Computer Lab 4

Martynas Lukosevicius

25/11/2020

Question 1: Computation with Metropolis - Hastings

The probability density function is :

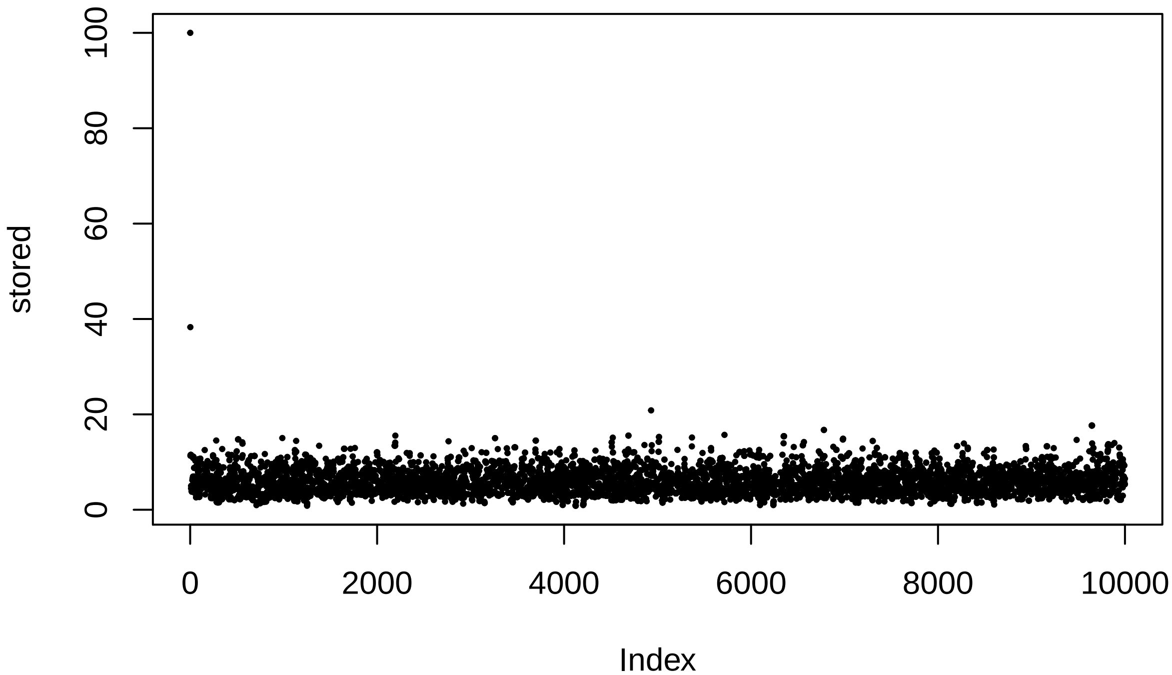
$$f(X) \sim x^5 e^{-x}, x > 0$$

1 Generating samples from target distribution, with metropolis-hastings algorithm, by using Log-normal distribution as proposal distribution.

```r
target <- function(x){
    return((x^5)*exp(-x))}


### density of Log-normal function by sigma=1
##dlnorm gives the density
proposed <- function(x, mu){
  res <- dlnorm(x,log(mu), sd=1)
  return(res)
}
### rlnorm generates random deviates.
metro_hast <- function(startvalue, iteration){
  stored <- rep(startvalue, iteration)
  vN<-1:iteration
  for (i in 2:iteration){
    x <- stored[i-1]
    xprime <- rlnorm(1,log(x),1)

    ratio <- min(c(1,
                   target(xprime) * proposed(x, xprime)) / (target(x) * proposed(xprime, x)))
    accept <- (runif(1) <= ratio)
    stored[i] <- ifelse(accept, xprime, x)
  }
  #return(stored)
  plot(stored,pch=19,cex=0.3)
  #hist(stored[2000:10000] , breaks = 30 )
  #plot(vN,stored,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",
  #ylim=c(min(x-0.5,-5),max(5,x+0.5)))
}
metro_hast(100,10000)
```



Judging by the plot, the burn-in period consists in the first three samples, followed by a period of convergence around Y=6.

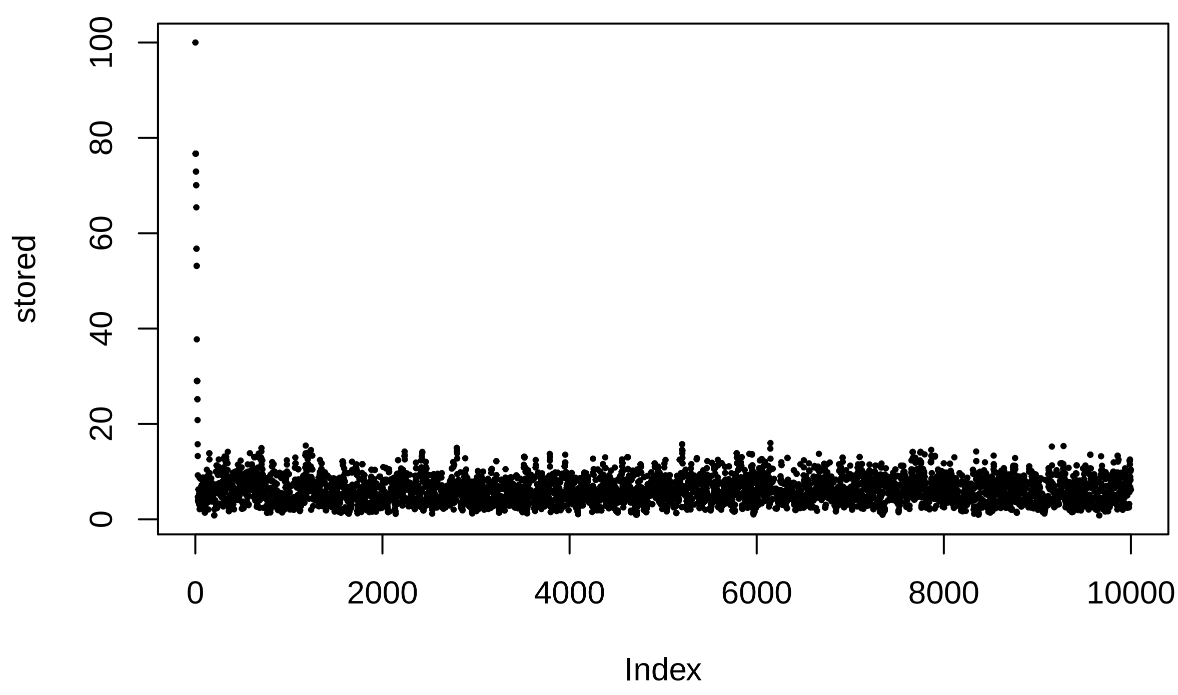
```
metro_hast2 <- function(startvalue, iteration){
  stored <- rep(startvalue, iteration)
  vN<-1:iteration
  for (i in 2:iteration){  ### I am not sure about 2
    x <- stored[i-1]
    xprime <- rlnorm(1,log(x),1)  # proposed a value for start and I am not sure about x=1
    ratio <- min(c(1,
                  target(xprime) * proposed(x, xprime)) / (target(x) * proposed(xprime, x)))
    accept <- (runif(1) <= ratio)
    stored[i] <- ifelse(accept, xprime, x)
  }
  return(stored)
}
```

2 - performing step 1 by using Chi- Square distribution

$$\tilde{\chi}^2([X_t + 1])$$

```
####chi- square distribution
##The dchisq() function gives the density
p_chi_square <- function(x, df){
  return(dchisq(x, df=floor(x+1)))
}
```

```r
metro_chi_hast <- function(startvalue, iteration){
  stored <- rep(startvalue, iteration)
  vN<-1:iteration
  for (i in 2:iteration){  ### I am not sure about 2
    x <- stored[i-1]
    xprime <- rchisq(1,df=floor(x+1))  # proposed a value for start and I am not sure about x=1
    ratio <- min(c(1,
                   target(xprime) * p_chi_square(x, xprime)) / (target(x) * p_chi_square(xprime, x)))
    accept <- (runif(1) <= ratio)
    stored[i] <- ifelse(accept, xprime, x)
  }
  #return(stored)
  plot(stored,pch=19,cex=0.3)
  #hist(stored[5000:10000] , breaks = 30 )
  #plot(vN,stored,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",ylim=c(min(x-0.5,-5),max(5,x+
}
metro_chi_hast(100,10000)
```



3 Comparing result of step1 and step2:

Our plots lead us to thing that our sampling by means of both techniques (normal-logarithm and chi-squared) converge to the same value and have a similar variance just looking at their oscillation. Both arrays present a similar standard deviation around 2.5 and mean value around 6. Therefore, we can conclude by stating both procedures lead to similar results.

4 Generating 10 MCMC sequences using the generator from step2, by using Gelman-Rubin method:

```
### Generate 10 MCMC
p_chi_square <- function(x, df){
  return(dchisq(x, df=floor(x+1)))
}
metro_chi_hast2 <- function(startvalue, iteration){
  stored <- rep(startvalue, iteration)
  vN<-1:iteration
  for (i in 2:iteration){  ### I am not sure about 2
    x <- stored[i-1]
    xprime <- rchisq(1,df=floor(x+1))  # proposed a value for start and I am not sure about x=1
    ratio <- min(c(1,
                   target(xprime) * p_chi_square(x, xprime)) / (target(x) * p_chi_square(xprime, x)))
    accept <- (runif(1) <= ratio)
    stored[i] <- ifelse(accept, xprime, x)
  }
  return(stored)
}
library(coda)
```

```
## Warning: package 'coda' was built under R version 4.0.3
```

```
k <- 10
f1 <- mcmc.list()
for(i in 1:k){
  f1[[i]]<-as.mcmc(metro_chi_hast2(i, 10000))
}
print(gelman.diag(f1))
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

5

Estimate this function by sampling from step1 and step2:

$$\int_0^\infty x.f(x)\,dx$$

```
log_norm <- metro_hast2(100,10000)
chi_square <- metro_chi_hast2(100,10000)
mean(log_norm)
```

```
## [1] 6.003046
```

```r
mean(chi_square)
```

```
## [1] 5.875944
```

6

The value of the integral defined for a Gamma distribution is $\alpha\beta$, being these the parameters alpha and beta of such a kind of distribution. It coincides with its Expected Value. In our case, according to the function of probability density $f(X) \sim x^5 e^{-x}, x > 0$, where $\alpha = 6$ and $\beta = 1$, let $E[Y]$ be:

$$E[Y] = \alpha\beta = 6$$

It is the same as the obtained through Metropolis-Hastings Sampling in the previous two approaches.