Generating Random Variables with a Prescribed Distribution

Monte Carlo

Suppose we wish to calculate the integral

$$I = \int m(\boldsymbol{x}) f(\boldsymbol{x}) \, d\boldsymbol{x}$$

If we can generate an *i.i.d.* sequence

$$\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n$$

with common density f, then

$$\hat{I} = \frac{1}{n} \sum_{i=1}^{n} m(\boldsymbol{X}_i) \tag{1}$$

is a **consistent, unbiased and asymptotically normal** estimate of I.

In some cases, implementation of (1) may not be convenient nor feasible. For example,

- it is not easy (or possible) to generate independent random vectors with joint density f
- the density f is only known up to a multiplicative constant

Markov Chain Monte Carlo (MCMC)

An alternative procedure is to generate a **Markov Chain** sequence

$$\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n$$

with

- **transition kernel**

$$h(\boldsymbol{y}|\boldsymbol{x})$$

- **invariant density** $f(\boldsymbol{x})$

$$\int_{\mathcal{S}} h(\boldsymbol{y}|\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} = f(\boldsymbol{y})$$

Then

$$\hat{I} = \frac{1}{n} \sum_{i=1}^{n} m(\boldsymbol{X}_i)$$

is still a **consistent and unbiased** estimate of I.

A simple condition for invariance of $f(\boldsymbol{x})$ is

$$h(\boldsymbol{y}|\boldsymbol{x}) f(\boldsymbol{x}) = f(\boldsymbol{y}) h(\boldsymbol{x}|\boldsymbol{y}), \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$$

In fact, in this case

$$\int_{\mathcal{S}} h(\boldsymbol{y}|\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} = f(\boldsymbol{y}) \int_{\mathcal{S}} h(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} = f(\boldsymbol{y}) \quad \text{for all } \boldsymbol{y} \in \mathcal{S}$$

A Key Result

Suppose that the Markov process is **irreducible and aperiodic** and has a **positive invariant distribution** $f^{(\infty)}(\boldsymbol{y}) > 0$, for all $\boldsymbol{y} \in \mathcal{S}$.

Then:

1.

$$f^{(\infty)}(\boldsymbol{y}) = \lim_{n\to\infty} p^{(n)}(\boldsymbol{y}|\boldsymbol{x}), \text{ for all } \boldsymbol{y} \in \mathcal{S}$$

   independent from x.

   NOTE: in the finite state space case, all the rows of $P^{(\infty)}$ are the same.

2. $f^{(\infty)}(\boldsymbol{y})$ is the unique, positive solution of the equation

$$f^{(\infty)}(\boldsymbol{y}) = \sum_{x\in\mathcal{S}} f^{(\infty)}(\boldsymbol{x})\, p(\boldsymbol{y}|\boldsymbol{x})$$

   and $\left\{f^{(\infty)}(\boldsymbol{y})\right\}$ satisfies

$$\sum_{x\in\mathcal{S}} f^{(\infty)}(\boldsymbol{y}) = 1$$

   That is, $f^{(\infty)}(\boldsymbol{y})$ is a pmf.

3. In the continuous case $f^{(\infty)}(\boldsymbol{y}) > 0$, for all $\boldsymbol{y} \in \mathcal{S}$

$$f^{(\infty)}(\boldsymbol{y}) = \int_{\mathcal{S}} f^{(\infty)}(\boldsymbol{x})\, p(\boldsymbol{y}|\boldsymbol{x})\, d\boldsymbol{x}$$

   and $f^{(\infty)}(\boldsymbol{y})$ satisfies

$$\int_{\mathcal{S}} f^{(\infty)}(\boldsymbol{y})\, d\boldsymbol{y} = 1$$

   That is, $f^{(\infty)}(\boldsymbol{y})$ is a cdf.

4. In the discrete case $f^{(\infty)}(\boldsymbol{y})$ can be interpreted as the long-run proportion of time that the process is in state $\boldsymbol{y}$.

5. Let $F^{(\infty)}(y)$ be the corresponding invariant distribution function. Suppose that

$$E_{F^{(\infty)}}\left(\left|g\left(\mathbf{Y}\right)\right|\right) < \infty$$

Then

$$\begin{aligned} \frac{1}{n}\sum_{n=1}^{\infty} g\left(\boldsymbol{X}_n\right) &= \sum_{y} g\left(\boldsymbol{y}\right) f^{(\infty)}\left(\boldsymbol{y}\right) \\ &= E_{F^{(\infty)}}\left\{g\left(\mathbf{Y}\right)\right\} \end{aligned}$$

In the continuous case

$$\begin{aligned} \frac{1}{n}\sum_{n=1}^{\infty} g\left(\boldsymbol{X}_n\right) &= \int_{\mathcal{S}} g\left(\boldsymbol{y}\right) f^{(\infty)}\left(\boldsymbol{y}\right) d\boldsymbol{y} \\ &= E_{F^{(\infty)}}\left\{g\left(\mathbf{Y}\right)\right\} \end{aligned}$$

The Metropolis-Hasting Algorithm

The Algorithm

- **INPUT:** A function $g(\boldsymbol{x})$ which is proportional to the the target density $f(\boldsymbol{x})$. In other words

$$f(\boldsymbol{x}) = \alpha g(\boldsymbol{x})$$

  for some possibly unknown constant α.

- Let $\boldsymbol{x}=\boldsymbol{x}^{(m)}$ be the current "state" of the sequence

- Let

$$q(\boldsymbol{y}|\boldsymbol{x})$$

  be an auxiliary (irreducible-aperiodic) kernel. It should be possible (easy) to generate a random vector with distribution $q\left(\bullet|\boldsymbol{x}^{(m)}\right)$, for any given "current state" $\boldsymbol{x}^{(m)}$.

- The candidate transition: generate $\boldsymbol{y}\sim q\left(\bullet|\boldsymbol{x}^{(m)}\right)$

- Let u be an independent random variable with uniform distribution on the interval $(0,1)$.

- The next state, $\boldsymbol{x}^{(m+1)}$, is defined as follows

$$\boldsymbol{x}^{(m+1)} = \begin{cases} \boldsymbol{y} & u \leq f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}) / f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x}) \\ \boldsymbol{x} & u > f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}) / f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x}) \end{cases}$$

Then it can be shown that

$$\left\{ \boldsymbol{x}^{(m)} \right\}$$

is a realization of a Markov Chain with stationary distribution $f(\boldsymbol{x})$.

Example 1: Suppose we wish to estimate

$$p = P\left(X_1 + X_2 > 1\right) \tag{2}$$

where

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

has joint density

$$f(\boldsymbol{x}) \propto \exp\left\{-\|\boldsymbol{x}\|\right\}.$$

Solution

To estimate p we can use the M-H algorithm to generate a Markov chain with stationary density $f(\boldsymbol{x})$. For instance, we can use the "independent kernel"

$$q(\boldsymbol{y}|\boldsymbol{x}) = \left(\frac{1}{2\pi}\right)^{3/2} \exp\left\{-\|\boldsymbol{y}\|^2\right\}.$$

That is, independent from the current value $\boldsymbol{x}^{(m)}$, the candidate $\boldsymbol{y} = (y_1, y_2, y_3)$ is formed by independent standard normal random variables. We can take $\boldsymbol{x}^{(0)} = \mathbf{0}$, and given $\boldsymbol{x}^{(m)}$ accept $\boldsymbol{y}$ if

$$\begin{aligned} u &\leq \frac{f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y})}{f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x})} \\ &= \frac{\exp\left\{-\|\boldsymbol{y}\|\right\} \exp\left\{-\frac{1}{2}\|\boldsymbol{x}\|^2\right\}}{\exp\left\{-\|\boldsymbol{x}\|\right\} \exp\left\{-\frac{1}{2}\|\boldsymbol{y}\|^2\right\}} \\ &= \frac{\exp\left\{\|\boldsymbol{x}\| - \frac{1}{2}\|\boldsymbol{x}\|^2\right\}}{\exp\left\{\|\boldsymbol{y}\| - \frac{1}{2}\|\boldsymbol{y}\|^2\right\}} \end{aligned}$$

A Markov chain of length $n = 200,000$, with a burn in of $1,000$ gave $\hat{p} = 0.3282117$.

```
#######################################
## This is an R function to performs the MCMC calculations


mh=function(n,n0){
#n  is the desired sequence length
#n0  is the "burn-in" parameter
count=0
 res=matrix(0,n+n0,3)
 res[1,]=c(0,0,0)
   for(i in 2:(n+n0)){
 y=rnorm(3);x=res[i-1,]
 u=runif(1)
 yy=sqrt(sum(y^2))
 xx=sqrt(sum(x^2))
  test=(exp(-yy)/exp(-xx))*exp(-0.5*xx^2)/exp(-0.5*yy^2)
 if(u<=test){res[i,]=y;count=count+1}
 if(u>test){res[i,]=x}
                        }
                   return(list(res[(n0+1):(n+n0),],count))       }

  set.seed(13)
  re=mh(2000000,10000)
  accept = (re[[2]]/dim(re[[1]])[1])
  res = re[[1]]


test1=res[,1]+res[,2]
 r1=mean(test1>1)
test2=res[,1]+res[,3]
 r2=mean(test2>1)
test3=res[,2]+res[,3]
 r3=mean(test3>1)
(r1+r2+r3)/3

 c(r1,r2,r3)
 accept




plot(1:1000,test1[1:1000])
#####################################################
```

Example 2: Suppose that we wish to generate random permutations $(x_1, x_2, ..., x_n)$ of the set $\{1, 2, ..., n\}$ such that

$$\sum_{j=1}^{n} j x_j > a, \tag{3}$$

where a is a given constant.

Let $\mathcal{C}_0$ be the set of all the permutations $(x_1, x_2, ..., x_n)$ that satisfy (3). Then the conditional density of interest is

$$f(\boldsymbol{x}) = \frac{1}{|\mathcal{C}_0|}$$

for all $\boldsymbol{x} \in \mathcal{C}_0$. Here,

$$|\mathcal{C}_0| = \text{number of elements in } \mathcal{C}_0.$$

Unfortunately, $|\mathcal{C}_0|$ is unknown.

Solution

Approach 1:

We begin by defining the set $N(\boldsymbol{x})$ of "neighbors of $\boldsymbol{x}$" for all $\in \boldsymbol{x}\ \mathcal{C}_0$:

A permutations $\boldsymbol{y}$ in $\mathcal{C}_0$ is a neighbor of $\boldsymbol{x}$ if $\boldsymbol{x}$ and $\boldsymbol{y}$ differ at most on a pair of entries. For example, let $\boldsymbol{x} = (1, 2, 3, 4, 5, 6)$, then $(1, 2, 3, 6, 5, 4)$ is a neighbor of $\boldsymbol{x}$ but $(1, 2, 3, 5, 6, 4)$ is not.

The transition kernel in this approach is defined as follows

$$q(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} \frac{1}{|N(\boldsymbol{x})|} & \text{if} \quad \boldsymbol{y} \in N(\boldsymbol{x}) \\ 0 & \text{otherwise} \end{cases},$$

where $|N(\boldsymbol{x})|$ is equal to the number of elements in $N(\boldsymbol{x})$.

A neighbor $\boldsymbol{y}$ of the current state $\boldsymbol{x}$ is randomly chosen (a list of such neighbors must be constructed) together with an independent uniform random variable u in $(0,1)$.

The new state $\boldsymbol{y}$ is accepted if

$$u \leq \frac{f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y})}{f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x})} = \frac{q(\boldsymbol{x}|\boldsymbol{y})}{q(\boldsymbol{y}|\boldsymbol{x})}$$

$$= \frac{1/|N(\boldsymbol{y})|}{1/|N(\boldsymbol{x})|} = \frac{|N(\boldsymbol{x})|}{|N(\boldsymbol{y})|}.$$

Notice that if the new state $\boldsymbol{y}$ has fewer neighbors than the current state, then $\boldsymbol{y}$ is accepted with probability one.

Approach 2: Let now $\mathcal{C}$ be the set of all the permutations of $\{1,2,...,n\}$, irrespective of whether they satisfy (3) or not.

The neighbors $N(\boldsymbol{x})$ of $\boldsymbol{x}\in\mathcal{C}$ are now all the permutations $\boldsymbol{y}$ in $\mathcal{C}$ that differ from $\boldsymbol{x}$ in at most a pair of entries.

It is clear now that

$$|N(\boldsymbol{x})| = |N(\boldsymbol{y})| = \binom{n}{2}, \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in\mathcal{C}.$$

The transition kernel in this approach is

$$q(\boldsymbol{y}|\boldsymbol{x}) = \frac{2}{n(n-1)}, \quad \text{for all } \boldsymbol{x},\boldsymbol{y}\in\mathcal{C}.$$

Moreover, the density of interest is

$$f(\boldsymbol{x}) = \begin{cases} \frac{1}{|\mathcal{C}_0|} & \text{if} \quad \boldsymbol{y}\in\mathcal{C}_0 \\ 0 & \text{if} \quad \boldsymbol{y}\in\mathcal{C}\backslash\mathcal{C}_0 \end{cases},$$

A neighbor $\boldsymbol{y}$ of the current state $\boldsymbol{x}$ is randomly chosen (a list of such neighbors is no longer needed) together with an independent uniform random variable u in $(0,1)$.

The new state $\boldsymbol{y}$ is accepted if

$$u \leq \frac{f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y})}{f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x})} = \frac{f(\boldsymbol{y})}{f(\boldsymbol{x})}$$

$$= \begin{cases} 1 & \text{if} \quad \boldsymbol{y} \in \mathcal{C}_0 \\ \\ 0 & \text{if} \quad \boldsymbol{y} \in \mathcal{C} \backslash \mathcal{C}_0 \end{cases}$$

Notice that if the new state $\boldsymbol{y}$ is accepted with probability one if it belongs to $\mathcal{C}_0$.

Some Remarks

Remark 1: the probability of making a transition from state $\boldsymbol{x}$ to state $\boldsymbol{y}$ is

$$q\left(\boldsymbol{y}|\boldsymbol{x}\right)P\left(u<\frac{f\left(\boldsymbol{y}\right)q\left(\boldsymbol{x}|\boldsymbol{y}\right)}{f\left(\boldsymbol{x}\right)q\left(\boldsymbol{y}|\boldsymbol{x}\right)}\right)=q\left(\boldsymbol{y}|\boldsymbol{x}\right)\min\left\{1,\frac{f\left(\boldsymbol{y}\right)q\left(\boldsymbol{x}|\boldsymbol{y}\right)}{f\left(\boldsymbol{x}\right)q\left(\boldsymbol{y}|\boldsymbol{x}\right)}\right\}$$

and the probability $p\left(\boldsymbol{x}\right)$ of sticking to the state $\boldsymbol{x}$ is

$$p\left(\boldsymbol{x}\right)=1-\int_{y\neq x}q\left(\boldsymbol{y}|\boldsymbol{x}\right)\min\left\{1,\frac{f\left(\boldsymbol{y}\right)q\left(\boldsymbol{x}|\boldsymbol{y}\right)}{f\left(\boldsymbol{x}\right)q\left(\boldsymbol{y}|\boldsymbol{x}\right)}\right\}\mathbf{d}\boldsymbol{y}$$

Hence, the transition kernel $h\left(\boldsymbol{y}|\boldsymbol{x}\right)$ of $\left\{\boldsymbol{x}^{(m)}\right\}$ is

$$h\left(\boldsymbol{y}|\boldsymbol{x}\right)=\underbrace{q\left(\boldsymbol{y}|\boldsymbol{x}\right)\min\left\{1,\frac{f\left(\boldsymbol{y}\right)q\left(\boldsymbol{x}|\boldsymbol{y}\right)}{f\left(\boldsymbol{x}\right)q\left(\boldsymbol{y}|\boldsymbol{x}\right)}\right\}}_{\text{making a transition}}+\underbrace{\delta_{\boldsymbol{x}}\left(\boldsymbol{y}\right)p\left(\boldsymbol{x}\right)}_{\text{sticking to the current state}}$$

where $\delta_{\boldsymbol{x}}\left(\boldsymbol{y}\right)$ is the *Dirac delta function,* which satisfies the conditions

(i) $\delta_{\boldsymbol{x}}\left(\boldsymbol{y}\right)=0$ almost everywhere, when $\boldsymbol{x}\neq\boldsymbol{y}$

(ii) $\int\delta_{\boldsymbol{x}}\left(\boldsymbol{y}\right)D\left(\boldsymbol{y}\right)\mathbf{d}\boldsymbol{y}=D(\boldsymbol{x})$, for all continuous function D

Notice that in particular, taking $D(\boldsymbol{x})=1$, we get

$$\int\delta_{\boldsymbol{x}}\left(\boldsymbol{y}\right)\mathbf{d}\boldsymbol{y}=1.$$

Remark 2: The M–H algorithm works. To see this we will show that the Markov chain $\left\{\boldsymbol{x}^{(m)}\right\}$ has stationary density $f(\boldsymbol{x})$.

For that, it suffices to show that $f(\boldsymbol{x})$ satisfies the stationarity condition:

$$f(\boldsymbol{x})\, h(\boldsymbol{y}|\boldsymbol{x}) = f(\boldsymbol{y})\, h(\boldsymbol{x}|\boldsymbol{y}) \tag{4}$$

where $h(\boldsymbol{y}|\boldsymbol{x})$ is the transition kernel of $\left\{\boldsymbol{x}^{(m)}\right\}$.

We can assume without loss of generality that $\boldsymbol{x}\neq\boldsymbol{y}$ because equation **(4) is** trivially satisfied when $\boldsymbol{x}=\boldsymbol{y}$. Therefore $\delta_{\boldsymbol{x}}(\boldsymbol{y}) = 0$ and

$$h(\boldsymbol{y}|\boldsymbol{x}) = q(\boldsymbol{y}|\boldsymbol{x}) \min\left\{1, \frac{f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y})}{f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x})}\right\}$$

$$f(\boldsymbol{x})\, h(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}), & f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}) \leq f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x}) \\ f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x}), & f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}) \geq f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x}) \end{cases}$$

Notice that in the "equality case" $f(\boldsymbol{y})\, q(\boldsymbol{x}|\boldsymbol{y}) = f(\boldsymbol{x})\, q(\boldsymbol{y}|\boldsymbol{x})$ and therefore there is no inconsistency in the equations above.

Reversing the roles of $\boldsymbol{x}$ and $\boldsymbol{y}$ we have

$$f(\boldsymbol{y})\,h(\boldsymbol{x}|\boldsymbol{y}) = \begin{cases} f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x})\,, & f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \leq f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \\ f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y})\,, & f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \geq f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \end{cases}$$

$$= \begin{cases} f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x})\,, & f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \geq f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \\ f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y})\,, & f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \leq f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \end{cases}$$

$$= \begin{cases} f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y})\,, & f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \leq f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \\ f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x})\,, & f(\boldsymbol{y})\,q(\boldsymbol{x}|\boldsymbol{y}) \geq f(\boldsymbol{x})\,q(\boldsymbol{y}|\boldsymbol{x}) \end{cases}$$

$$= f(\boldsymbol{x})\,h(\boldsymbol{y}|\boldsymbol{x})\,.$$

Remark 3: Notice that in order to implement the Metropolis-Hasting algorithm we only need the ratio

$$\frac{f(\boldsymbol{x})}{f(\boldsymbol{y})}$$

and not the density values $f(\boldsymbol{x})$ and $f(\boldsymbol{y})$ themselves. For example, we could have

$$f(\boldsymbol{x}) = \alpha g(\boldsymbol{x})$$

with $g(\boldsymbol{x})$ known but α possibly unknown. In such case,

$$\begin{aligned}\frac{f(\boldsymbol{x})}{f(\boldsymbol{y})} &= \frac{Kg(\boldsymbol{x})}{Kg(\boldsymbol{y})} \\ &= \frac{g(\boldsymbol{x})}{g(\boldsymbol{y})}\end{aligned}$$

can be calculated using the known values of $g(\boldsymbol{x})$ and $g(\boldsymbol{y})$.

Example. Consider the data

y_i	2	3	3	5	4	6
n_i	20	24	18	25	12	14

Suppose that the y_i are independent binomial random variables with parameters (n_i, p_i) where the n_i are the number of trials and the p_i satisfy

$$0 < p_1 < p_2 < \cdots < p_6 < 1. \tag{5}$$

Consider the Bayesian estimation of the p_i using a constant prior on the vectors $(p_1, p_2, ..., p_6)$ that satisfy (5).

1. Derive the posterior distribution of $(p_1, p_2, ..., p_6)$ given $(y_1, y_2, ..., y_6)$, up to a multiplicative constant.
2. Explain how you can use the Gibbs sampler to generate a Markov Chain with stationary distribution equal to the posterior distribution of $(p_1, p_2, ..., p_6)$ given $(y_1, y_2, ..., y_6)$.
3. Apply the procedure described in part **2** to the given data.

Solution

1. Likelihood

$$f(y_1, y_2, ..., y_6 | p_1, p_2, ..., p_6) = \prod_{i=1}^{6} p_i^{y_i} (1 - p_i)^{n_i - y_i}$$

2. Prior

$$\pi(p_1, p_2, ..., p_6) \propto \begin{cases} 1, & \text{if} \quad 0 < p_1 < p_2 < \cdots < p_6 < 1 \\ 0 & \text{otherwise} \end{cases}$$

3. Joint distribution for the data and the parameters

$$f(y_1, y_2, ..., y_6, p_1, p_2, ..., p_6) \propto \begin{cases} \prod p_i^{y_i} (1 - p_i)^{n_i - y_i}, & \text{if} \quad 0 < p_1 < p_2 < \cdots < p_6 < 1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the posterior is

$$f\left(p_1, p_2, ..., p_6 | y_1, y_2, ..., y_6\right) \propto f\left(y_1, y_2, ..., y_6, p_1, p_2, ..., p_6\right)$$

$$\propto \begin{cases} \prod p_i^{y_i}\left(1-p_i\right)^{n_i-y_i}, & \text{if} \quad p_1 < p_2 < \cdots < p_6 \\ 0 & \text{otherwise} \end{cases}$$

4. To apply the Gibbs sampler we must derive the full conditional distributions for each p_i, that is, the conditional distribution of p_i given $\mathbf{p}_{(-i)}$, and $\boldsymbol{y}$, with

$$\mathbf{p}_{(-i)} = \begin{pmatrix} p_1 \\ \vdots \\ p_{i-1} \\ p_{i+1} \\ \vdots \\ p_6 \end{pmatrix}, \quad i = 2,3,4,5, \qquad \mathbf{p}_{(-1)} = \begin{pmatrix} p_2 \\ \vdots \\ p_i \\ p_{i+1} \\ \vdots \\ p_6 \end{pmatrix}, \qquad \mathbf{p}_{(-6)} = \begin{pmatrix} p_1 \\ \vdots \\ p_i \\ p_{i+1} \\ \vdots \\ p_5 \end{pmatrix}$$

To facilitate the notations set $p_0 = 0$ and $p_7 = 1$, so we can write

$$0 = p_0 < p_1 < p_2 < \cdots < p_{i-1} < p_{i+1} < \cdots < p_6 < p_7 = 1$$

For $1 \leq i \leq 6$,

$$f\left(p_i | \boldsymbol{y}, \mathbf{p}_{(-i)}\right) = \frac{f\left(\mathbf{p}|\boldsymbol{y}\right)}{f\left(\mathbf{p}_{(-i)}|\boldsymbol{y},\right)}$$

$$\propto \frac{\prod_{i=1}^{6} p_i^{y_i}\left(1-p_i\right)^{n_i-y_i}}{f\left(\mathbf{p}_{(-i)}|\boldsymbol{y},\right)}, \quad \text{provided} \quad p_{i-1} < p_i < p_{i+1},$$

and

$$f\left(p_i | \boldsymbol{y}, \mathbf{p}_{(-i)}\right) = 0, \quad \text{otherwise.}$$

Moreover,

$$f\left(\mathbf{p}_{(-i)}|\boldsymbol{y}\right) \ltimes \int_{p_{i-1}}^{p_{i+1}} f\left(\mathbf{p}|\boldsymbol{y}\right) dp_i$$

$$\ltimes \left(\prod_{j\neq i} p_j^{y_j}\left(1-p_j\right)^{n_j-y_j}\right)\int_{p_{i-1}}^{p_{i+1}} p_i^{y_i}\left(1-p_i\right)^{n_i-y_i} dp_i$$

So, for $p_0 < p_1 < p_2 < \cdots < p_{i-1} < p_{i+1} < \cdots < p_k < p_7$, and $p_{i-1} < p_i < p_{i+1}$

$$f\left(p_i|\boldsymbol{y},\mathbf{p}_{(-i)}\right) \ltimes \frac{\prod_{j=1}^{6} p_j^{y_j}\left(1-p_i\right)^{n_j-y_j}}{\left(\prod_{j\neq i} p_j^{y_j}\left(1-p_i\right)^{n_j-y_j}\right)\int_{p_{i-1}}^{p_{i+1}} p_i^{y_i}\left(1-p_i\right)^{n_i-y_i} dp_i}$$

$$\Rightarrow f\left(p_i|\boldsymbol{y},\mathbf{p}_{(-i)}\right) \ltimes \frac{p_i^{y_i}\left(1-p_i\right)^{n_i-y_i}}{\int_{p_{i-1}}^{p_{i+1}} p_i^{y_i}\left(1-p_i\right)^{n_i-y_i} dp_i}, \quad p_{i-1} < p_i < p_{i+1}$$

Therefore, $f\left(p_i|\boldsymbol{y},\mathbf{p}_{(-i)}\right)$ is a $Beta\left(y_i+1, n_i-y_i+1\right)$, conditional on the event

$$B = \left\{p_{i-1} < p_i < p_{i+1}\right\}.$$

5. The key issue now is to generate random variables p_i from

$$Beta\left(y_i+1, n_i-y_i+1\right),$$

conditional on the event

$$B = \left\{p_{i-1} < p_i < p_{i+1}\right\}.$$

This is rather simple: we keep generating random variables

$$z_i \sim Beta\left(y_i+1, n_i-y_i+1\right)$$

until we get one value that satisfies the condition

$$p_{i-1} < z_i < p_{i+1}.$$

Then set

$$p_i = z_i.$$

```
######################################
## This is an R function to performs the MCMC calculations

 mh=function(n,y,N0=0,N)    {

 NN=N+N0
 res=matrix(0,NN,6)
 res[1,]=sort(y/n)

 for(i in 2:NN){

        for(j in 1:6){


 if(j == 1){
        yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)
        while(yy >=res[i-1,2]){yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)}
        res[i,j]=yy
          }
 else { if(j == 6){
        yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)
        while(yy <=res[i,5]){yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)}
        res[i,j]=yy

      }
 else {
        yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)
        while(yy <= res[i,j-1] || yy >= res[i-1,j+1]){yy=rbeta(1,shape1=y[j]+1,shape2=n[j]-y[j]+1)}
        res[i,j]=yy
       }
       }
                    }

            #print(c(i,res[i,]))
                    }
return(res[(N0+1):NN,])
 }

y=c(2,3,3,5,4,6)
n=c(20,24,18,25,12,14)

res=mh(n,y,N=1000,N0=50)
boxplot(res,bycol=T)
```