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 dsn from which I can easily sample and a constant $M > 0$ so that

$$Mh(\theta) > L(\theta)g(\theta)$$

$g(\theta|x)$ known only up to a multiplicative constant

To generate $\theta^* \sim g(\theta|x)$ I can

1. generate $\theta^{**} \sim h(\theta)$
2. independently generate $u \sim U(0,1)$
3. if $Muh(\theta^{**}) < L(\theta^{**})g(\theta^{**})$ set $\theta^* = \theta^{**}$ otherwise return to 1

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

Why

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

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

So $P[\text{algorithm fails to stop on a given iteration}]$

$$= 1 - \int \frac{L(\theta)g(\theta)}{Mh(\theta)} h(\theta)\,d\theta$$

$$= 1 - \frac{1}{M}\int L(\theta)g(\theta)\,d\theta$$

So $P[i \text{ iterations are required and } \theta^* \underbrace{\text{is near } \theta}_{\text{in } "\Delta\theta"}]$

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

$$P\left[\theta^* \text{ is } \underbrace{\text{near } \theta}_{\text{in } "\Delta\theta"}\right] = \sum_{i=1}^{\infty} \text{above}$$

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

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

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

It can be difficult/impossible to find appropriate $h(\theta)$ and M — The most efficient algorithm is

☺ $h(\theta) = g(\theta|x)$ and $M = 1$

A key idea in modern Bayes analysis is that often I can find stochastic models for $\theta_1^*, \theta_2^*, \ldots$ 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

$$\frac{1}{n}\sum_{i=1}^{n} q(\theta_i^*) \xrightarrow{P} Q$$

(P: probability for the scheme used to do the simulation)

a class of models that facilitates this is the class of Markov Chain models - we'll look very briefly at 2 methods of MCMC

Successive Substitution Sampling ("Gibbs Sampling")

often the following works to produce a sequence $\theta_1^*, \theta_2^*, \ldots$ with desired ergodicity property

In what follows, abbreviate (for iterates θ_i^*)

for each $j = 1, 2, \ldots, k$ = dimension of θ

$$\theta_{i,<j}^* = (\theta_{i,1}^*, \theta_{i,2}^*, \ldots, \theta_{i,j-1}^*)$$

and $\theta_{i,>j}^* = (\theta_{i,j+1}^*, \ldots, \theta_{i,k}^*)$

Start with θ_0^* (possibly generated from an approximation to $g(\theta, x)$ or from $g(\theta)$)

With θ_i^* in hand, generate θ_{i+1}^* as follows - For each $j = 1, 2, \ldots, k$ generate $\theta_{i+1,j}^*$ from

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

i.e. hold all entries of θ at their current iterate values except the jth and generate a replacement

for the current value of θ^*_{ij} from the resulting density

Sometimes we 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.

$$\text{empirical dsn of } \theta^*_1, \ldots, \theta^*_n \xrightarrow{P} G(\theta \mid z)$$

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

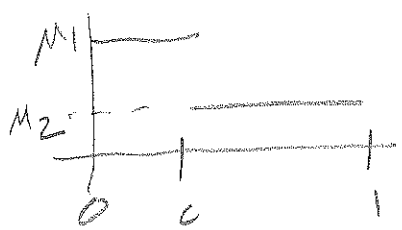
Jan 20

Example Consider a problem where we have parameters $\mu_1, \mu_2 \in \mathbb{R}$ and $c \in (0,1)$ — given these parameters we'll suppose that (X,Y) has a joint dsn specified by

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

$$Y \mid X \sim \begin{cases} N(\mu_1, 1) & \text{if } X < c \\ N(\mu_2, 1) & \text{if } X > c \end{cases}$$

i.e. $\exists$ some step function $E[Y \mid X]$



I might want to do Bayes analysis for parameter vector (μ_1, μ_2, c) or the parametric function

$$E[Y|X=x] = \begin{cases} \mu_1 & \text{if } x<c \\ \mu_2 & \text{if } x>c \end{cases}$$

If you give me n iid observations, I have likelihood

$$L(\theta) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp -\frac{1}{2}\left(\sum_{x_i<c}(y_i-\mu_1)^2 + \sum_{x_i>c}(y_i-\mu_2)^2\right)$$

I might, then use a prior of independence with a prior

$$c \sim U(0,1)$$
$$\mu_1 \sim N(0, \gamma^2)$$
$$\mu_2 \sim N(0, \gamma^2)$$

So that $g(\theta) = \left(\frac{1}{\sqrt{2\pi\gamma^2}}\right)^2 \exp -\frac{1}{2\gamma^2}(\mu_1^2+\mu_2^2)$

A Gibbs algorithm might then start at θ_0^* with (c, μ_1, μ_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 $c_i^*, \mu_{1i}^*, \mu_{2i}^*$ in hand

(1) update μ_{1i}^* sampling from

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

$\bar{y}_1$: sample mean y_l for $x_l < c_i^*$

$n_1 = \#[x_l < c_i^*]$

(2) update μ_{2i}^* sampling from

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

(3) update c_i^* sampling from a density on $[0,1]$ that is constant between each pair of ordered $x_{(l)}$ with value between $x_{(l)}$ and $x_{(l+1)}$ proportional to

(take $x_{(0)}=0$ and $x_{(n+1)}=1$)

$$h_m = \exp -\frac{1}{2}\left(\sum_{l \text{ with } x_l \le x_{(m)}} (y_l - \mu_{1,i+1}^*)^2 + \sum_{l \text{ with } x_l \ge x_{(m+1)}} (y_l - \mu_{2,i+1}^*)^2\right)$$

i.e. density $h_m / \left(\sum_{m=0}^{n} h_m (x_{(m+1)} - x_{(m)})\right)$ on $(x_{(m)}, x_{(m+1)})$

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

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 one should begin an averaging process to approximate Q's (what "burn in" is adequate)

(ii) deciding how long to run the simulation

Another MCMC Algorithm (that can be used in its own right or to substitute for a step that is not easy to do) is the

day 21

Metropolis/Hastings Algorithm - the M-H (alone) version of this is

Start with θ_0^*

With θ_i^* in hand let $J_{i+1}(\theta'|\theta)$ specify for each θ a dsn for θ' over Θ from which one can sample - generate a "proposal"/"candidate" replacement for θ_i^*

$$\theta_{i+1}^{**} \sim J_{i+1}(\cdot \mid \theta_i^*)$$ ← the jumping kernel/proposal dsn

and accept it based on

$$r_{i+1} = \frac{L(\theta_{i+1}^{**})\, g(\theta_{i+1}^{**}) \,/\, J_{i+1}(\theta_{i+1}^{**} \mid \theta_i^*)}{L(\theta_i^*)\, g(\theta_i^{**}) \,/\, J_{i+1}(\theta_i^* \mid \theta_{i+1}^{**})}$$

i.e. with probability $\min(r_{i+1}, 1)$ - i.e. with $Y_{i+1} \sim \text{Bernoulli}(\min(r_{i+1}, 1))$

$$\theta_{i+1}^* = Y_{i+1}\,\theta_{i+1}^{**} + (1 - Y_{i+1})\,\theta_i^*$$

Often, this algorithm will produce a θ_i^* sequence for which the ergodicity property holds

This is more or less a kind of "adaptive rejection sampling" methodology - one often chooses the $J_{i+1}(\theta'|\theta)$ to more or less specify θ' as being a small random perturbation of θ

A very nice special instance of this is one where J_{i+1} is symmetric, i.e. $J_{i+1}(\theta'|\theta) = J_{i+1}(\theta|\theta')$ - in this special case the jumping ratio is

$$r_{i+1} = \frac{L(\theta^{**}_{i+1})\, g(\theta^{**}_{i+1})}{L(\theta^{*}_{i})\, g(\theta^{*}_{i})}$$

(and one always jumps to the proposal if it takes one uphill 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 $\theta^{*}_{i,l}$ with $l<j$) I can specify

$$J_{i+1,j}(\theta'_j|\theta_j)$$

(that can actually depend not only on θ_j but on the current values $\theta^{*}_{i+1,<j}$ and $\theta^{*}_{i,>j}$) and generate a proposal

$$\theta^{**}_{i+1,j} \sim J_{i+1,j}(\cdot|\theta^{*}_{i,j})$$

and accept it based on

$$r_{i+1,j} = \frac{L(\theta^*_{i+1,<j}, \theta^{**}_{i+1,j}, \theta^*_{i,>j})\; g(\text{same})/J_{i+1,j}(\theta^{**}_{i+1,j} \mid \theta^*_{ij})}{L(\quad, \theta^*_{ij}, \quad)\; g(\text{same})/J_{i+1,j}(\text{reverse})}$$

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 $d = \log\frac{c}{1-c}$ $\left(\text{so } c = \frac{1}{1+e^{-d}}\right)$

$$P[d \le t] = P\left[c \le \frac{1}{1+e^{-t}}\right] = \frac{1}{1+e^{-t}}$$

so that d takes values on $\mathbb{R}$ 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

$$d^{**}_{i+1} \sim N(d^*_i, \tau^2)$$

(τ^2: a tuning parameter for the algorithm)

This makes $J_{i+1}(d' \mid d) = J_{i+1}(d \mid d')$

accept it based on

$$r_{i+1} = \frac{L(\mu^*_{1,i+1}, \mu^*_{2,i+1}, c^{**}_{i+1})}{L(\mu^*_{1,i+1}, \mu^*_{2,i+1}, c^*_i)} \frac{\dfrac{e^{-d^{**}_{i+1}}}{(1+e^{-d^{**}_{i+1}})^2}}{\dfrac{e^{-d^*_i}}{(1+e^{-d^*_i})^2}}$$

$$= \frac{L(\quad)}{L(\quad)} \frac{c^{**}_{i+1}(1-c^{**}_{i+1})}{c^*_i(1-c^*_i)}$$