Gamma}(2.5, 1)$ distribution from the example. The chain is initialized at $X_0 = 20$, and then run for 20000 steps.

Metropolis-Hastings algorithm

Example: Gamma density

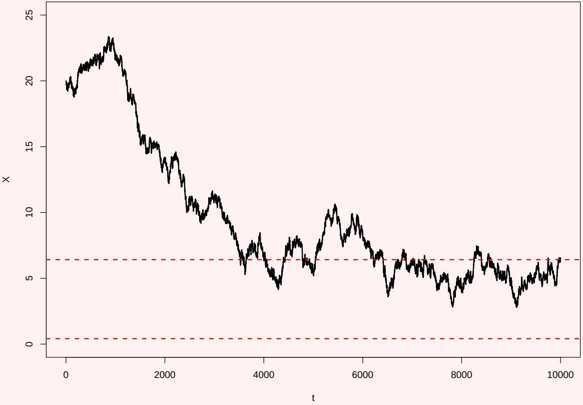
Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 1000 steps.



Metropolis-Hastings algorithm

Example: Gamma density

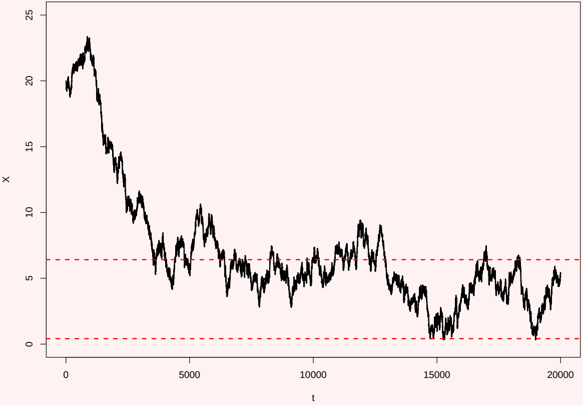
Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 10000 steps.



Metropolis-Hastings algorithm

Example: Gamma density

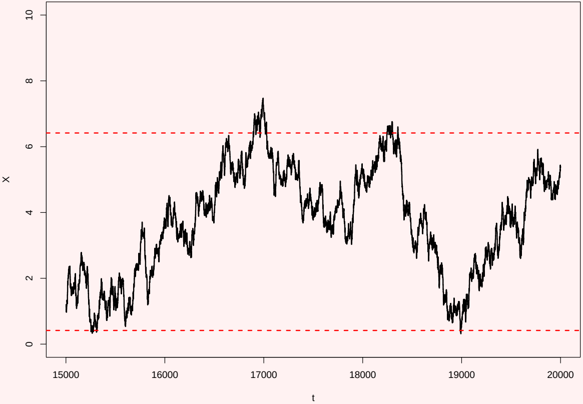
Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 20000 steps.



Metropolis-Hastings algorithm

Example: Gamma density

Starting value $X_0 = 20$, $\sigma_q = 0.1$: steps 15000 to 20000.



Metropolis-Hastings algorithm

In this final section of the chain, the generated values appear to oscillate, despite the fact that the chain has apparently reached the stationary phase. This oscillation is a result of the high autocorrelation present in the chain.

It is often difficult to distinguish such high autocorrelation from the case where a chain has not converged.

The high autocorrelation results here from the choice $\sigma_q = 0.1$; this value is smaller than is optimal.

Metropolis-Hastings algorithm

Example:

See `knitr 7`

Gibbs Sampler

The Metropolis-Hastings algorithm is valid for univariate and multivariate probability distributions, but is more complicated in high dimensions.

The objective is to choose a transition density q that moves around the space $\mathbb{X}$ quickly, which means that we wish to have the acceptance probability reasonably large.

In high dimensions, this is often difficult to achieve. The *Gibbs Sampler* algorithm attempts to solve this problem by breaking down a high-dimensional problem into several lower dimensional problems that are solved iteratively.

Gibbs Sampler

Suppose that π is a probability density in K dimensions, and let the variables be denoted $(X_1, ..., X_K)$. Define the conditional density $\pi_k(.|.)$ for

$$X_k | X_1, ..., X_{k-1}, X_{k+1}, ...X_K$$

by

$$\pi_k \left(x_k ; x_{(k)} \right) = \frac{\pi \left(x_1, ..., x_K \right)}{\pi \left(x_1, ..., x_{k-1}, x_{k+1}, ...x_K \right)} \propto \pi \left(x_1, ..., x_K \right)$$

where the denominator is the marginal distribution of $X_{(k)}$, the $K - 1$ variables excluding X_k.

Gibbs Sampler

The Gibbs Sampler is implemented as follows:

1. Set starting values for the K variables $\left(x_{10},\ldots,x_{K0}\right)$.
2. Sample the conditional distributions with updating:
   (a) sample x_{11} from $\pi_1\left(x_1;x_{20},x_{30},\ldots,x_{K0}\right)$
   (b) sample x_{21} from $\pi_2\left(x_2;x_{11},x_{30},\ldots,x_{K0}\right)$
   (c) sample x_{31} from $\pi_3\left(x_3;x_{11},x_{21},\ldots,x_{K0}\right)$
   $\vdots$
   (K) sample x_{K1} from $\pi_K\left(x_K;x_{11},x_{21},\ldots,x_{K-1\ 1}\right)$

   This completes *one* step of the Gibbs sampler.
3. Return to 2 (a), and repeat to obtain, at step t, the sampled variates $\left(x_{1t},x_{2t},\ldots,x_{Kt}\right)$

Gibbs Sampler

Each of the updates can be achieved using direct sampling from the conditional distribution or individual MH steps, with acceptance probabilities

$$\alpha_k(x,z) = \min\left\{1, \frac{\pi_k\left(z; x_{(k)}\right)}{\pi_k\left(x; x_{(k)}\right)} \frac{q_k(z,x)}{q_k(x,z)}\right\}$$

for $k = 1, \ldots, K$. In 2., the steps can also be completed in *random order*

The steps can be achieved with the scalar variables X_1,...,X_K or with these components as vector variables; deciding on which *blocks* of variables to update simultaneously is often a key issue.

Gibbs Sampler as Metropolis-Hastings

The Gibbs sampler is a special case of the MH algorithm: we can regard the individual updates in 2. as implementing K separate transition kernels that act on the components of $\mathbf{X}$; note that these kernels in isolation yield *reducible* Markov chains.

A more general form of MH algorithm is based on a *mixture* transition kernel

$$P(x, B) = \sum_j \omega_j P_j(x, B)$$

where $0 \leqslant \omega_j \leqslant 1$ and $\sum\limits_j \omega_j = 1$, and the P_j are themselves transition kernels. This allows for the possibility of choosing several proposal densities q_j.

Gibbs Sampler: example

Example: Bivariate Normal

Suppose $\mathbf{X} = (X_1, X_2)^\top \sim \mathcal{N}_2(\mathbf{0}, \Sigma)$ where

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

The general result for multivariate normal distribution conditional distributions is that if $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)^\top \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$, where $\mathbf{X}_1$ is $(d_1 \times 1)$, and

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

then

$$\mathbf{X}_1 | \mathbf{X}_2 = \mathbf{x}_2 \sim \mathcal{N}_{d_1}\left(\boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

Gibbs Sampler: example

Example: Bivariate Normal

Therefore, here $d = 2$, and

$$\begin{aligned} X_1|X_2 = x_2 &\sim \mathcal{N}(\rho x_2, (1-\rho^2)) \\ X_2|X_1 = x_1 &\sim \mathcal{N}(\rho x_1, (1-\rho^2)) \end{aligned}$$

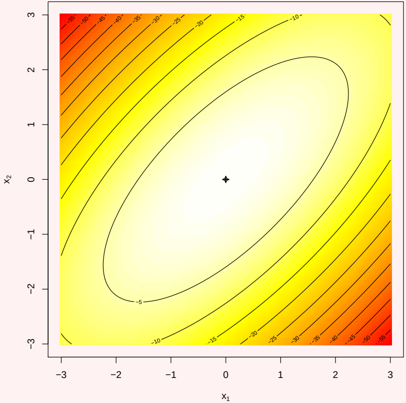
These distributions are sampled repeatedly with updating of the conditioning value after each sampling.

Suppose we start at $(x_{10}, x_{20}) = (0, 0)$.

Gibbs Sampler: example

Example: Bivariate Normal

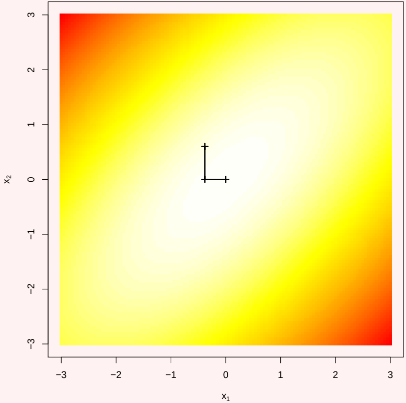
Initial point: (x_{10}, x_{20})



Gibbs Sampler: example

Example: Bivariate Normal

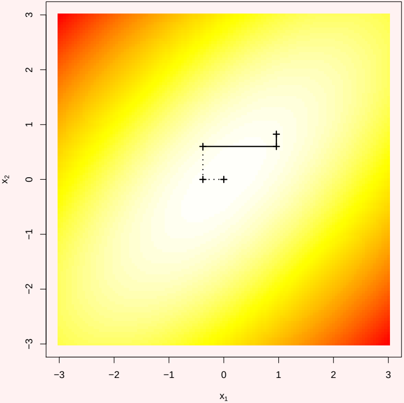
After one update: (x_{11}, x_{21})



Gibbs Sampler: example

Example: Bivariate Normal

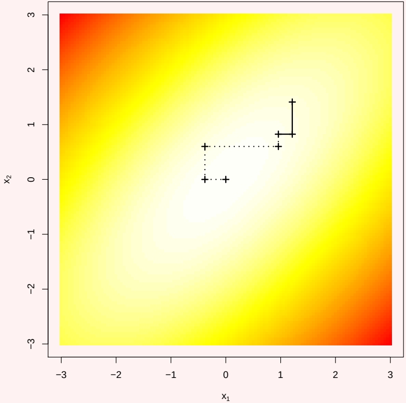
After two updates: (x_{12}, x_{22})



Gibbs Sampler: example

Example: Bivariate Normal

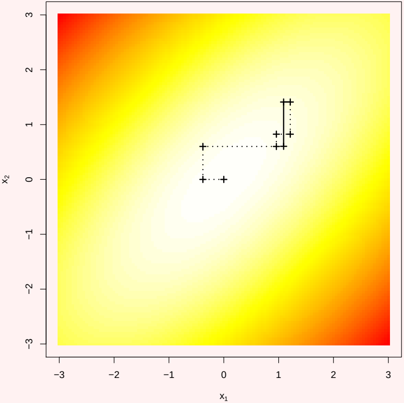
After three updates: (x_{13}, x_{23})



Gibbs Sampler: example

Example: Bivariate Normal

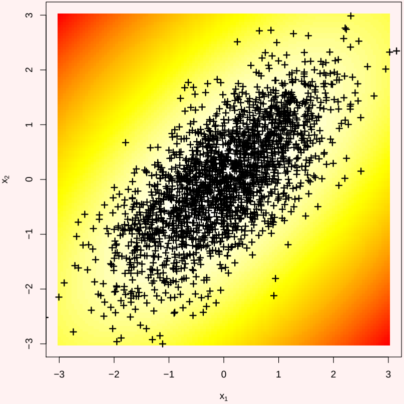
After four updates: (x_{14}, x_{24})



Gibbs Sampler: example

Example: Bivariate Normal

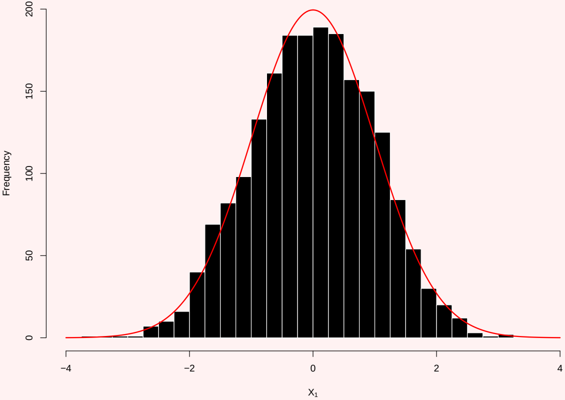
After 2000 updates: entire collected sample



Gibbs Sampler: example

Example: Bivariate Normal

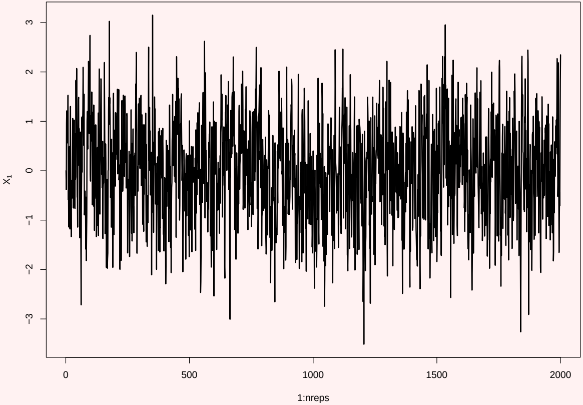
Histogram for X_1 with true marginal density (solid):



Gibbs Sampler: example

Example: Bivariate Normal

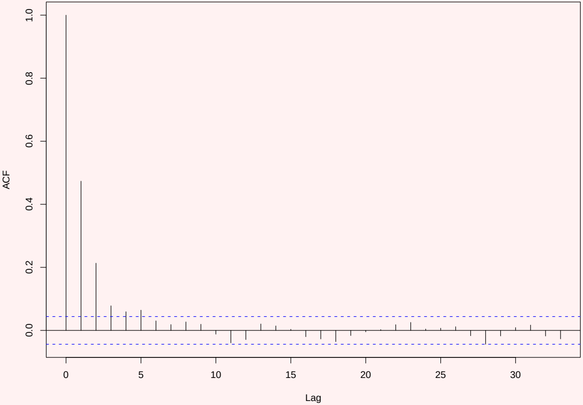
Trace for X_1:



Gibbs Sampler: example

Example: Bivariate Normal

ACF for X_1:



Gibbs Sampler: example

Example: Bivariate Normal

We note that

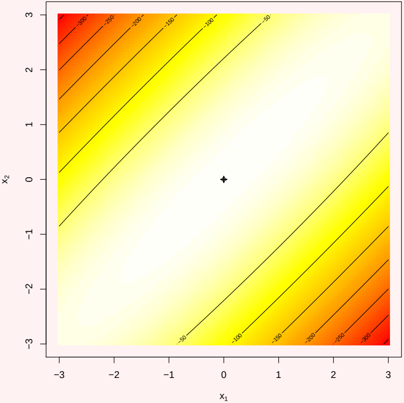
- the Gibbs sampler makes one-step moves along the coordinate axes
- the moves can traverse the support of the joint density fairly well
- there is no “tuning" of a proposal parameter (like σ_q)
- the samples of x_1 that are collected across steps are (dependent) samples from the correct marginal distribution for X_1; the same result holds for X_2

We can re-run the Gibbs sampler from the same starting value, but now with $\rho = 0.95$.

Gibbs Sampler: example

Example: Bivariate Normal

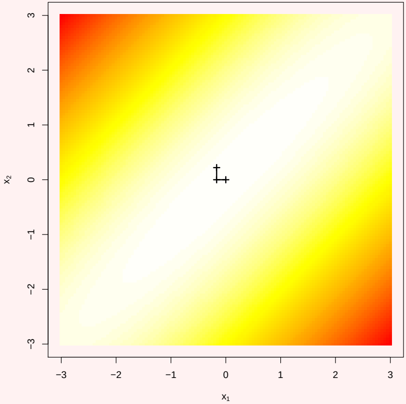
Initial point: (x_{10}, x_{20})



Gibbs Sampler: example

Example: Bivariate Normal

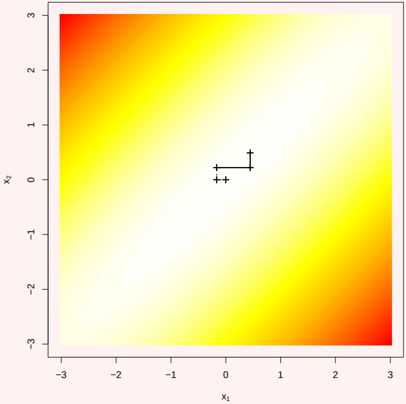
After one update: (x_{11}, x_{21})



Gibbs Sampler: example

Example: Bivariate Normal

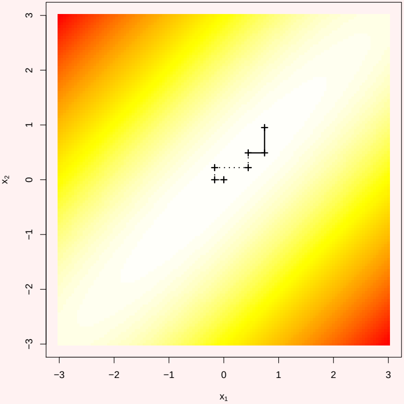
After two updates: (x_{12}, x_{22})



Gibbs Sampler: example

Example: Bivariate Normal

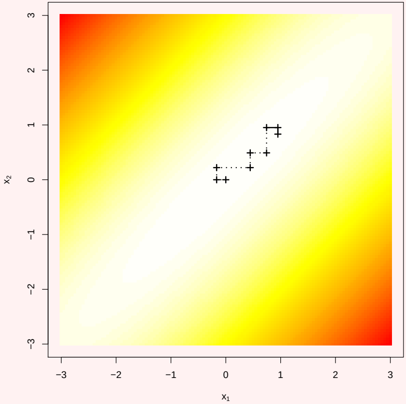
After three updates: (x_{13}, x_{23})



Gibbs Sampler: example

Example: Bivariate Normal

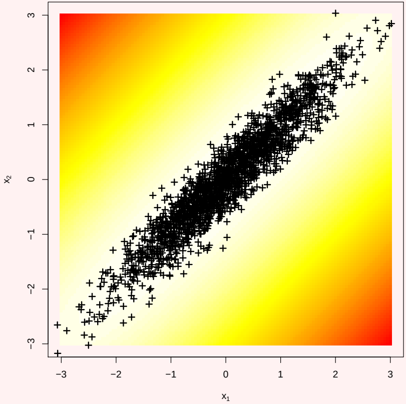
After four updates: (x_{14}, x_{24})



Gibbs Sampler: example

Example: Bivariate Normal

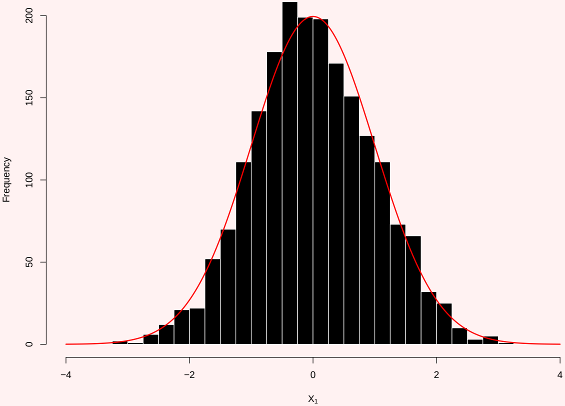
After 2000 updates: entire collected sample



Gibbs Sampler: example

Example: Bivariate Normal

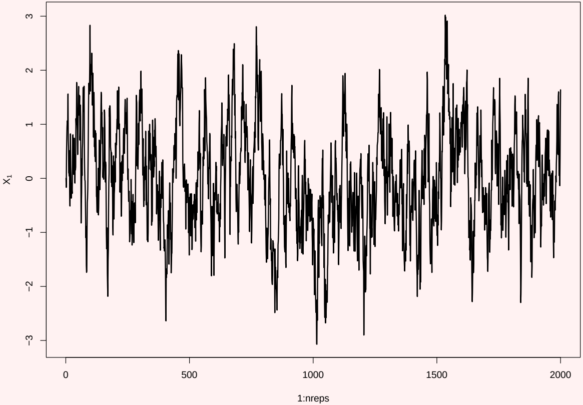
Histogram for X_1 with true marginal density (solid):



Gibbs Sampler: example

Example: Bivariate Normal

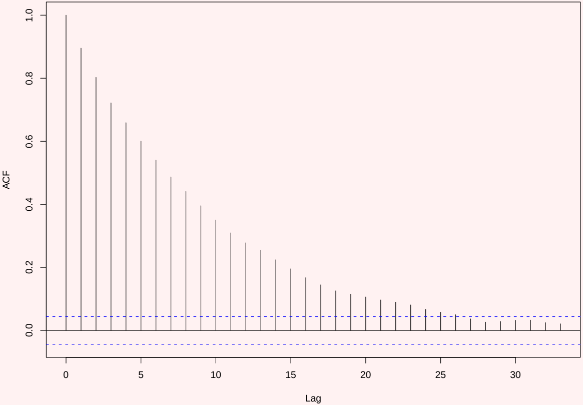
Trace for X_1:



Gibbs Sampler: example

Example: Bivariate Normal

ACF for X_1:



Gibbs Sampler: example

Example: Bivariate Normal

With $\rho = 0.95$, the Gibbs sampler still performs adequately, but the moves made are smaller, and exploring the distribution is much more difficult.

This illustrates a potential general problem with the Gibbs sampler: although it is straightforward to implement as it involves only sampling variates from univariate densities, the restriction to moves along the coordinate axes can cause problems if the variables are *highly correlated*.

Gibbs Sampler: example

Example: Weibull posterior distribution

Suppose that $Y_1, \ldots, Y_n$ are conditionally iid from the Weibull distribution with density

$$f_Y(y; \gamma, \lambda) = \gamma \lambda y^{\gamma-1} \exp\{-\lambda y^\gamma\} \qquad y > 0$$

and zero otherwise, for parameters $\gamma, \lambda > 0$. We seek to perform Bayesian inference for the two unknown parameters.

The likelihood for data $y_1, \ldots, y_n$ takes the form

$$\mathcal{L}_n(\gamma, \lambda) = \gamma^n \lambda^n \left(\prod_{i=1}^{n} y_i \right)^{\gamma-1} \exp\left\{ -\lambda \sum_{i=1}^{n} y_i^\gamma \right\}$$

Gibbs Sampler: example

Example: Weibull posterior distribution

We assume independent *Exponential*(0.1) priors for γ and λ

$$\pi_0(\gamma, \lambda) = 0.01 e^{-0.1(\gamma+\lambda)} \qquad \gamma, \lambda > 0.$$

This yields the posterior distribution up to proportionality as

$$\pi_n(\gamma, \lambda) \propto \mathcal{L}_n(\gamma, \lambda) \pi_0(\gamma, \lambda) \qquad \gamma, \lambda > 0$$

which is a non-standard distribution.

Gibbs Sampler: example

Example: Weibull posterior distribution

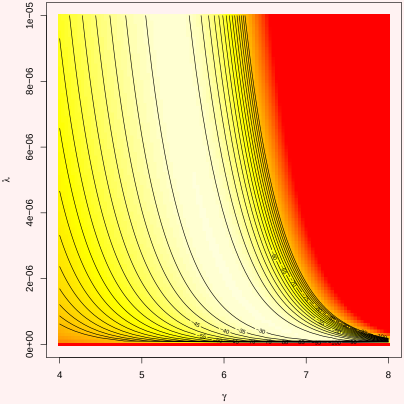
We seek to produce a sample from this joint posterior distribution for data $(n = 15)$.

10.3959	6.2281	6.5331	10.7086	7.6138
8.9423	8.8254	6.1461	7.2988	8.8081
7.5316	8.2238	8.9831	6.4174	9.7648

Gibbs Sampler: example

Example: Weibull example

Joint posterior (up to proportionality)



Gibbs Sampler: example

Example: Weibull example

In this case, neither full conditional posterior

$$\pi_n(\gamma|\lambda) \qquad\qquad \pi_n(\lambda|\gamma)$$

is a standard distribution, and cannot be sampled from easily.

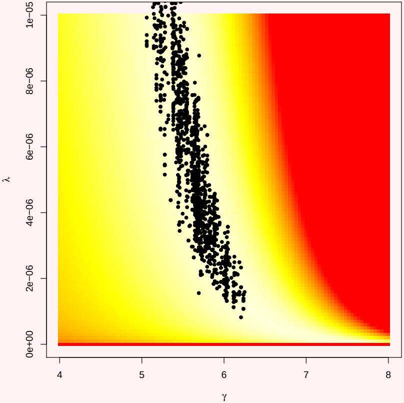
We adopt a *Metropolis-within-Gibbs* strategy; this uses MH accept-reject steps for each parameter and its full conditional.

Specifically, as both parameters are positive, we use the reflected normal proposal distribution from the previous example with $\sigma_q = 1$ for γ and $\sigma_q = 10^{-3}$ for λ proposals.

Gibbs Sampler: example

Example: Weibull example

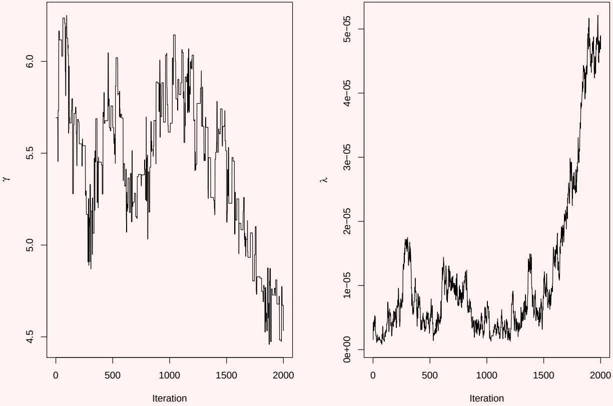
First 2000 Gibbs sampler steps.



Gibbs Sampler: example

Example: Weibull example

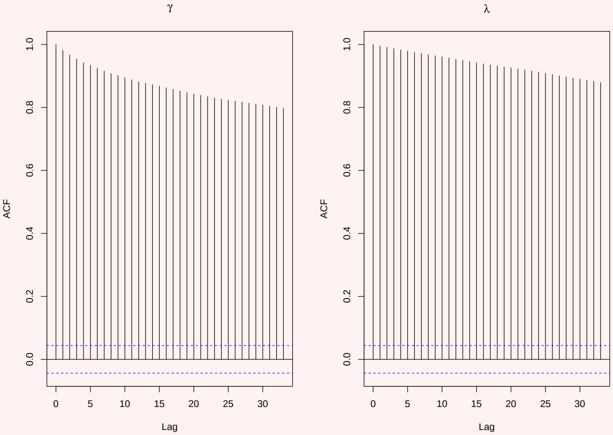
Trace plots:



Gibbs Sampler: example

Example: Weibull example

Acf:



Gibbs Sampler: example

Example: Weibull example

The posterior correlation here is approximately 0.86; this severely affects the performance of the Gibbs sampler.

A reparameterization helps with this problem: define ϕ by

$$\phi = \left(\frac{1}{\lambda}\right)^{1/\gamma} \qquad \therefore \qquad \lambda = \left(\frac{1}{\phi}\right)^{\gamma}$$

For this new parameterization, we must remember to include the Jacobian in the prior for the new parameters

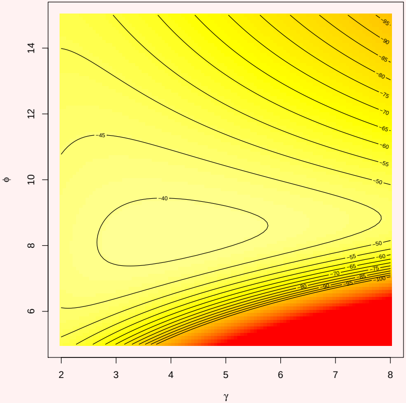
$$\pi_0(\gamma, \phi) = \pi_0(\gamma, \lambda(\gamma, \phi))|J(\gamma, \phi)|$$

We again must use Metropolis-within-Gibbs.

Gibbs Sampler: example

Example: Weibull example

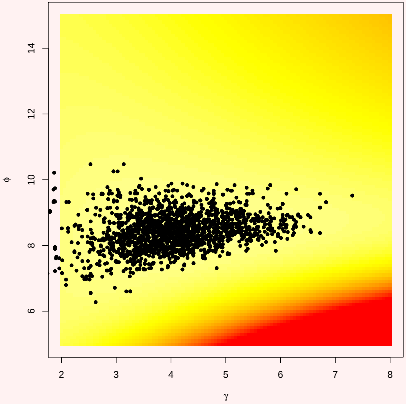
Joint posterior for new parameterization



Gibbs Sampler: example

Example: Weibull example

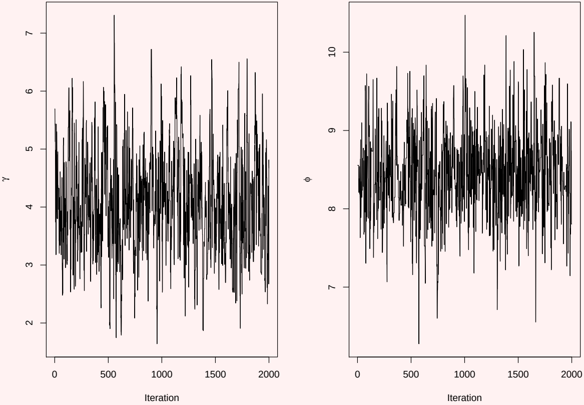
Sample from joint posterior for new parameterization



Gibbs Sampler: example

Example: Weibull example

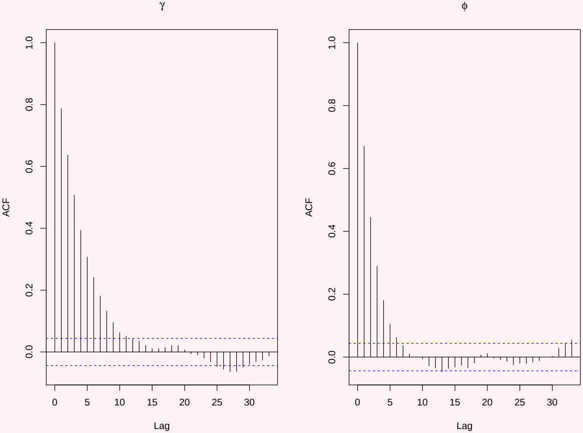
Trace plots: new parameterization



Gibbs Sampler: example

Example: Weibull example

Acf: new parameterization



Gibbs Sampler: example

Example: Weibull example

The posterior correlation here is approximately 0.25, and the Gibbs sampler can effectively traverse the parameter space.

Parameter estimates for the new parameters can be obtained from the posterior samples: the mean and 95% credible interval for each parameter is

$$\begin{aligned} \gamma &: \quad 4.079\,(2.455, 6.038) \\ \phi &: \quad 8.446\,(7.310, 9.593) \end{aligned}$$

It is also possible to obtain posterior summaries for other functions of the posterior parameters.

Gibbs Sampler: example

Example: Weibull example

For example, the survivor function, $S(y)$, is defined by

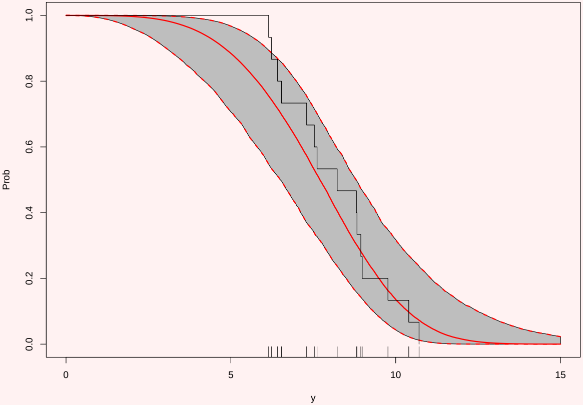
$$S(y) = \Pr[Y > y] = \exp\{-(y/\phi)^{\gamma}\}.$$

For each $y \in \mathbb{R}^{+}$, we can compute this function for each pair of generated points (γ, ϕ) obtained from the Gibbs sampler. We can then compute the pointwise credible intervals.

Gibbs Sampler: example

Example: Weibull example

Posterior survivor function: Bayes estimate and 95 % credible interval (shaded). Solid line is empirical survivor function.



Effective sample size

We seek to measure the adequacy of the collected samples for estimating parameters, in particular, we wish to assess the variance of the estimators.

For an iid sample of size N, the variance of the Monte Carlo estimator $\widehat{I}_N(g)$ is

$$\frac{\mathrm{Var}[g(X)]}{N}.$$

However, for a *dependent* sample, the variance is

$$\frac{\mathrm{Var}[g(X)]}{N_{\mathrm{eff}}}$$

where N_{eff} is the *effective sample size*.

Effective sample size

We have that, for a series of N observations from a dependent stochastic process, the effective sample size is given by

$$N_{\text{eff}} = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho(k)}$$

where $\rho(k)$ is the true lag-k autocorrelation for the Markov chain. The denominator is the *integrated autocorrelation time*.

The true autocorrelations are typically not known, so must be estimated from the data. Most typically this is achieved using spectral methods. The calculation is available in `R` from the library `coda`, in the function

`effectiveSize`.

Effective sample size

For the Weibull example, with $N = 2000$ in runs shown above

	N_{eff}	
	γ	λ
(γ, λ) parameterization	8.46	4.14
(γ, ϕ) parameterization	217.20	393.69

From this we can tell that the second MCMC run, in the (γ, ϕ) parameterization, has larger effective sample sizes.

Rejection sampling for the Weibull example

Note that we could attempt to address the problem of sampling from the posterior distribution in the Weibull problem above by *rejection sampling*. In the (γ, ϕ) parameterization, we have

$$\pi_n(\gamma, \phi) \ \propto \ \mathcal{L}_n(\gamma, \phi)\pi_0(\gamma, \lambda(\gamma, \phi))|J(\gamma, \phi)|$$

where $\pi_0(.,.)$ is the product of independent *Exponential*(0.01) priors.

Rejection sampling for the Weibull example

We use proposal function f_0 which is the product of *Gamma* densities; we choose

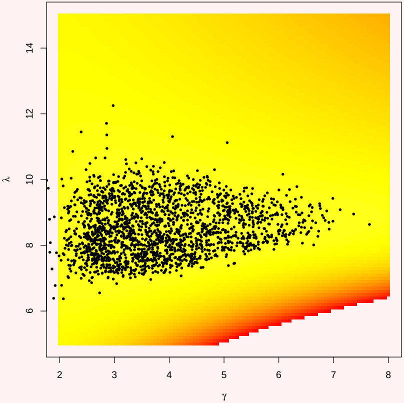
$$\mathit{Gamma}(2, 1/2) \qquad \mathit{Gamma}(4, 2)$$

for proposing (γ, ϕ), and then use numerical maximization to find the bound M.

Rejection sampling for the Weibull example

Example: Weibull example

Rejection sampling: acceptance rate is approximately 0.116.



Examples

Example: Weibull

See `knitr 8`

Example: Non-linear regression

See `knitr 9`

Metropolis-Hastings in higher dimensions

The MH algorithm can be used for probability distributions in arbitrary dimension. Note that it is always the case that, for the full conditional distributions, if

- $\mathbf{x}_1$ is a sub-vector of the entire vector $\mathbf{x}$ of variables, and
- $\mathbf{x}_{(1)}$ is $\mathbf{x}$ with the components $\mathbf{x}_1$ removed,

then

$$\pi(\mathbf{x}_1|\mathbf{x}_{(1)}) \;\propto\; \pi(\mathbf{x})$$

as the normalizing constant $\pi(\mathbf{x}_{(1)})$ does not depend on $\mathbf{x}_1$.

Knowing $\pi(\mathbf{x}_1|\mathbf{x}_{(1)})$ up to proportionality is sufficient.

Examples

Example: Auxiliary variable methods

See `knitr 10`

Example: Missing data problems

See `knitr 11`

Examples

Example: Multi-level models

See `knitr 12`

Example: Hierarchical linear regression

See `knitr 13`

Example: Hierarchical non-linear regression

See `knitr 14`

Bayesian nonparametrics

Bayesian non-parametric inference

In Bayesian statistical inference, we compute and summarize the posterior distribution of the unknown parameters in the probability model, in light of observed data.

- In Bayesian *parametric* inference, the parameter is the usual θ, λ, μ say that appears in the presumed (conditional) data generating model.
- In Bayesian *non-parametric* inference, the parameter is the *distribution* from which the data are drawn.

As part of Bayesian inference, we need to specify the *prior* probability distribution for these parameters.

Random Discrete Distributions

In the *discrete* univariate case, a pmf is constructed by

(I) choosing some locations $x_1, x_2, x_3, \ldots$ on the real line

(II) placing a probability mass $p_1, p_2, p_3, \ldots$ at the locations, where the probabilities sum to 1.

Then the function

$$f(x) = \sum_{i=1}^{\infty} p_i \delta_{x_i}(x)$$

is a discrete probability distribution on the set $\{x_1, x_2, x_3, \ldots\}$.

Random Discrete Distributions

How do we make this random ?

(I) Choose the locations *randomly* (and independently) from some distribution G_X; denote them $x_1, x_2, x_3, \ldots$.

(II) Choose the probabilities *randomly* in such a way such that they sum to 1; denote them $\pi_1, \pi_2, \pi_3, \ldots$.

Then

$$\widetilde{f}(x) = \sum_{i=1}^{\infty} \pi_i \delta_{x_i}(x)$$

is a random mass function.

Random Discrete Distributions

If the number of locations is *finite*, equal to K say, then we can generate the probabilities $\pi_1, \ldots, \pi_K$ from a *Dirichlet* distribution

$$\textit{Dirichlet}(K; \alpha_1, \ldots, \alpha_{K+1})$$

where the αs are fixed constants.

In this case, the x_is are fixed, or the x_is simulated from G_X.

Random Discrete Distributions

If the number of locations is *infinite*, it is helpful to order the π_i in descending order

$$\pi_1 \geqslant \pi_2 \geqslant \pi_3 \geqslant \cdots$$

so that the terms in the infinite sum become negligible, so that the truncation

$$\widetilde{f}(x) \bumpeq \widetilde{f}_N(x) = \sum_{i=1}^{N} \pi_i \delta_{x_i}(x)$$

can be computed.

Random Discrete Distributions

For example, consider for $\alpha > 0$

$$V_1, V_2, V_3, \ldots \sim Beta(1, \alpha)$$

independently distributed, and

$$\begin{aligned}
\pi_1 &= V_1 \\
\pi_2 &= (1 - V_1)V_2 \\
\pi_3 &= (1 - V_1)(1 - V_2)V_3 \\
\vdots & \;\vdots\; \vdots
\end{aligned}$$

so that the π_i sequence are *decreasing in expectation*.

Random Discrete Distributions

- Two hyperparameters α, G_X.
- α small gives a few large masses
- α large gives many small masses
- α large reproduces a distribution much like G_X

Thus α is like a *precision* parameter, G_X is like a *location* parameter for the distribution.

Random Discrete Distributions

The method described above is the *stick-breaking* construction of the *Dirichlet Process* with parameters (α, G_X).

$$\widetilde{f} \sim DP(\alpha, G_X)$$

For any partition of $\mathbb{R}$ into disjoint subsets $B_1, B_2, \ldots, B_K, B_{K+1}$ the Dirichlet process assigns random probabilities

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_K, p_{K+1})^\top$$

to the subsets, where

$$\boldsymbol{p} \sim \mathit{Dirichlet}(K; \alpha_1, \alpha_2, \ldots, \alpha_K, \alpha_{K+1})$$

and $\alpha_k = \alpha G_X(B_k)$ for each k

Posterior Calculation

Suppose *a priori* $\widetilde{f} \sim DP(\alpha, G_X)$. What is the posterior if data $y_1, y_2, \ldots, y_n$ are independent draws from $\widetilde{f}$?

We have a *conjugate* model: *a posteriori*

$$\widetilde{f} \sim DP(\alpha^\star, G_X^\star)$$

where

$$\begin{aligned} \alpha^\star &= \alpha + n \\ G_X^\star &= \frac{\alpha G_X + \sum_{j=1}^{n} \delta_{y_j}}{\alpha + n} \end{aligned}$$

Posterior Calculation

For any partition of $\mathbb{R}$ into disjoint subsets $B_1, B_2, \ldots, B_K, B_{K+1}$ with associated probabilities

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_K, p_{K+1})^\top$$

- in the *prior*

  $$\boldsymbol{p} \sim \mathit{Dirichlet}(K; \alpha_1, \alpha_2, \ldots, \alpha_K, \alpha_{K+1})$$

  and $\alpha_k = \alpha G_X(B_k)$.

- in the *posterior*

  $$\boldsymbol{p} \sim \mathit{Dirichlet}(K; \alpha_1^\star, \alpha_2^\star, \ldots, \alpha_K^\star, \alpha_{K+1}^\star)$$

  and $\alpha_k^\star = \alpha^\star G_X^\star(B_k)$.

Posterior Calculation

It follows that

$$\alpha_k^\star = \alpha^\star G_X^\star(B_k) = G_X(B_k) + n_k$$

where

$$n_k = \text{Number of } y_j \text{ in the set } B_k.$$

Therefore we have the usual kind of Bayesian updating rule.

Also

- α small: posterior looks like empirical cdf
- α large: posterior looks like G_X.

So, overall, things proceed much like parametric inference !

Dirichlet Process: Some issues

1. How do we report the inference ?
    - Which partition, that is, which sets $B_1, B_2, \ldots, B_K, B_{K+1}$ should we choose to report the posterior ?
2. In this formulation, the posterior can only support *discrete* distributions
    - that is, any estimate of the true f obtained from the model is almost surely a discrete distribution.

We solve 1. by using *simulation* methods.

We solve 2. by *extending the model*.

Simulating from a Dirichlet Process Model

To sample data from $DP(\alpha, G_X)$: sample $Z_1 \sim G_X$, and for $j = 2, \ldots, n$, sample

$$Z_j | Z_1, Z_2, \ldots, Z_{j-1} \sim \frac{\alpha}{\alpha + j - 1} G_X(\cdot) + \frac{1}{\alpha + j - 1} \sum_{l=1}^{j-1} \delta_{Z_l}(\cdot)$$

i.e. sample Z_j from

G_X with probability $\alpha/(\alpha + j - 1)$.

or from the discrete set

$\{Z_1, \ldots, Z_{j-1}\}$ with probability $1/(\alpha + j - 1)$ for each element.

The Dirichlet Process and Clustering

This algorithm is a *Polya Urn* scheme.

It demonstrates that the Dirichlet process model induces a *clustering* mechanism: in the simulated Z sample, we have many identical values due to the sampling of Z_j uniformly on

$$\{Z_1, \ldots, Z_{j-1}\}$$

at each j.

The Dirichlet Process and Clustering

In a sample of n items from a $DP(\alpha, G_X)$ model, the probability of having k clusters is

$$\Pr[K = k] = \frac{\alpha^k B(n,k)}{A_n(\alpha)}$$

where $B(n,k)$ is the *Stirling number of the first kind*.

$$A_n(x) = \sum_{j=1}^{n} B(n,j) x^j$$

The Dirichlet Process and Clustering

The expected number of clusters *a priori* is

$$\sum_{j=1}^{n} \frac{\alpha}{\alpha + j - 1} = \mathrm{O}(\alpha \log n)$$

For $n = 200$:

α	$\mathbb{E}[K]$
0.5	3.631
2.0	9.766
4.0	16.238
10.0	30.930

Extension to The Continuous Case

We add another stage that brings in a continuous distribution.

For example, could treat each x_i as the location of a normal density, and consider generating a y for each

$$\begin{aligned} x_1, x_2, \ldots &\sim G_X \\ \pi_1, \pi_2, \ldots & \quad \text{generated by stick-breaking.} \\ y &\sim \phi((y - x_i)/\sigma) \qquad i = 1, 2, \ldots \end{aligned}$$

Then,

$$\widetilde{f}(y) = \sum_{i=1}^{n} \pi_i \phi((y - x_i)/\sigma)$$

that is, an *infinite mixture model*.

Dirichlet Process Mixture

This construction is called the

Dirichlet Process Mixture (DPM)

with a Normal kernel. Any continuous kernel g_Y can be used in place of ϕ.

Under this model, $\widetilde{f}$ is almost surely *continuous*.

- α small implies less bumpy
- α large implies more bumpy.

Bayesian Inference

Suppose we have the usual de Finetti construction

- a prior model for f that is $DPM(\alpha, G_X, g_Y; \theta)$ where θ are parameters that appear in G_X and g_y.
- conditional on f, data $y_1, y_2, \ldots, y_n \sim f$

We wish to compute the posterior for f. We use the hierarchical formulation

$$\begin{aligned} y_j | x_j &\overset{\text{ind.}}{\sim} g_Y(y_j | x_j, \theta) \qquad j = 1, \ldots, n \\ x_1, \ldots, x_n &\sim DP(\alpha, G_X; \theta) \\ \theta &\sim p(\theta). \end{aligned}$$

Bayesian Inference: Algorithm 1

The latent variables $x_1, \ldots, x_n$ are also treated as parameters. They can be sampled using an MCMC *Gibbs sampler* scheme.

For $j = 1, \ldots, n$, we sample

$$x_j \mid \boldsymbol{x}_{(j)}, \boldsymbol{y} \sim w_0 p(x_j|y_j) + \sum_{l \neq j} w_l \delta_{x_l}$$

where

- $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$
- $\boldsymbol{x}_{(j)} = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)^\top$.
- w_0 is proportional to α times the *prior predictive* of y_j
- w_l is proportional to the *likelihood* of $y_j|x_l$
- $p(x_j|y_j)$ is the *posterior* for x_j given y_j.

Bayesian Inference: Algorithm 2

We can use the *clustering* property: suppose that at a given iteration of the MCMC, there are K clusters labelled 1 to K, where $K \leqslant n$.

Label the K distinct x values

$$z_1, \ldots, z_K$$

and for each j, define the corresponding cluster label c_j where

$$c_j = k \quad \Longleftrightarrow \quad x_j = z_k$$

We can update the c_js instead of the x_js which will be more computationally efficient; we are clustering xs to the *cluster centres* at the z values.

Bayesian Inference: Algorithm 2

For $i = 1, \ldots, n$, let

- $n_1(i), \ldots, n_K(i)$ denote the number of items in clusters $1, \ldots, K$
- $\boldsymbol{y}_1(i), \ldots, \boldsymbol{y}_K(i)$ denote the vectors of y values currently allocated to the K clusters

if the ith data point is removed.

Bayesian Inference: Algorithm 2

For $i = 1, \ldots, n$, we sample the cluster labels in a Gibbs sampler with conditional probabilities

$$\Pr[c_i = k \mid c_{(i)}] \propto \frac{n_k(i)}{n-1+\alpha} p(y_i|\boldsymbol{y}_k(i)) \qquad k = 1, \ldots, K$$

and

$$\Pr[c_i = K+1 \mid c_{(i)}] \propto \frac{\alpha}{n-1+\alpha} p(y_i)$$

Bayesian Inference: Algorithm 2

In this expression

- $p(y_i|\mathbf{y}_k(i))$ is the *posterior predictive* density for y_i in the DPM model, assuming that y_i comes from cluster k.
- $p(y_i)$ is the *prior predictive* density for y_i in the DPM model, assuming that y_i comes from a new cluster not currently represented in the data.

We have *integrated out* the Dirichlet process.

Thus we can simply sample the cluster labels in turn, and then sample the $z_1, \ldots, z_k$ values; this will allow us to do density estimation.

Bayesian Inference: Algorithm 2

By the usual calculation

$$p(y_i|\boldsymbol{y}_k(i)) = \int g_Y(y_i \mid x)p(x \mid \boldsymbol{y}_k(i))\, dx$$

where

$$p(x \mid \boldsymbol{y}_k(i)) \quad \propto \quad p(\boldsymbol{y}_k(i) \mid x)p(x) = \left\{ \prod_{l \neq i} g_Y(y_l \mid x) \right\} p(x)$$

gives the posterior distribution for the kth cluster centre.

Similarly

$$p(y_i) = \int g_Y(y_i \mid x)p(x)\, dx$$

Bayesian Inference: Algorithm 2

In the earlier Gaussian model, suppose for simplicity that G_X is the $N(0, \lambda^2)$ density:

$$\begin{aligned} x_i &\sim N(0, \lambda^2) \\ y_i \mid x_i &\sim N(x_i, \sigma^2) \end{aligned}$$

Then

$$p(y_i|\boldsymbol{y}_k(i)) \equiv N\left(\frac{n_k(i)\overline{y}_k(i)}{n_k(i)/\sigma^2 + 1/\lambda^2}, \frac{(n_k(i)+1)/\sigma^2 + 1/\lambda^2}{n_k(i)/\sigma^2 + 1/\lambda^2}\sigma^2\right)$$

and

$$p(y_i) \equiv N\left(0, \sigma^2 + \lambda^2\right)$$

Extensions

Easy to extend to

- unknown σ^2
- non-Gaussian conjugate models
- blocked Gibbs sampler
- Metropolis-Hastings MCMC for cluster labels
- multivariate conjugate models

Not so easy to do non-conjugate models.

Resampling Approaches to Inference

Resampling methods allow the study of frequentist properties of statistical quantities by producing pseudo-replicate data sets of the same size as the observed data, and examining the statistical variation across these replicate data sets.

Notation: independence case

Suppose $Y_1, \dots, Y_n \sim F$ are a random sample, and let $\theta = \theta(F)$ be the target parameter. For example

$$\theta(F) = \int y \, dF(y) \qquad \text{or} \qquad \theta(F) = \inf_y \{F(y) \geqslant p\}$$

etc. Let $y_1, \dots, y_n$ be the observed data.

If $\widehat{F}_n$ is the empirical cdf,

$$\widehat{F}_n(y) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{[y_i, \infty)}(y) \qquad\qquad d\widehat{F}_n(y) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_i\}}(y)$$

then a natural 'plug-in' estimator of $\theta(F)$ is $T_n = \theta(\widehat{F}_n)$,

Notation: independence case

For each y, under mild regularity conditions

$$\widehat{F}_n(y) \xrightarrow{p} F(y)$$

but also

$$\sup_y |\widehat{F}_n(y) - F(y)| \xrightarrow{p} 0$$

as $n \longrightarrow \infty$. Therefore

$$\theta(\widehat{F}_n) \xrightarrow{p} \theta(F)$$

which justifies (asymptotically) the use of the plug-in estimator.

Notation: independence case

The bias and variance of the estimator are

$$\begin{aligned} b_{T_n}(F) &= \mathbb{E}_F[T_n] - \theta(F) \\ v_{T_n}(F) &= \mathrm{Var}_F[T_n] \end{aligned}$$

both of which depend on the true F.

Notation: independence case

We wish to study these properties of the estimator. In some cases, it is possible to study these quantities analytically. Suppose, however, $\widehat{\theta}$ is the solution of

$$\sum_{i=1}^{n} m(y_i; \theta) = 0$$

for some *m-estimating function*.

The corresponding estimator is not analytically available, so its finite sample properties are hard to study.

The Bootstrap

Suppose we wish to summarize an aspect of the sampling distribution of $T_n = \theta(\widehat{F}_n)$. Let

$$s(F) \equiv s(T_n; F)$$

denote the statistical summary of interest; it is written as a function of F as the statistical properties of T_n are entirely dictated by F.

The Bootstrap

The quantity $s(F)$ can again usually be expressed in terms of an integral with respect to F

$$s(F) = \int s(t(\mathbf{y}))dF(\mathbf{y}).$$

for function $t(.)$ that defines the estimator. Occasionally, this expression can be computed analytically.

The Bootstrap

The key idea of the *bootstrap* is to replace calculations wrt F by calculations wrt $\widehat{F}_n$, and to compute

$$s(\widehat{F}_n)$$

numerically, that is

$$s(\widehat{F}_n) = \int s(t(\mathbf{y}))d\widehat{F}_n(\mathbf{y}).$$

The Bootstrap

For a random sample, $y_1, \ldots, y_n$, the bootstrap proceeds as follows:

1. Set B (the number of bootstrap resamples)
2. For $b = 1, \ldots, B$,
   (a) generate a sample of size n $y_1^{(b)}, \ldots, y_n^{(b)}$ at random *with replacement* from $\widehat{F}_n$
   (b) form the statistic of interest $t_n^{(b)}$
3. Summarize the resampled estimates

$$t_n^{(1)}, \ldots, t_n^{(B)}$$

using the desired statistical summary, $s(.)$.

The Bootstrap is a Bayesian procedure

Recall Bayesian inference based on the Dirichlet process:

- $Y_1, \ldots, Y_n \sim \widetilde{f}$ (conditionally independent)
- *a priori* $\widetilde{f} \sim DP(\alpha, G_Y)$.
  - $\alpha > 0$
  - $G_Y(.)$ some distribution on $\mathbb{R}$
- *a posteriori*

$$\widetilde{f} \sim DP(\alpha^\star, G_Y^\star)$$

where

$$\alpha^\star = \alpha + n \qquad\qquad G_Y^\star = \frac{\alpha G_Y + \sum_{j=1}^{n} \delta_{y_j}}{\alpha + n}$$

The Bootstrap is a Bayesian procedure

The Dirichlet Process is a distribution on distributions that are discrete (with probability 1), that is,

- mass function of the form

$$\widetilde{f}(y) = \sum_{i=1}^{\infty} \pi_i \delta_{Y_i}(y)$$

- random locations $Y_1, Y_2, \ldots \sim G_Y$;
- random probabilities $\pi_1, \pi_2, \ldots$ constructed according to the stick-breaking mechanism with parameter α.

The Bootstrap is a Bayesian procedure

We can easily produce an iid sample $Y_1, Y_2, \ldots, \sim \widetilde{f}(.)$ as it is merely a discrete distribution: this construction ensures that $\{Y_n\}$ is an *exchangeable* sequence by de Finetti's theorem.

- $\widetilde{f} \sim DP(\alpha, G_Y)$
- $Y_1, Y_2, \ldots, Y_n \mid \widetilde{f} \sim \widetilde{f}$, independently.

The Bootstrap is a Bayesian procedure

In light of observed data $y_1, \ldots, y_n$, if the prior is $DP(\alpha, G_Y)$, then the posterior is

$$DP\left(\alpha + n, \frac{\alpha G_Y + \sum_{i=1}^{n} \delta_{y_i}(.)}{\alpha + n}\right).$$

Denote the posterior parameters where

$$\begin{aligned} \alpha^\star &= \alpha + n \\ G_Y^\star &= \frac{\alpha G_Y + \sum_{j=1}^{n} \delta_{y_j}(.)}{\alpha + n} \end{aligned}$$

The Bootstrap is a Bayesian procedure

Recall the predictive calculation given by de Finetti:

$$p_n(y_{(n+1):(n+m)}) = \int \prod_{i=n+1}^{n+m} f(y_i)\ \pi_n(df).$$

To sample from p_n, we

- sample $f \sim \pi_n$;
- sample $Y_{n+1}, \ldots, Y_{n+m}$ independently from f.

We sample a random f from π_n, and then obtain a sample $Y_{n+1}, \ldots, Y_{n+m}$ from the predictive distribution using this f.

This may be achieved using the Polya urn.

The Bootstrap is a Bayesian procedure

The posterior mean of the DP is the measure

$$\frac{\alpha G_Y + \sum_{i=1}^{n} \delta_{y_i}(.)}{\alpha + n}.$$

Instead of using the full Polya urn scheme, consider a *plug-in* procedure that replaces a sample of f by this posterior mean.

Independently for $j = n+1, \ldots, n+m$,

- w.p. $\alpha/(\alpha+n)$: draw from G_Y;
- w.p. $1/(\alpha+n)$: draw y_i, for each $i = 1, \ldots, n$.

The Bootstrap is a Bayesian procedure

In the limit as $\alpha \longrightarrow 0$, this procedure becomes

- sample y_i w.p. $1/n$, $i = 1, \ldots, n$

independently for $j = n+1, \ldots, n+m$. This is identical to the bootstrap.

Therefore bootstrap calculations are Monte Carlo calculations made with respect to the *predictive distribution* computed for a Dirichlet process prior and posterior, *in the limit as* $\alpha \longrightarrow 0$, using a *plug-in* approach.

Predictive distributions

Recall the frequentist justification of maximum likelihood: in a potentially mis-specified model $f(y;\theta)$, we identify the true value of θ, θ_0 as

$$\theta_0 = \arg\min_{\theta} KL(f_0, f(.;\theta)) = \arg\min_{\theta} \int \log\left\{\frac{f_0(y)}{f(y;\theta)}\right\} f_0(y)\, dy$$

Predictive distributions

The corresponding estimator is obtained when we replace the integral by a ‘Monte Carlo’ version based on an i.i.d. sample

$$\widehat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log f(y_i;\theta) = \arg\max_{\theta} \ell_n(\theta)$$

where the (Monte Carlo) sample is the data drawn from f_0. In alternative form, $\widehat{\theta}$ is the solution to the estimating equation

$$\dot{\ell}_n(\theta) = \sum_{i=1}^{n} \frac{\partial \log f(y_i;\theta)}{\partial\theta} = 0$$

Predictive distributions

A Bayesian version of the calculation replaces the original sample by a sample from the predictive distribution

$$p_n\left(y_{(n+1):(n+m)}\right).$$

However, we are not restricted to use the 'score' function as the basis of an estimation procedure

Predictive distributions

- we may use *any* loss function $L(y, \theta)$ say, and define the Bayesian estimator as

$$\arg\min_{\theta} \int L(y, \theta) p_n(y)\, dy = \arg\min_{\theta} \mathbb{E}_{p_n}[L(Y, \theta)]$$

- this is a valid fully Bayesian estimator as it minimizes an expected posterior loss;
- via this route, we may achieve fully Bayesian inference in a *semi-parametric* fashion.

The Bayesian Bootstrap

The *Bayesian bootstrap* replaces the $1/n$ weights in the bootstrap by repeated draws of $W = (W_1, \ldots, W_n)$

$$W \sim \mathit{Dirichlet}(n-1; 1, 1, \ldots, 1)$$

where

$$\mathbb{E}[W_i] = \frac{1}{n}$$

and uses this as the predictive distribution $p_n(.)$.

The Bayesian Bootstrap

That is, $p_n(.)$ is the predictive distribution derived from a Dirichlet process model, in the limiting case with $\alpha \longrightarrow 0$, so that, given that we have an observed draw

$$w = (w_1, \dots, w_n)$$

of $W \sim \textit{Dirichlet}(n-1; 1, 1, \dots, 1)$, the predictive distribution takes the form

$$p_n(y) = \sum_{i=1}^{n} w_i \delta_{y_i}(y).$$

The Bayesian Bootstrap

This yields the calculation

$$\mathbb{E}_{p_n}[L(Y,\theta)] = \sum_{i=1}^{n} w_i L(y_i,\theta)$$

and in the specific case of the log-density loss

$$\mathbb{E}_{p_n}[L(Y,\theta)] = -\sum_{i=1}^{n} w_i \ell(y_i;\theta)$$

The Bayesian Bootstrap

Hence, we must perform the calculation of

$$\theta_{\mathrm{OPT}} = \arg\max_{\theta} \sum_{i=1}^{n} w_i \ell(y_i; \theta)$$

to minimize the loss.

The quantity θ_{OPT} is a functional of the Dirichlet process posterior, and so we may build up a posterior distribution for it by *repeatedly sampling* the Dirichlet weights, and recomputing θ_{OPT} for each sample.

Example

Example: Bayesian bootstrap

See `knitr 21`

Extensions & Open Problems

- Extensions :
  - Polya Tree Models
  - Hypothesis Testing
  - Spatial Problems
  - Normalized Random Measures
  - Connections with Lévy Processes
- Technical challenges:
  - Properties of Estimators (consistency etc.)
  - Model Selection
  - Comparison with *Bayesian Semi-Parametrics*