CPMA 573 - Homework 8

Michael Guyer
Department of Mathematics and Computer Science
Duquesne University, Pittsburgh, PA, USA
March 17, 2015

Exercise 1

Method 1: The following code was used in R for Method 1:

```
M <- 5000
xc <- runif(1,0,1)

xValues <- NULL

for (i in 1:M) {

xstar <- runif(1,xc-0.2,xc+0.2)
while (xstar <= 0 | xstar >= 1) {
xstar <- runif(1,xc-0.2,xc+0.2)
}

xc <- xstar

xValues <- c(xValues, xc)

}
```

This method accepts all values of x^* within the interval (0,1) and so has an acceptance probability of 1, while the necessary lag was found to be 47. Below are the auto-correlation plot and histogram of the uniform (0,1) realizations.

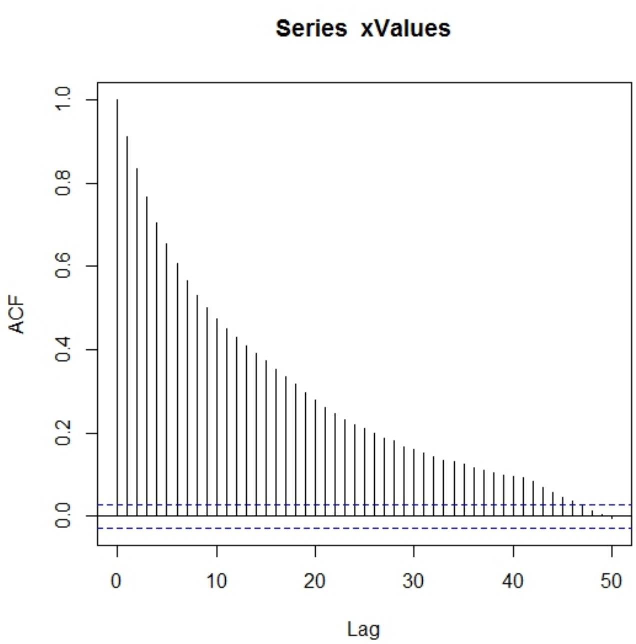

Figure 1: The necessary lag to obtain independent uniform (0,1) realizations using Method 1

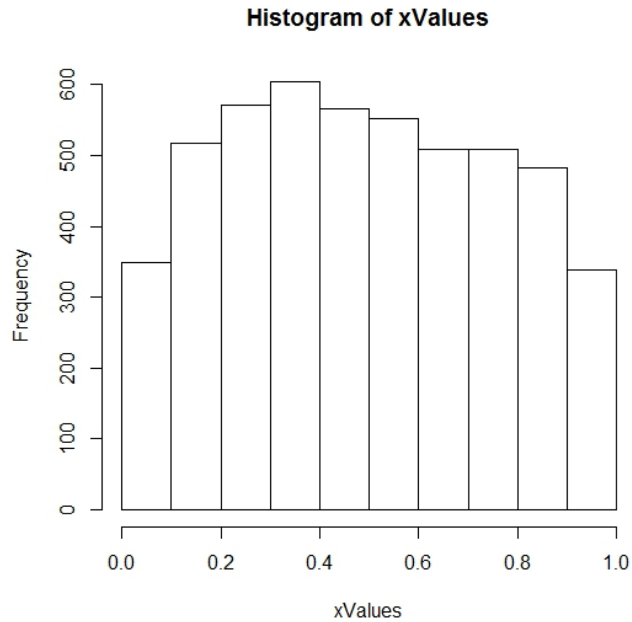

Figure 2: A histogram of the uniform (0,1) realizations obtained using Method 1

The above histogram is non-uniform; in particular it indicates that numbers near the end points of the interval (i.e. 0 and 1) are less likely to be drawn than those numbers in the middle portion of the interval. This is because values of x^* proposed outside of the interval (0,1) are ignored, and so when x_c is near 0 or 1 (in the case of this proposal density within 0.2 of either end point, i.e. $x_c < 0.2$ or $x_c > 0.8$) it is more likely that the next accepted x^* value will be moving closer toward the center of the interval as opposed to closer to whichever end point x_c is currently near. More specifically, if $x_c < 0.2$, then x^* will be less than x_c with probability $\frac{x_c}{x_c+0.2} < \frac{0.2}{x_c+0.2}$, which is the probability that x^* will be greater than x_c. Similarly, if $x_c > 0.8$, then x^* will be greater than x_c with proability $\frac{x_c}{1-x_c+2} < \frac{0.2}{1-x_c+0.2}$, which is the probability that x^* will be less than x_c.

Method 2: The following code was used in R for Method 2:

```
M <- 5000
count2 <- 0
xc <- runif(1,0,1)

xValues2 <- NULL

for (j in 1:M) {

xstar <- runif(1,xc-0.2,xc+0.2)
while (xstar <= 0 | xstar >= 1) {
xstar <- runif(1,xc-0.2,xc+0.2)
}

alpha <- max(1/0.4, 1/(xstar+0.2), 1/(1-xstar+0.2))/max(1/0.4, 1/(xc+0.2), 1/(1-xc+0.2))

if (alpha >= 1) {
xc <- xstar
count2 <- count2 + 1
} else {
x_p <- runif(1,0,1)
if (x_p < alpha) {
xc <- xstar
```

```
count2 <- count2 + 1
}
}

xValues2 <- c(xValues2, xc)
}
```

The acceptance probability was found using the following line in R:

```
count2/5000
```

This acceptance probability was found to be 0.945, while the necessary lag was found to be 61. Below are the auto-correlation plot and histogram of the uniform (0,1) realizations.

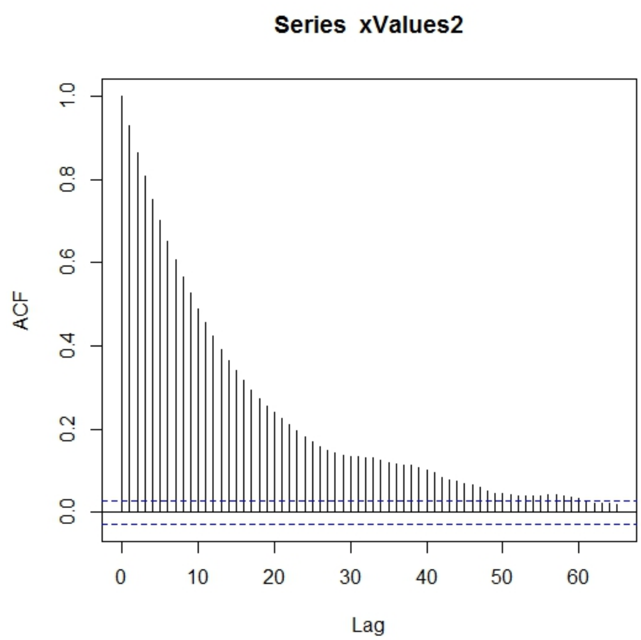

Figure 3: The necessary lag to obtain independent uniform (0,1) realizations using Method 2

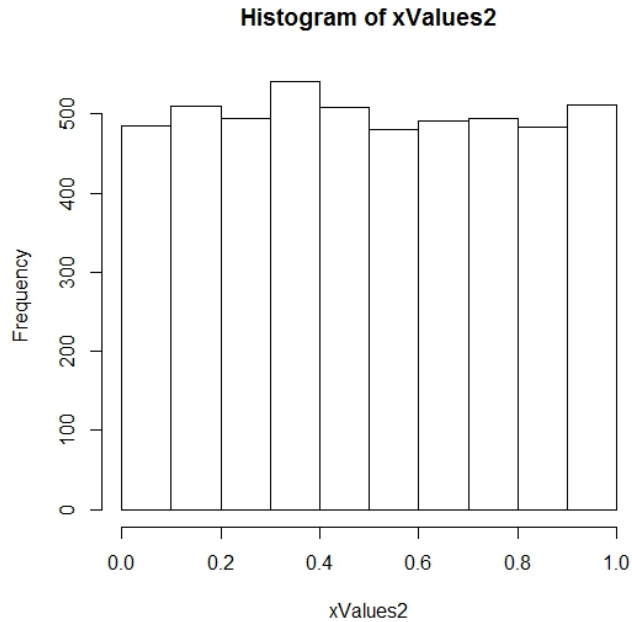

Figure 4: A histogram of the uniform (0,1) realizations obtained using Method 2

Unlke the histogram obtained using Method 1, the above histogram appears to be nearly uniform, indicating that this method (i.e. Metropolis-Hastings sampling) is better at accurately simulating independent uniform (0,1) realizations than Method 1 (i.e. Metropolis sampling).

Exercise 2

The following code in R was used to read in the given data (which was saved to a file) and store both the mean and variance of the sample data:

```
data <- read.table(file.choose(), header=F)
colnames(data) <- NULL
dataAvg <- sapply(data,mean)
dataVar <- sapply(data,var)
```

The following code in R was then used to implement Metropolis-Hastings sampling with normal proposal densities:

```
mu <- dataAvg
sigmaSq <- dataVar

N <- 10000

muValues <- NULL
sigmaSqValues <- NULL

muCount <- 0
sigmaSqCount <- 0

b <- 0.25
c <- 0.25

for (i in 1:N) {

muStar <- rnorm(1,mu,b^2)
```

```
sigmaSqStar <- rnorm(1,sigmaSq,c^2)

muExpComp <- 155*(dataAvg - muStar)^2 + muStar^2/10 - (155*(dataAvg - mu)^2 + mu^2/10)
muAlphaComp <- exp(-muExpComp/(2*sigmaSq))

if (muAlphaComp >= 1) {
mu <- muStar
muCount <- muCount + 1
} else {
mu_p <- runif(1,0,1)
if (mu_p < muAlphaComp) {
mu <- muStar
muCount <- muCount+1
}
}
muValues <- c(muValues, mu)

AStar <- (1/sigmaSqStar)^(155/2+4)
BStar <- (1/(2*sigmaSqStar))*(154*dataVar+155*(dataAvg-muStar)^2+muStar^2/10+2)
A <- (1/sigmaSq)^(155/2+4)
B <- (1/(2*sigmaSq))*(154*dataVar+155*(dataAvg-muStar)^2+muStar^2/10+2)
sigmaSqExpComp <- log(AStar)-BStar-log(A)+B
hastingsNum <- dnorm(sigmaSq,sigmaSqStar,c)/(1-pnorm(0,sigmaSqStar,c))
hastingsDenom <- dnorm(sigmaSqStar,sigmaSq,c)/(1-pnorm(0,sigmaSq,c))
hastings <- hastingsNum/hastingsDenom
sigmaSqAlphaComp <- exp(sigmaSqExpComp)*hastings

if (sigmaSqAlphaComp >= 1) {
sigmaSq <- sigmaSqStar
sigmaSqCount <- sigmaSqCount+1
} else {
sigmaSq_p <- runif(1,0,1)
if (sigmaSq_p < sigmaSqAlphaComp) {
sigmaSq <- sigmaSqStar
sigmaSqCount <- sigmaSqCount+1
}
}
sigmaSqValues <- c(sigmaSqValues, sigmaSq)
}
```

The acceptance probabilities for realizations of μ and σ^2 were found using the following lines in R:

```
muCount/10000
sigmaSqCount/10000
```

These acceptance probabilities were found to be 0.6947 and 0.6907 respectively, while the necessary lags to obtain independent realizations of μ and σ^2 were found to be 12 and 26 respectively. Below are the auto-correlation plots for the μ and σ^2 realizations, as well as the trace plots and histograms of these realizatons.

i.

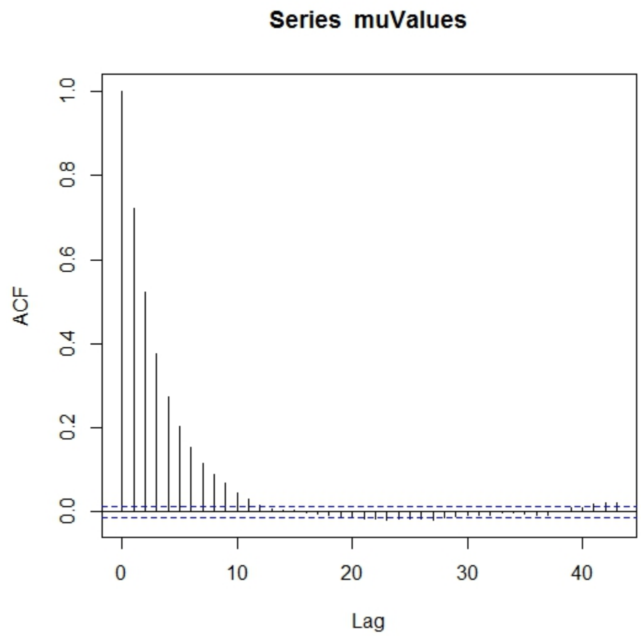

Figure 5: The necessary lag to obtain independent realizations for μ

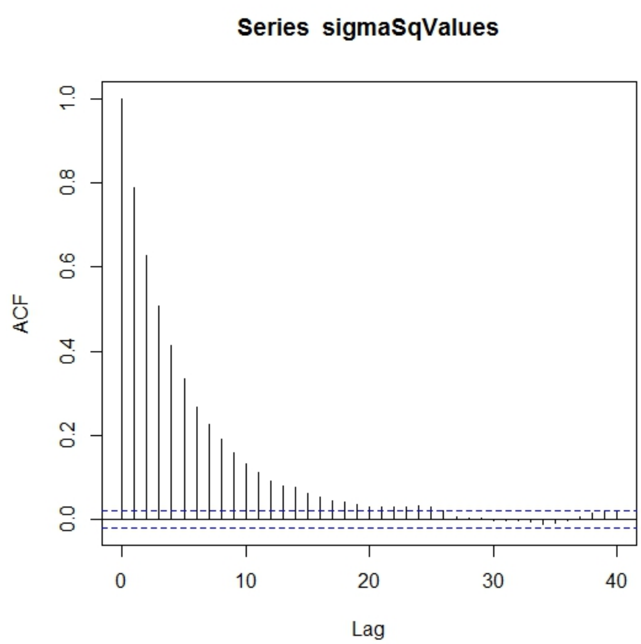

Figure 6: The necessary lag to obtain independent realizations for σ^2

i.

$$\pi(\mu, \sigma^2|\vec{y}) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+4} \cdot e^{-\frac{1}{2\sigma^2}[(n-1)s^2+n(\bar{y}-\mu)^2+\frac{\mu^2}{10}+2]}$$

ii.

$$\pi(\sigma^2|y_1, ..., y_n) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+4} e^{-\frac{1}{2\sigma^2}[(n-1)s^2+2]} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[n(\bar{y}-\mu)^2+\frac{\mu^2}{10}]} d\mu$$

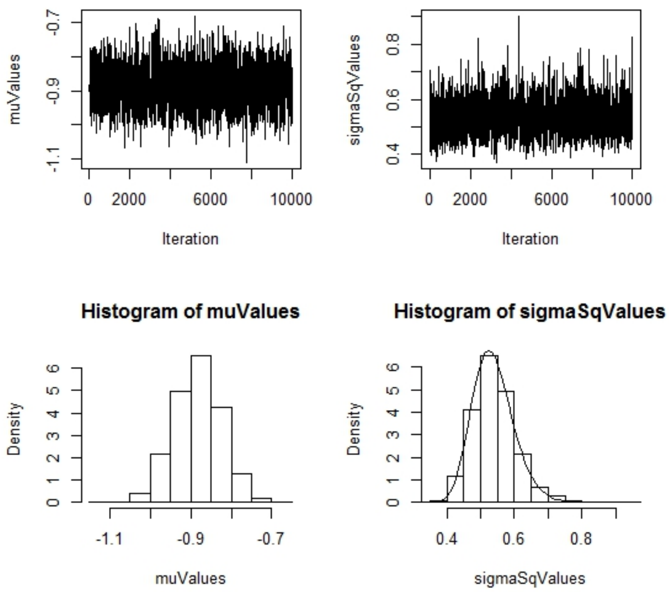

Figure 7: The trace plots and histograms of the μ and σ^2 realizations with the theoretical marginal density of σ^2 superposed

Aside:

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}[n(\bar{y}-\mu)^2+\frac{\mu^2}{10}]}d\mu = \int_{-\infty}^{\infty} e^{-\frac{n}{2\sigma^2}(\bar{y}^2-2\bar{y}\mu+\mu^2+\frac{\mu^2}{10n})}d\mu = \int_{-\infty}^{\infty} e^{-\frac{n(1+\frac{1}{10n})}{2\sigma^2}(\mu^2-\frac{10n}{10n+1}2\bar{y}\mu+\frac{10n}{10n+1}\bar{y}^2)}d\mu$$

$$= \int_{-\infty}^{\infty} e^{-\frac{(n+\frac{1}{10})}{2\sigma^2}[(\mu-\frac{10n}{10n+1}\bar{y})^2+\frac{10n}{10n+1}\bar{y}^2-(\frac{10n}{10n+1})^2\bar{y}^2]}d\mu$$

$$= \sqrt{2\pi\frac{\sigma^2}{n+\frac{1}{10}}}e^{-\frac{(n+\frac{1}{10})}{2\sigma^2}[\frac{10n}{10n+1}\bar{y}^2-(\frac{10n}{10n+1})^2\bar{y}^2]}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\frac{\sigma^2}{n+\frac{1}{10}}}}e^{-\frac{(n+\frac{1}{10})}{2\sigma^2}(\mu-\frac{10n}{10n+1}\bar{y})^2}d\mu$$

$$= \sqrt{2\pi\frac{\sigma^2}{n+\frac{1}{10}}}e^{-\frac{1}{2\sigma^2}[(n-\frac{10n}{10n+1})\bar{y}^2]} = \sqrt{\frac{2\pi}{n+\frac{1}{10}}}\cdot\left(\frac{1}{\sigma^2}\right)^{-\frac{1}{2}}e^{-\frac{1}{2\sigma^2}[(n-\frac{10n}{10n+1})\bar{y}^2]}$$

Thus,

$$\pi(\sigma^2|y_1,...,y_n) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n+5}{2}+1}\cdot e^{\frac{-1}{2\sigma^2}[(n-1)s^2+(n-\frac{10n^2}{10n+1})\bar{y}^2+2]}$$

That is, $\pi(\sigma^2|y_1,...,y_n) \sim \text{IG}(\frac{n+5}{2}, \frac{1}{2}[(n-1)s^2+(n-\frac{10n^2}{10n+1})\bar{y}^2+2])$

iii. μ does not require use of the Hastings ratio, as there are no restrictions on that values that μ can take. On the other hand, realizations of σ^2 cannot be less than 0. The sample variance of the data was found to be 0.5378437, which is close to 0, and so certain choices of c can easily cause values less than 0 to be proposed as σ^{*2} values. Thus, the Hastings ratio must be utilized for σ^2.

iv.

$$\frac{q(\sigma^{*2}|\sigma_0^2)}{q(\sigma_0^2|\sigma^{*2})} = \frac{\frac{1}{\sqrt{2\pi c^2}}e^{-\frac{1}{2c^2}(\sigma^{*2}-\sigma_0^2)^2}\frac{1}{\int_0^\infty \frac{1}{\sqrt{2\pi c^2}}e^{-\frac{1}{2c^2}(x-\sigma_0^2)^2}dx}}{\frac{1}{\sqrt{2\pi c^2}}e^{-\frac{1}{2c^2}(\sigma_0^2-\sigma^{*2})^2}\frac{1}{\int_0^\infty \frac{1}{\sqrt{2\pi c^2}}e^{-\frac{1}{2c^2}(x-\sigma^{*2})^2}dx}}$$

v. In Homework 7, the data used had a sample variance of 25.94479, which is relatively far from 0 and large compared to the values of $c = 1$ and $c = 4$ that were utilized in the proposal densities. Thus, simply using the Metropolis algorithm is sufficient as it is almost a certainty that a negative value of σ^{*2} will not be proposed.