that doesn't require knowing $\int L(\theta) g(\theta) d\theta$.

**Rejection Algorithm** (for sampling from $L(\theta) g(\theta)$)

Suppose that $h(\theta)$ specifies a density from which I can easily sample and a constant $M > 0$ so that

$$M h(\theta) > L(\theta) g(\theta) \quad \text{for } g(\theta) \text{ known only up to a multiplicative constant.}$$

To generate $\theta^* \sim g(\theta | \mathcal{Z})$, I can:

1. generate $\theta^{**} \sim h(\theta)$
2. independently generate $U \sim U(0,1)$
3. if

$$M h(\theta^{**}) < L(\theta^{**}) g(\theta^{**})$$

set $\theta^* = \theta^{**}$

otherwise return to 1.

(and upon approximating $\theta_1^*, \theta_2^*, ..., \theta_n^*$ this way I can approximate $Q$)

Here's an argument for why the rejection algorithm works — note that

$$P[\text{algorithm stops at the current iteration} | \theta^{**}] = \frac{L(\theta^{**}) g(\theta^{**})}{M h(\theta^{**})}$$

So

$$P[\text{algorithm fails to stop on a given iteration}] = 1 - \int \frac{L(\theta) g(\theta)}{M h(\theta)} d\theta$$

$$= 1 - \frac{1}{M} \int L(\theta) g(\theta) d\theta$$
So, \( P[\text{iterations are required and } \theta^* \text{ is near } \theta] \) in "\( \Delta \theta \)"

\[
\approx (1 - \frac{1}{M} \int L(\theta) g(\theta) d\theta)^{i-1} \left( \frac{L(\theta) g(\theta)}{M \cdot h(\theta)} \right) h(\theta) \left( \text{vol} \left( \frac{\text{"\( \Delta \theta \)"}}{\text{vol}(\text{"\( \Delta \theta \)"})} \right) \right)
\]

\[
P[\theta^* \text{ is near } \theta] = \sum_{i=1}^{\infty} \text{above in "\( \Delta \theta \)"}
\]

\[
= \frac{\text{L}(\theta) g(\theta) \text{vol}(\text{"\( \Delta \theta \)"})}{M} \left( \frac{1}{1 - (1 - \frac{1}{M} \int L(\theta) g(\theta) d\theta)} \right)
\]

\[
= \frac{\text{L}(\theta) g(\theta) \text{vol}(\text{"\( \Delta \theta \)"})}{\int L(\theta) g(\theta) d\theta}
\]

i.e. \( \theta^* \sim g(\theta|x) \)

It can be difficult/impossible to find appropriate \( h(\theta) \) and \( M \) - The most efficient algorithm is
\( \theta \) \( h(\theta) = g(\theta|x) \) and \( M = 1 \)

A key idea in modern Bayes analysis is that after
I can find stochastic models for \( \theta^*, \theta^*_2, \ldots \)
related to \( g(\theta|x) \) that

\( i) \) can be simulated from easily (sometimes without knowing \( \int L(\theta) g(\theta) d\theta \)

\( ii) \) while not giving iid \( \theta^*_1, \theta^*_2, \ldots \) nevertheless has the ergodicity property. That
The scheme used to do the simulation

\[ \frac{1}{n} \sum_{i=1}^{n} q_i(\theta_i^*) \rightarrow \tilde{Q} \]

is to use a class of models that facilitates this. This is the class of Markov Chain models — we'll look very briefly at 2 methods of MCMC.

**Successive Substitution Sampling** ("Gibbs Sampling") often works to produce a sequence \( \theta_1^*, \theta_2^*, \ldots \) with desired ergodicity property.

In what follows, abbreviate (for iterates \( \theta_i^* \))

- for each \( j = 1, 2, \ldots, k \) = dimension of \( \theta \)
  \[ \theta_i^{*<j} = (\theta_{i1}^*, \theta_{i2}^*, \ldots, \theta_{ij-1}^*) \]

- and \( \theta_i^{*\geq j} = (\theta_{ij+1}^*, \ldots, \theta_{ik}^*) \)

Start with \( \theta_0^* \) (possibly generated from an approximation to \( q(\theta, x) \) or from \( q(\theta) \)).

With \( \theta_i^* \) in hand, generate \( \theta_{i+1}^* \) as follows. For each \( j = 1, 2, \ldots, k \) generate \( \theta_{i+1,j}^* \) from

\[ L(\theta_{i+1,<j}^*, \theta_{i+1,j}^*) g(\theta_{i+1,<j}^*, \theta_{i+1,j}^*) \]

i.e. hold all entries of \( \theta \) at their current iterate values except the \( j \)th and generate a replacement.
for the current value of $\Theta_{ij}$. The resulting density

Sometimes one can see by inspection how to do this, sometimes one can use the rejection algorithm, other times clever new tricks are needed ... but under appropriate conditions this can produce a sequence $\Theta^*_1, \Theta^*_2, \ldots$ s.t.

$$P \quad G(\Theta|z)$$

empirical

$$\Theta^*_1, \ldots, \Theta^*_n$$

and so integrals (probabilities, moments, etc.) of the empirical dsn approximate those of the posterior.

Example: Consider a problem where we have parameters $M_1, M_2 \in \mathbb{R}$ and $c \in (0, 1)$ — given these parameters, we'll suppose that $(X, Y)$ has a joint dsn specified by

$$X \sim U(0, 1)$$

$$Y | X \sim N(M_1, 1) \quad \text{if} \quad X < c$$

$$N(M_2, 1) \quad \text{if} \quad X > c$$

i.e. $E[X | Y]$ is a step function.
I might want to do Bayes analysis for parameter vector \((m_1, m_2, c)\) or the parametric function

\[
E[Y | X = x] = \begin{cases} 
  m_1 & \text{if } x < c \\
  m_2 & \text{if } x \geq c
\end{cases}
\]

If you give me \(n\) iid observations, I have likelihood:

\[
L(\theta) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{1}{2}\left(\frac{\sum_{x_i < c} (y_i - m_1)^2 + \sum_{x_i \geq c} (y_i - m_2)^2}{\sigma^2}\right)\right)
\]

I might then use a prior of independence with a priori:

\[
c \sim U(0, 1) \\
m_1 \sim N(0, \theta^2) \\
m_2 \sim N(0, \theta^2)
\]

So that \(g(\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma^2}\right)^n \exp\left(-\frac{1}{2\sigma^2}(m_1^2 + m_2^2)\right)\)

A Gibbs algorithm might then start at \(\theta_0^*\) with \((c, m_1, m_2)\) generated from prior - then considering the form

\[
L(\theta)g(\theta)
\]

for fixed \((x_1, y_1), \ldots, (x_n, y_n)\), it is clear that I could, e.g., with \(c_i^*, m_1^*, m_2^*\) in hand
1. Update $M^*_1$ sampling from

$$N \left( \frac{\bar{y}_1}{n_1 \sigma^2 + 1}, \frac{\sigma^2}{n_1 \sigma^2 + 1} \right)$$

Sample mean $y_1$

$n_1 = \# \left[ x_i < z^*_1 \right]$

for $z < z^*_1$

2. Update $M^*_2$ sampling from

$$N \left( \frac{\bar{y}_2}{n_2 \sigma^2 + 1}, \frac{\sigma^2}{n_2 \sigma^2 + 1} \right)$$

3. Update $\epsilon_i$ sampling from a density on $[0,1]$ that is constant between each pair of ordered $x(i)$ with value between $x(i)$ and $x(i+1)$ proportional to

$$h_m = \exp \left( -\frac{1}{2} \left( \sum_{x \text{ with } x \leq x(m)} (y_e - M^*_{y(i+1)})^2 + \sum_{x \text{ with } x \geq x(m)} (y_e - M^*_{y(i+1)})^2 \right) \right)$$

i.e. density $h_m/(\sum_{m=0}^M h_m(x(m+1) - x(m))$ on $(x(m), x(m+1))$

This gives an algorithm that can be used to make $\theta^*$ that will have empirical dsn approximating $G(\theta|data)$

In practice, big issues with Bayes MCMC (including SSS) are

i) making sure the set-up is not "pathological" and the ergodicity property holds

ii) determining when any "transient"/"start up" problems with the simulation have washed out
and we should begin an averaging process to approximate Q's. What "burn in" is adequate.

iii) deciding how long to run the simulation.

Another MCMC Algorithm (that can be used in its own right or to substitute for a SSS step that is not easy to do) is the Metropolis/Hastings Algorithm - the M-H (alone) version of this is

Start with $\theta_0$

With $\theta^*_l$ in mind let $J_{hi}(\theta' | \theta)$ specify for each $\theta$ a dsm for $\theta$ over $\Theta$ from which one can sample - generate a "proposal"/"candidate" replacement for $\theta^*_l$

$$\theta^{**}_{ih} \sim J_{hi}(\cdot | \theta^*_l)$$

The jumping kernel/proposal dsm

and accept it based on

$$r_{ih} = \frac{L(\theta^{**}_{ih}) g(\theta^{**}_{ih}) / J_{hi}(\theta^{**}_{ih} | \theta^*_l)}{L(\theta^*_h) g(\theta^*_h) / J_{hi}(\theta^*_h | \theta^*_l)}$$

i.e., with probability $\min(r_{ih}, 1)$ - i.e., with

$Y_{ih} \sim$ Bernoulli ($\min(r_{ih}, 1)$)

$$\theta^*_{i+1} = Y_{ih} \theta^{**}_{ih} + (1 - Y_{ih}) \theta^*_i$$

Often, this algorithm will produce a $\theta^*$ sequence for which the ergodicity property holds.
This is more or less a kind of "adaptive rejection sampling" methodology - one often chooses the $J_{i+i}^i(\theta'|\theta)$ to more or less specify $\theta'$ as being a small random perturbation of $\theta$.

A very nice special instance of this is one where $J_{i+i}^i$ is symmetric, i.e. $J_{i+i}^i(\theta'|\theta) = J_{i+i}^i(\theta|\theta')$ - in this special case the jumping ratio is

$$r_{i+i} = \frac{L(\theta_{i+i}^*)g(\theta_{i+i}^*)}{L(\theta_i^*)g(\theta_i^*)}$$

(and one always jumps to the proposal if it takes one up-hill on $L(\theta)g(\theta)$) and one has a "Metropolis" algorithm.

Also (in a very important development) one can use the M-H idea to replace straight Gibbs updates in a SSS algorithm - that is, when updating $\theta_{i,j}$ (having updated all $\theta_{i,k}$ with $k < j$)

I can specify

$$J_{i+i,j}^i(\theta_j'|\theta_j)$$

(that can actually depend not only on $\theta_j$ but on the current values $\theta_{i,j}^*$ and $\theta_{i,j}^*$) and generate a proposal

$$\theta_{i+j,j}^{**} \sim J_{i+i,j}^i(\cdot | \theta_{i,j}^*)$$
and accept it based on

\[
R_{i + 1, j} = \frac{L(\theta_{1,i+1}^*, \theta_{2,i+1}^*, \theta_{j}^*, \text{same})}{L(\theta_{1,i+1}^*, \theta_{2,i+1}^*, \theta_{j}^*, \text{same})} \frac{J_{i+1,j}(\theta_{i+1,j}^*, \theta_{j}^*)}{J_{i+1,j}(\text{reversed})}
\]

**Example**  Previous one but replacing The c Gibbs step (step 3) with a M-H step (keeping same prior)

Note that the prior for c being uniform (0,1) makes, e.g., with \( c = \log \frac{1}{1 + e^d} \) (so \( c = \frac{1}{1 + e^d} \))

\[
P[ d < t ] = P[ c < \frac{1}{1 + e^t} ] = \frac{1}{1 + e^t}
\]

so that \( c \) takes values on \( R \) with prior density

\[
\frac{d}{dt} \left( \frac{1}{1 + e^{-t}} \right) = \frac{e^{-t}}{(1 + e^{-t})^2}
\]

So, one way to replace the c Gibbs step with a Metropolis step is to operate on d rather than c and propose

\[
d_{i+1}^* \sim N(d_i^*, \tau^2)
\]

**tuning parameter for the algorithm**

This makes \( J_{i+1}(d, id) = J_{i+1}(d, id') \)

accept it based on

\[
R_{i+1} = \frac{L(M_{1,i+1}^*, M_{2,i+1}^*, c_{i+1}^*)}{L(M_{1,i+1}^*, M_{2,i+1}^*, c_{i+1}^*)} \frac{e^{-d_{i+1}^*}}{(1 + e^{-d_{i+1}^*})^2} \frac{e^{-d_{i+1}^*}}{(1 + e^{-d_{i+1}^*})^2}
\]

\[= \frac{L(\text{same})}{L(\text{same})} \frac{c_{i+1}^*(1 - c_{i+1}^*)}{c_i^*(1 - c_i^*)} \]