Bayesian Methods with Monte Carlo Markov Chains II

Henry Horng-Shing Lu
Institute of Statistics
National Chiao Tung University

hslu@stat.nctu.edu.tw
http://tigpbp.iis.sinica.edu.tw/courses.htm
Part 3
An Example in Genetics
Example 1 in Genetics (1)

- Two linked loci with alleles A and a, and B and b
  - A, B: dominant
  - a, b: recessive
- A double heterozygote AaBb will produce gametes of four types: AB, Ab, aB, ab

F (Female)  \[1 - r'\]
M (Male)  \[1 - r\]
r' (female recombination fraction)
r (male recombination fraction)
Example 1 in Genetics (2)

- $r$ and $r'$ are the recombination rates for male and female.
- Suppose the parental origin of these heterozygote is from the mating of $AABB \times aabb$. The problem is to estimate $r$ and $r'$ from the offspring of selfed heterozygotes.
- [http://www2.isye.gatech.edu/~brani/isyebayes/bank/handout12.pdf](http://www2.isye.gatech.edu/~brani/isyebayes/bank/handout12.pdf)
### Example 1 in Genetics (3)

<table>
<thead>
<tr>
<th>Female</th>
<th>Male</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AB</td>
</tr>
<tr>
<td>AB</td>
<td>(1-r)/2</td>
</tr>
<tr>
<td>aB</td>
<td>(1-r)/2</td>
</tr>
<tr>
<td>aB</td>
<td>r/2</td>
</tr>
<tr>
<td>aB</td>
<td>r/2</td>
</tr>
<tr>
<td>aB</td>
<td>r/2</td>
</tr>
</tbody>
</table>
Example 1 in Genetics (4)

- Four distinct phenotypes: $A^*B^*$, $A^*b^*$, $a^*B^*$ and $a^*b^*$.
  - $A^*$: the dominant phenotype from $(Aa, AA, aA)$.
  - $a^*$: the recessive phenotype from $aa$.
  - $B^*$: the dominant phenotype from $(Bb, BB, bB)$.
  - $b^*$: the recessive phenotype from $bb$.
  - $A^*B^*$: 9 gametic combinations.
  - $A^*b^*$: 3 gametic combinations.
  - $a^*B^*$: 3 gametic combinations.
  - $a^*b^*$: 1 gametic combination.
  - Total: 16 combinations.
Example 1 in Genetics (5)

Let $\phi = (1 - r)(1 - r')$, then

\[ P(A^* B^*) = \frac{2 + \phi}{4}, \]

\[ P(A^* b^*) = P(a^* B^*) = \frac{1 - \phi}{4}, \]

\[ P(a^* b^*) = \frac{\phi}{4}. \]
Example 1 in Genetics (6)

Hence, the random sample of $n$ from the offspring of selfed heterozygotes will follow a multinomial distribution:

$$\text{Multinomial} \left[ n; \frac{2 + \varphi}{4}, \frac{1 - \varphi}{4}, \frac{1 - \varphi}{4}, \frac{\varphi}{4} \right].$$

We know that $\varphi = (1-r)(1-r')$, $0 \leq r \leq 1/2$, and $0 \leq r' \leq 1/2$. So, $1 \geq \varphi \geq 1/4$. 
Bayesian for Example 1 in Genetics (1)

- To simplify computation, we let
  \[ P(A^*B^*) = \varphi_1, P(A^*b^*) = \varphi_2 \]
  \[ P(a^*B^*) = \varphi_3, P(a^*b^*) = \varphi_4 \]

- The random sample of \( n \) from the offspring of selfed heterozygotes will follow a multinomial distribution:
  \[ \text{Multinomial} \left[ n; \varphi_1, \varphi_2, \varphi_3, \varphi_4 \right]. \]
Bayesian for Example 1 in Genetics (2)

- If we assume a Dirichlet prior distribution with parameters: \( D(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \) to estimate probabilities for \( A*B*, A*b*, a*B* \) and \( a*b* \).

- Recall that
  - \( A*B* \): 9 gametic combinations.
  - \( A*b* \): 3 gametic combinations.
  - \( a*B* \): 3 gametic combinations.
  - \( a*b* \): 1 gametic combination.

We consider \( (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (9, 3, 3, 1) \).
Suppose that we observe the data of \( y = (y_1, y_2, y_3, y_4) = (125, 18, 20, 24) \).

So the \textbf{posterior} distribution is also Dirichlet with parameters

\[
D(134, 21, 23, 25)
\]

The Bayesian estimation for probabilities are: 

\[
(\phi_1, \phi_2, \phi_3, \phi_4) = (0.660, 0.103, 0.113, 0.123)
\]
Consider the original model, 

\[ P(A^* B^*) = \frac{2+\varphi}{4}, P(A^* b^*) = P(a^* B^*) = \frac{1-\varphi}{4}, P(a^* b^*) = \frac{\varphi}{4}. \]

The random sample of \( n \) also follow a multinomial distribution:

\[ (y_1, y_2, y_3, y_4) \sim \text{Multinomial} \left[ n; \frac{2+\varphi}{4}, \frac{1-\varphi}{4}, \frac{1-\varphi}{4}, \frac{\varphi}{4} \right]. \]

We will assume a Beta prior distribution: \( \text{Beta}(\nu_1, \nu_2). \)
Bayesian for Example 1 in Genetics (5)

- The posterior distribution becomes

\[ P(\phi \mid y_1, y_2, y_3, y_4) = \frac{P(y_1, y_2, y_3, y_4 \mid \phi)P(\phi)}{\int P(y_1, y_2, y_3, y_4 \mid \phi)P(\phi)d\phi}. \]

- The integration in the above denominator,

\[ \int P(y_1, y_2, y_3, y_4 \mid \phi)P(\phi)d\phi, \]

does not have a close form.
Bayesian for Example 1 in Genetics (6)

- How to solve this problem?
  Monte Carlo Markov Chains (MCMC) Method!

- What value is appropriate for $(\nu_1, \nu_2)$?
Part 4
Monte Carlo Methods
Monte Carlo Methods (1)

- Consider the game of solitaire: what’s the chance of winning with a properly shuffled deck?


Chance of winning is 1 in 4!
Monte Carlo Methods (2)

- Hard to compute analytically because winning or losing depends on a complex procedure of reorganizing cards
- Insight: why not just play a few hands, and see empirically how many do in fact win?
- More generally, can approximate a probability density function using only samples from that density
Monte Carlo Methods (3)

- Given a very large set $X$ and a distribution $f(x)$ over it.
- We draw a set of $N$ i.i.d. random samples.
- We can then approximate the distribution using these samples.

\[
f_N(x) = \frac{1}{N} \sum_{i=1}^{N} 1(x^{(i)} = x) \xrightarrow{N \to \infty} f(x)
\]
Monte Carlo Methods (4)

- We can also use these samples to compute expectations:
  \[ E_N(g) = \frac{1}{N} \sum_{i=1}^{N} g(x^{(i)}) \to E(g) = \sum_x g(x)f(x) \]

- And even use them to find a maximum:
  \[ \hat{x} = \text{arg max}_{x^{(i)}} [f(x^{(i)})] \]
Monte Carlo Example

- $X_1, \ldots, X_n$ be i.i.d. $N(0,1)$, find $E(X_i^4) =$?

- Solution:

$$E(X_i^4) = \int_{-\infty}^{\infty} x^4 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx = \frac{4!}{2^{4/2} \left(\frac{4}{2}\right)!} = \frac{24}{8} = 3$$

- Use Monte Carlo method to approximation

```R
> x<-rnorm(100000) # 100000 samples from N(0,1)
> x<-x^4
> mean(x)
[1] 3.00567
```
Part 5
Monte Carlo Method vs. Numerical Integration
Numerical Integration (1)

- **Theorem (Riemann Integral):**
  If $f$ is continuous, integrable, then the Riemann Sum:
  \[
  S_n = \sum_{i=1}^{n} f(c_i)(x_{i+1} - x_i), \text{ where } c_i \in [x_i, x_{i+1}]
  \]
  \[
  S_n \xrightarrow{n \to \infty} I = \int_{a}^{b} f(x)dx, \text{ a constant}
  \]

Numerical Integration (2)

- Trapezoidal Rule:

\[ x_i = a + \frac{b - a}{n} i, \text{ equally spaced intervals} \]

\[ x_{i+1} - x_i = h = \frac{b - a}{n} \]

\[ S_T = \sum_{i=1}^{n} \frac{f(x_i) + f(x_i + 1)}{2} h \]

\[ = h\left[\frac{1}{2} f(x_1) + f(x_2) + \ldots + f(x_n) + \frac{1}{2} f(x_{n+1})\right] \]

Numerical Integration (3)

- An Example of Trapezoidal Rule by R:

```R
f <- function(x) ((x-4.5)^3+5.6)/1234
x <- seq(-100, 100, 1)
plot(x, f(x), ylab="f(x)", type='l', lwd=2)

temp <- matrix(0, nrow=6, ncol=4)
temp[, 1] <- seq(-100, 100, 40); temp[, 2] <- rep(f(-100), 5)
temp[, 3] <- temp[, 1]; temp[, 4] <- f(temp[, 1])
segments(temp[, 1], temp[, 2], temp[, 3], temp[, 4], col="Red")
temp <- matrix(0, nrow=5, ncol=4)
temp[, 1] <- seq(-100, 80, 40); temp[, 2] <- f(temp[, 1])
temp[, 3] <- seq(-60, 100, 40); temp[, 4] <- f(temp[, 3])
segments(temp[, 1], temp[, 2], temp[, 3], temp[, 4], col="Red", lwd=2)
```
Numerical Integration (4)

- Simpson’s Rule:
  
  Specifically, \( \int_a^b f(x)dx \approx \frac{b-a}{6}[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)] \)

- One derivation replaces the integrand \( f(x) \) by the quadratic polynomial \( P(x) \) which takes the same values as \( f(x) \) at the end points \( a \) and \( b \) and the midpoint \( m = (a+b)/2 \).

\[
P(x) = f(a)\frac{(x-m)(x-b)}{(a-m)(a-b)} + f(m)\frac{(x-a)(x-b)}{(m-a)(m-b)} + f(b)\frac{(x-a)(x-m)}{(b-a)(b-m)}
\]

\[
\int_a^b P(x) = \frac{b-a}{6}[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)]
\]

Numerical Integration (5)

Example of Simpson’s Rule by R:

```r
f<-function(x) ((x-4.5)^3+100*x^2+5.6)/1234
x<-seq(-100,100,1)

p<-function(x)
{f(-100) * (x-0) * (x-100) / (-100-0) / (-100-100) + f(0) * (x+100) * (x-100)
 + f(100) * (x-0) * (x+100) / (0-100) / (100-0) / (100+100)}

matplot(x,cbind(f(x),p(x)),type="l",lwd=2,lty=1,ylab="")

temp<-matrix(0,nrow=3,ncol=4)
temp[,1]<-seq(-100,100,100);temp[,2]<-rep(p(-60),3)
temp[,3]<-temp[,1];temp[,4]<-f(temp[,1])
segments(temp[,1],temp[,2],temp[,3],temp[,4],col="Red",lwd=2)
```
Numerical Integration (6)

- Two Problems in numerical integration:

  1. \((b, a) = (\infty, -\infty)\), i.e. \(\int_{-\infty}^{\infty} f(x)dx\)

     How to use numerical integration?

     Logistic transform: \(y = \frac{1}{1 + e^x}\)

  2. Two or more high dimension integration? Monte Carle Method!

Monte Carlo Integration

- \[ I = \int f(x)dx = \int \frac{f(x)}{g(x)} g(x)dx, \]

where \( g(x) \) is a probability distribution function and let \( h(x) = \frac{f(x)}{g(x)} \).

- If \( x_1, \ldots, x_n \sim g(x) \), then

\[
\frac{1}{n} \sum_{i=1}^{n} h(x_i) \xrightarrow{LLN} E(h(x)) = \int h(x)g(x)dx = \int f(x)dx = I
\]

- \url{http://en.wikipedia.org/wiki/Monte_Carlo_integration}
Example (1)

\[ \int_{0}^{1} \frac{\Gamma(18)}{\Gamma(7)\Gamma(11)} x^{10} (1 - x)^{10} \, dx = ? \]

(It is \( \frac{2}{57} \approx 0.0350877! \))

- Numerical Integration by Trapezoidal Rule:

The numerical integration is 0.0350866. Press any key to continue.
Example (2)

Monte Carlo Integration:

\[ \int_{0}^{1} \frac{\Gamma(18)}{\Gamma(7)\Gamma(11)} x^{10} (1-x)^{10} \, dx = \int_{0}^{1} x^{4} \frac{\Gamma(7 + 11)}{\Gamma(7)\Gamma(11)} x^{7-1} (1-x)^{11-1} \, dx \]

Let \( x \sim \text{Beta}(7, 11) \), then

\[ \int_{0}^{1} x^{4} \frac{\Gamma(7 + 11)}{\Gamma(7)\Gamma(11)} x^{7-1} (1-x)^{11-1} \, dx = E(x^{4}) \]

\[ \frac{1}{n} \sum_{i=1}^{n} x_{i}^{4} \xrightarrow{LLN} E(x^{4}) = \int_{0}^{1} \frac{\Gamma(18)}{\Gamma(7)\Gamma(11)} x^{10} (1-x)^{10} \, dx \]
#include <imsls.h>

#define function(x) (pow(x,10.)*pow((1-x),10.))

void main()
{
    int i, iN=10000;
    double dLow=0.0, dUpper=1.0;
    double fLwrA, fUprA, fAvgA;
    double delta;
    float dconstant;
    double xi1, xi2;

    dconstant=imsls_f_gamma(18.0)/imsls_f_gamma(7.0)/imsls_f_gamma(11.0);

    delta = (dUpper-dLow)/iN;
    fLwrA=0.0; fUprA=0.0; fAvgA=0.0;
    for(i=0; i<iN; i++)
    {
        xi1 = dLow + (delta*i);
        fLwrA += delta * dconstant*function(xi1);

        xi2 = dLow + (delta*(i+1));
        fUprA += delta * dconstant*function(xi2);
        fAvgA = (fLwrA+fUprA)/2.0;
    }

    printf("The numerical integration is %lf .\n", fAvgA);
}

> x<-rbeta(10000,7,11)
> x<-x^4
> mean(x)
[1] 0.03551215
Part 6
Markov Chains
Markov Chains (1)

- A **Markov chain** is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by transition probability.

- Suppose the random variable $X_0, X_1, \ldots$ take **state space** ($\Omega$) that is a countable set of value. A **Markov chain** is a process that corresponds to the network.
Markov Chains (2)

- The current state in Markov chain only depends on the most recent previous states.

\[ P(X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) \]

\[ = P(X_{t+1} = j \mid X_t = i) = \text{Transition Probability} \]

where \((i_0, \ldots, i, j) \in \Omega\)

- [http://civs.stat.ucla.edu/MCMC/MCMC_tutorial/Lect1_MCMC_Intro.pdf](http://civs.stat.ucla.edu/MCMC/MCMC_tutorial/Lect1_MCMC_Intro.pdf)
An Example of Markov Chains

- $\Omega = (1, 2, 3, 4, 5)$
- $X = (X_0, X_1, ..., X_t, ...) \in \Omega$
  where $X_0$ is initial state and so on.
- $P$ is transition matrix.

\[
P = \begin{bmatrix}
1 & 0.4 & 0.6 & 0.0 & 0.0 & 0.0 \\
2 & 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\
3 & 0.0 & 0.3 & 0.0 & 0.7 & 0.0 \\
4 & 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \\
5 & 0.0 & 0.3 & 0.0 & 0.5 & 0.3
\end{bmatrix}
\]
**Definition (1)**

- Define the probability of going from state $i$ to state $j$ in $n$ time steps as

  $$p_{ij}^{(n)} = P(X_{t+n} = j | X_t = i)$$

- A state $j$ is **accessible** from state $i$ if there are $n$ time steps such that $p_{ij}^{(n)} > 0$, where $n=0,1,...$

- A state $i$ is said to **communicate** with state $j$ (denote: $i \leftrightarrow j$), if it is true that both $i$ is accessible from $j$ and that $j$ is accessible from $i$. 
Definition (2)

- A state $i$ has **period** $d(i)$ if any return to state $i$ must occur in multiples of $d(i)$ time steps.
- Formally, the period of a state is defined as
  
  $$d(i) = \gcd\{n : P_{ii}^{(n)} > 0\}$$

- If $d(i)= 1$, then the state is said to be **aperiodic**; otherwise ($d(i)>1$), the state is said to be **periodic with period** $d(i)$. 
Definition (3)

- A set of states $C$ is a **communicating class** if every pair of states in $C$ communicates with each other.
- Every state in a communicating class must have the same period.
- Example:
Definition (4)

- A finite Markov chain is said to be **irreducible** if its state space ($\Omega$) is a communicating class; this means that, in an irreducible Markov chain, it is possible to get to any state from any state.

- Example:
Definition (5)

- A finite state irreducible Markov chain is said to be **ergodic** if its states are aperiodic.

Example:
Definition (6)

- A state $i$ is said to be **transient** if, given that we start in state $i$, there is a non-zero probability that we will never return back to $i$.

- Formally, let the random variable $T_i$ be the next return time to state $i$ (the “hitting time”):
  $$T_i = \min\{n : X_n = i \mid X_0 = i\}$$

- Then, state $i$ is transient iff there exists a finite $T_i$ such that: $P(T_i < \infty) < 1$
Definition (7)

- A state $i$ is said to be **recurrent** or **persistent** iff there exists a finite $T_i$ such that: $P(T_i < \infty) = 1$.
- The mean recurrent time $\mu_i = E[T_i]$.
- State $i$ is **positive recurrent** if $\mu_i$ is finite; otherwise, state $i$ is **null recurrent**.
- A state $i$ is said to be **ergodic** if it is aperiodic and positive recurrent. If all states in a Markov chain are ergodic, then the chain is said to be ergodic.
Stationary Distributions

- **Theorem**: If a Markov Chain is irreducible and aperiodic, then

  \[ P_{ij}^{(n)} \rightarrow \frac{1}{\mu_j} \text{ as } n \rightarrow \infty, \ \forall i, j \in \Omega \]

- **Theorem**: If a Markov chain is irreducible and aperiodic, then \( \exists! \pi_j = \lim_{n \rightarrow \infty} P(X_n = j) \)

  and \( \pi_j = \sum_{i \in \Omega} \pi_i P_{ij}, \ \sum_{i \in \Omega} \pi_j = 1, \ \forall j \in \Omega \)

  where \( \pi \) is stationary distribution.
Definition (7)

- A Markov chain is said to be **reversible**, if there is a stationary distribution $\pi$ such that

\[ \pi_i P_{ij} = \pi_j P_{ji} \quad \forall i, j \in \Omega \]

- **Theorem**: if a Markov chain is reversible, then

\[ \pi_j = \sum_{i \in \Omega} \pi_i P_{ij} \]
An Example of Stationary Distributions

- A Markov chain:

\[ P = \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix} \]

- The stationary distribution is \( \pi = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \)

\[
\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}
\]
Properties of Stationary Distributions

- Regardless of the starting point, the process of irreducible and aperiodic Markov chains will converge to a stationary distribution.

- The rate of converge depends on properties of the transition probability.
Part 7
Monte Carlo Markov Chains
Applications of MCMC

- **Simulation:**
  Ex: \((x, y) \sim f(x, y) = c \frac{n!}{x!} y^{x+\alpha-1} (1-y)^{n-x+\beta-1},\)
  where \(x = 0, 1, 2, \ldots, n, \ 0 \leq y \leq 1, \ \alpha, \beta \) are known.

- **Integration:** computing in high dimensions.
  Ex:
  \[
  E(g(y)) = \int_0^1 g(y) f(y) dy.
  \]

- **Bayesian Inference:**
  Ex: Posterior distributions, posterior means...
Monte Carlo Markov Chains

- **MCMC** method are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution.
- The state of the chain after a large number of steps is then used as a sample from the desired distribution.
Inversion Method vs. MCMC (1)

- **Inverse transform sampling**, also known as the probability integral transform, is a method of sampling a number at random from any probability distribution given its cumulative distribution function (cdf).

Inversion Method vs. MCMC (2)

- A random variable with a cdf $F$, then $F$ has a uniform distribution on $[0, 1]$.
- The inverse transform sampling method works as follows:
  1. Generate a random number from the standard uniform distribution; call this $u$.
  2. Compute the value $x$ such that $F(x) = u$; call this $x_{\text{chosen}}$.
  3. Take $x_{\text{chosen}}$ to be the random number drawn from the distribution described by $F$. 
Inversion Method vs. MCMC (3)

- For one dimension random variable, Inversion method is good, but for two or more high dimension random variables, Inverse Method maybe not.

- For two or more high dimension random variables, the marginal distributions for those random variables respectively sometime be calculated difficult with more time.
Gibbs Sampling

- One kind of the MCMC methods.
- The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution rather than integrating over a joint distribution.
Example 1 (1)

- To sample $x$ from:

$$ (x, y) \sim f(x, y) = c \binom{n}{x} y^{x+\alpha-1} (1 - y)^{n-x+\beta-1}, $$

where $x = 0, 1, 2, ..., n$, $0 \leq y \leq 1$, $n, \alpha, \beta$ are known $c$ is a constant.

- One can see that

$$ f(x \mid y) = \frac{f(x, y)}{f(y)} \propto \binom{n}{x} y^x (1 - y)^{n-x} \sim \text{Binomial}(n, y), $$

$$ f(y \mid x) = \frac{f(x, y)}{f(x)} \propto y^{x+\alpha-1} (1-y)^{n-x+\beta-1} \sim \text{Beta}(x+\alpha, n-x+\beta). $$
Example 1 (2)

- Gibbs sampling Algorithm:
  - Initial Setting:
    \[
    \begin{cases}
    y_0 &\sim\text{Uniform}[0,1] \text{ or a arbitrary value } \in [0,1] \\
    x_0 &\sim\text{Bin}(n, y_0)
    \end{cases}
    \]
  - For \(t=0,\ldots,n\), sample a value \((x_{t+1}, y_{t+1})\) from
    \[
    \begin{cases}
    y_{t+1} &\sim\text{Beta}(x_t + \alpha, n - x_t + \beta) \\
    x_{t+1} &\sim\text{Bin}(n, y_{t+1})
    \end{cases}
    \]
  - Return \((x_n, y_n)\)
Example 1 (3)

- Under regular conditions: \( x_t \xrightarrow{t \to \infty} x, y_t \xrightarrow{t \to \infty} y \).
- How many steps are needed for convergence?
  - Within an acceptable error, such as
    \[
    \left| \sum_{t=i+1}^{i+10} x_t - \sum_{t=i+11}^{i+20} x_t \right| < 0.001, \ \forall i \in \mathbb{N}.
    \]
  - \( n \) is large enough, such as \( n=1000 \).
Example 1 (4)

- Inversion Method:
  - $x$ is Beta-Binomial distribution.
    \[ x \sim f(x) = \int_0^1 f(x, y) dy \]
    \[ = \binom{n}{x} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \frac{\Gamma(x + \alpha) \Gamma(n - x + \beta)}{\Gamma(n + \alpha + \beta)} \]
  - The cdf of $x$ is $F(x)$ that has a uniform distribution on $[0, 1]$.
    \[ F(x) = \sum_{i=0}^{x} \binom{n}{i} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \frac{\Gamma(i + \alpha) \Gamma(n - i + \beta)}{\Gamma(n + \alpha + \beta)} \]
Gibbs sampling by R (1)

N=1000; num=16; alpha=5; beta=7

tempy<-runif(1); tempx<-rbeta(1, beta, alpha)
j=0
Forward=1; Afterward=0

while( abs(Forward-Afterward) > 0.0001 && j <= 1000 )
{
    Forward=Afterward;
    Afterward=0;
    for( i in 1:N )
    {
        tempy<-rbeta(1, num-tempx+beta, tempx+alpha)
        tempx<-rbinom(1, num, tempy)
        Afterward=Afterward+tempx
    }
    Afterward=Afterward/N
    j=j+1
}
Gibbs sampling by R (2)

```r
sample<-matrix(0,nrow=N,ncol=2)
for(i in 1:N)
{
    tempy<-rbeta(1,tempx+alpha,num-tempx+beta)
    tempx<-rbinom(1,num,tempy)
    sample[i,1]=tempx
    sample[i,2]=tempy
}

sample_Inverse<-rbetabin(N,num,alpha,beta)
write(t(sample),"Sample for Ex1 by R.txt",ncol=2)

Xhist<-cbind(hist(sample[,1],nclass=num)$count,
              hist(sample_Inverse,nclass=num)$count)
write(t(Xhist),"Histogram for Ex1 by R.txt",ncol=2)
```
Gibbs sampling by R (3)

prob<-matrix(0,nrow=num+1,ncol=2)
for(i in 0:num)
{
    if(i==0)
    {
        prob[i+1,2]=mean(pbinom(i,num,sample[,2]))
        prob[i+1,1]=gamma(alpha+beta)*gamma(num+beta)
        prob[i+1,1]=prob[i+1,1]/(gamma(beta)*gamma(num+beta+alpha))
    }
    else
    {
        if(i==1)
        {
            prob[i+1,1]=num*alpha/(num-i+alpha+beta);
            for(j in 0:(num-2))
                prob[i+1,1]=prob[i+1,1]*(beta+j)/(alpha+beta+j)
        }
        else
            prob[i+1,1]=prob[i+1,1]*(num-i+1)/(i)*(i-1+alpha)/(num-i+beta)

        prob[i+1,2]=mean((pbinom(i,num,sample[,2])-pbinom(i-1,num,sample[,2])))
    }
    if(i!=num)
        prob[i+2,1]=prob[i+1,1]
}
write(t(prob),"ProbHistogram for Ex1 by R.txt",ncol=2)
Inversion Method by R

rbetabin<-function(N, size, alpha, beta)
{
    Usample<-runif(N)

    Pr_0=gamma(alpha+beta) * gamma(size+beta) / gamma(beta) / gamma(size+beta+alpha)

    Pr=size*alpha/ (size-1+alpha+beta);
    for(i in 0:(size-2))
        Pr=Pr*(beta+i)/ (alpha+beta+i);
    Pr_Initial=Pr;

    sample<-array(0,N)
    CDF<-array(0, (size+1))
    CDF[1]<-Pr_0

    for(i in 1:size)
        {
            CDF[i+1]=CDF[i]+Pr
            Pr=Pr*(size-i)/(i+1) * (i+alpha)/(size-i-1+beta)
        }
    for(i in 1:N)
        {
            sample[i]=which.min(abs(Usample[i]-CDF)) - 1
        }
    return(sample)
}
Gibbs sampling by C/C++ (1)

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <imsls.h>

#include "Distribution.h"

FILE *Output;

void main()
{
    int i,j,iN,iNum, iTempX=0, iTempN=0, iTempNumA=0, iTempNumB=0;
    int *pBetaBin_Sample, *pCondBin;
    double dTemp=0.0, dSumForward=0.0, dSumAfterward=0.0, dAlpha, dBeta, dTempY=0.0;
    double dTempPrA=0.0, dTempPrB=0.0, dTempX=0.0;
    double *pCondBeta;

    //***************************************************************************
    //Example 1***************************************************************************

    iN=1000;  //1000 samples
    iNum=16;
    dAlpha=5.0;
    dBeta=7.0;

    pBetaBin_Sample=(int *)malloc(iN*sizeof(int));
    pCondBin=(int *)malloc(iN*sizeof(int));
    pCondBeta=(double *)malloc(iN*sizeof(double));
```
Gibbs sampling by C/C++ (2)

```c
for(i=0;i<iN;i++)
{
    pBetaBin_Sample[i]=0;
    pCondBin[i]=0;
    pCondBeta[i]=0.0;
}
BetaBinomlial(pBetaBin_Sample,iNum,dAlpha,dBeta,iN,&dTemp,0);

Output=fopen("Sample for Ex1.txt","w");

Uniform(&dTempY,0,1,1,&dTemp,0); // sample for uniform(0,1)
Binomial(&iTempX,iNum,dTempY,1,&dTemp,0); // sample for binomial(iNum,y0)

i=0;
dSumForward=0.0;  //Check that Gibbs sampling converge
dSumAfterward=0.0;  //Check that Gibbs sampling converge
```

Inversion method
Gibbs sampling by C/C++ (3)

do
{
    dSumForward=dSumAfterward;
    dSumAfterward=0.0;

    for(i=0;i<iN;i++)
    {
        Beta_Dist(&dTempY,((double)iTempX+dAlpha),((double)(iNum-iTempX)+dBeta),1,&dTemp,0);
        Binomial(&iTempX,iNum,dTempY,1,&dTemp,0);
        dSumAfterward+=((double)iTempX;
    }
    dSumAfterward=dSumAfterward/(double)iN;
    j++;
}

}while(fabs(dSumForward-dSumAfterward)>0.0001&j<=1000);

i=0;
for(i=0;i<iN;i++)
{
    Beta_Dist(&dTempY,(double)iTempX+dAlpha,(double)(iNum-iTempX)+dBeta,1,&dTemp,0);
    Binomial(&iTempX,iNum,dTempY,1,&dTemp,0);
    pCondBeta[i]=dTempY;
    pCondBin[i]=iTempX;
    fprintf(Output,"%d %d\n",iTempX,dTempY);
}

fclose(Output);
Gibbs sampling by C/C++ (4)

```c
Output=fopen("Histogram for Ex1.txt","w");
//Count for Histogram
iTempNumA=0;
iTempNumB=0;
for(i=0;i<iNum;i++)
{
    for(j=0;j<iN;j++)
    {
        if(pBetaBin_Sample[j]==i)
            iTempNumA++;
        if(pCondBin[j]==i)
            iTempNumB++;
    }
    fprintf(Output,"%d %d\n",iTempNumA,iTempNumB);
    iTempNumA=0;
iTempNumB=0;
}
fclose(Output);
Output=fopen("ProHistogram for Ex1.txt","w");

//Calculate for Probability Histogram
dTempPrA=0.0;
dTempPrB=0.0;
```
Gibbs sampling by C/C++ (5)

```c
for(i=0;i<=iNum;i++)
{
    if(i==0)
    {
        for(j=0;j<iN;j++)
            dTempPrA+=imsls_d_binomial_cdf(0,iNum,pCondBeta[j]);
        dTempPrA=dTempPrA/(double)iN;
        dTempPrB=imsls_d_gamma (dAlpha+dBeta)*imsls_d_gamma ((double)iNum+dBeta);
        dTempPrB/=imsls_d_gamma (dBeta)*imsls_d_gamma ((double)iNum+dBeta+dAlpha);
    }
    else
    {
        if(i==1)
        {
            dTempPrB=(double)iNum*dAlpha/((double)iNum-1.0+dAlpha+dBeta);
            for(j=0;j<iNum-1;j++)
                dTempPrB*=(dBeta+(double)j)/((dAlpha+dBeta+(double)j));
        }
        else
        {
            dTempPrB*=(double)(iNum-i+1)/(double)(i)*((double)i-1.0+dAlpha);
            dTempPrB/=((double)iNum-(double)i+dBeta);
        }
        for(j=0;j<iN;j++)
            dTempPrA+=imsls_d_binomial_cdf(i,iNum,pCondBeta[j]);
        dTempPrA-=imsls_d_binomial_cdf(i-1,iNum,pCondBeta[j]);
        dTempPrA=dTempPrA/(double)iN;
    }
    fprintf(Output,"%lf %lf\n",dTempPrA,dTempPrB);
    dTempPrA=0.0;
}
fclose(Output);
printf("\nThe Example 1's Data product complete.\n");
```
Inversion Method by C/C++ (1)

```c
void BetaBinomial(int *pArray, int iNum, double dAlpha, double dBeta, int iN, double *dTime, int iInGate) {
    // generate iN samples from Beta-Binomial(N, α, β)
    // Notice: N >= 1 and Integer, α, β > 0

    int i, iTemp;
    double dTemp, dCDF, dCDF_Initial, dPr, dPr_0, dPr_Initial;
    double *pUniform_Sample;

    pUniform_Sample = (double *)calloc(iN, sizeof(double));
    for (i = 0; i < iN; i++)
        pUniform_Sample[i] = 0.;

    Uniform(pUniform_Sample, 0., 1., iN, dTime, 0);

    iTemp = 0;
    dPr_0 = imsls_d_gamma(dAlpha + dBeta) * imsls_d_gamma((double)iNum + dBeta);
    dPr_0 /= imsls_d_gamma(dBeta) * imsls_d_gamma((double)iNum + dBeta + dAlpha);

    dCDF = dPr_0;
    dCDF_Initial = dCDF;

    dPr = (double)iNum * dAlpha / ((double)iNum - 1.0 + dAlpha + dBeta);
    for (i = 0; i < iNum - 1; i++)
        dPr *= (dBeta + (double)i) / (dAlpha + dBeta + (double)i);
    dPr_Initial = dPr;
}
```
for(i=0;i<iN;i++)
{
    if((pUniform_Sample[i]-dCDF)>1.E-10)
    {
        do
        {
            iTemp++;
            dCDF+=dPr;
            dPr*=(double)(iNum-iTemp)/(double)(iTemp+1)*((double)iTemp+dAlpha);
            dPr/=((double)iNum-(double)iTemp-1.0+dBeta);
        }while((pUniform_Sample[i]-dCDF)>1.E-10);
        pArray[i]=iTemp;
    }
    else
        pArray[i]=iTemp;
    iTemp=0;
    dPr=dPr_Initial;
    dCDF=dCDF_Initial;
}

dTime[0]=(t_2-t_1)/(double)CLK_TCK;
free(pUniform_Sample);
Plot Histograms by Maple (1)

- Figure 1: 1000 samples with $n=16$, $\alpha = 5$ and $\beta = 7$. 

**Blue-Inversion method**
**Red-Gibbs sampling**
Plot Histograms by Maple (2)

Plot The Figure - Red is Beta-Binomial, Blue is Gibbs Sampler

> restart:
  with(plots):
  with(plottools):
  Data:=readdata(C:\\Histogram for Ex1.txt",2):
  n:=nops(Data):
  BetaBin:=i->[i-1,0],[i-1,Data[i][1]],[i-2/3,Data[i][1]],[i-2/3,0]:
  Gibbs:=[-1/3+i-1,0],[-1/3+i-1,Data[i][2]],[i-1,Data[i][2]],[i-1,0]]:

  BetaFigure:=display(seq(polygon(evalf(BetaBin(i)),color=red),i=1..n)):
  GibbsFigure:=display(seq(polygon(evalf(Gibbs(i)),color=blue),i=1..n)):
  display([BetaFigure,GibbsFigure]);
Figure 2:
- Blue histogram and yellow line are pmf of $x$.
- Red histogram is $\hat{P}(X = x) = \frac{1}{m} \sum_{i=1}^{m} P(X = x \mid Y_i = y_i)$ from Gibbs sampling.
The probability histogram of blue histogram of Figure 1 would be similar to the blue probability histogram of Figure 2, when the sample size $\rightarrow \infty$.

The probability histogram of red histogram of Figure 1 would be similar to the red probability histogram of Figure 2, when the iteration $n \rightarrow \infty$. 

Probability Histograms by Maple (2)
Probability Histograms by Maple (3)

Plot the figure - Red is Estimates of the marginal distribution of X, Blue is exact Beta-Binomial probability

restart:
with(plots):
with(plottools):
Data:=readdata("C:\\ProHistogram for Ex1.txt",2):
n:=nops(Data):
Estimate:=i->[i-1,0],[i-1,Data[i][1]],[i-2/3,Data[i][1]],[i-2/3,0]]:
BetaBin:=[[1/3+i-1,0],[1/3+i-1,Data[i][2]],[i-1,Data[i][2]],[i-1,0]]:
BetaBinpdf:=x->(16!/(x!((16-x)!))*(GAMMA(5+7)/GAMMA(5)/GAMMA(7))*(GAMMA(x+5)*GAMMA(16-x+7))/GAMMA(16+5+7)):
DensityFigure:=plot(BetaBinpdf(x),x=0..16,color=yellow,thickness=3):
EstimateFigure:=display(seq(polygon(evalf(Estimate(i)),color=red),i=1..n)):
BetaBinFigure:=display(seq(polygon(evalf(BetaBin(i)),color=blue),i=1..n)):
display([[DensityFigure,EstimateFigure,BetaBinFigure]]);
Exercises

- Write your own programs similar to those examples presented in this talk, including Example 1 in Genetics and other examples.

- Write programs for those examples mentioned at the reference web pages.

- Write programs for the other examples that you know.