Report_lab04

Thijs Quast

7-2-2019

Question 1 - Computations with Metropolis–Hastings

1

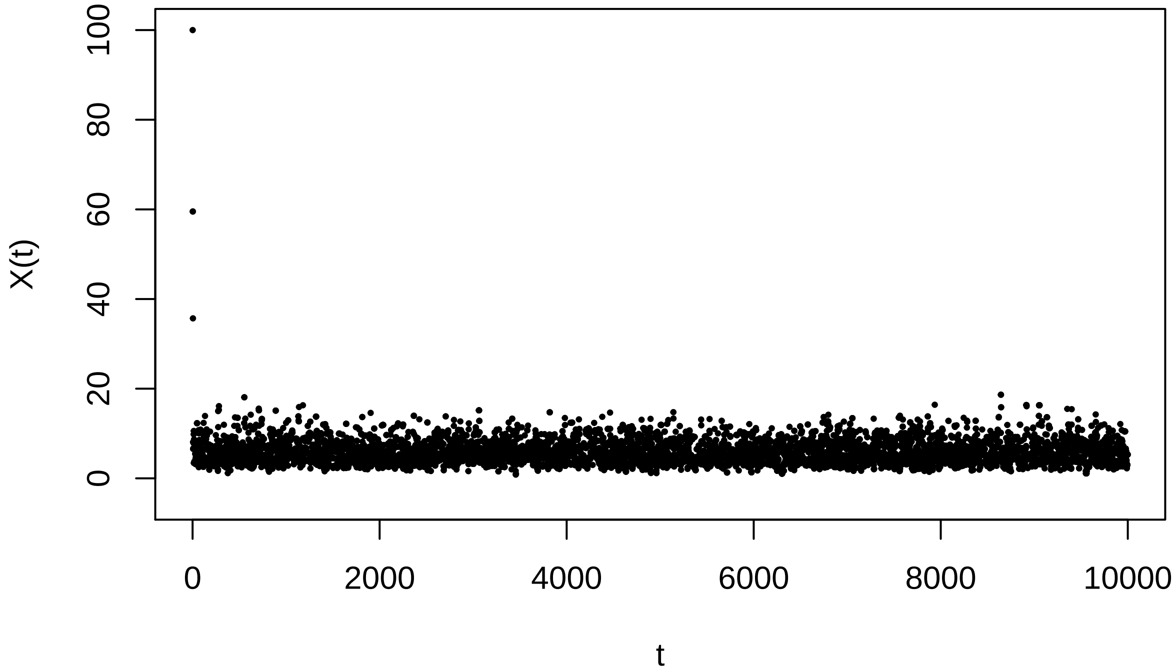
The CDF is provided:

```
f_x <- function(x){
  result <-(x^5)*(exp(-x))
  return(result)
}
```

We can omit the constant here because this will cancel out later on in the Metropolis-Hastings algorithm.

```
f.MCMC.MH<-function(nstep,X0,props){
    vN<-1:nstep
    vX<-rep(X0,nstep);
    for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rlnorm(1, log(X), sdlog = props)
    u<-runif(1)
    a<-min(c(1,(f_x(Y)*dlnorm(X,meanlog=log(Y),sdlog=props))/(f_x(X)*dlnorm(Y,meanlog=log(X),sdlog=props
    if (u <=a){
      vX[i]<-Y
    }else{
        vX[i]<-X
        }
    }
    plot(vN,vX,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",
         ylim=c(min(X0-0.5,-5),max(5,X0+0.5)))
}
```

```
f.MCMC.MH(10000, 100, 1)
```

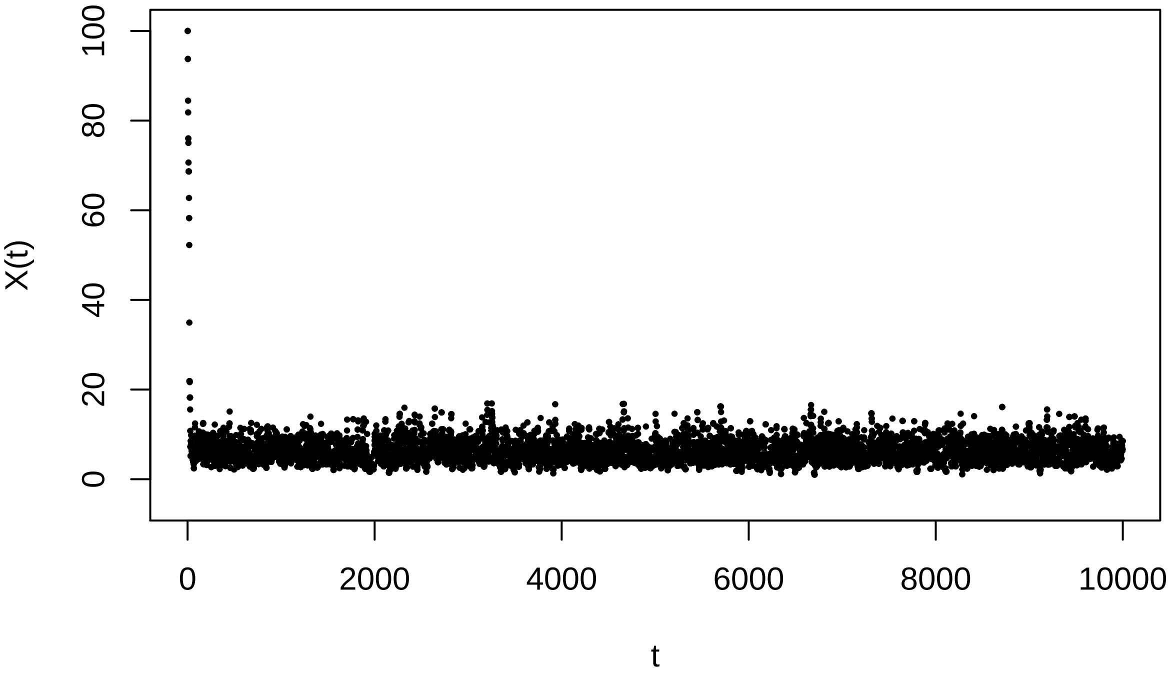


```
f.MCMC.MH2<-function(nstep,X0,props){
    vN<-1:nstep
    vX<-rep(X0,nstep);
    for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rlnorm(1, log(X), sdlog = props)
    u<-runif(1)
    a<-min(c(1,(f_x(Y)*dlnorm(X,meanlog=log(Y),sdlog=props))/(f_x(X)*dlnorm(Y,meanlog=log(X),sdlog=props
    if (u <=a){
      vX[i]<-Y
    }else{
        vX[i]<-X
        }
    }
    return(vX)
}
```

2

```
f.MCMC.chi<-function(nstep,X0,props){
    vN<-1:nstep
    vX<-rep(X0,nstep);
    for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rchisq(1, floor(abs(X)))
    u<-runif(1)
    a<-min(c(1,(f_x(Y)*dchisq(X,floor(abs(Y)))) / (f_x(X)*dchisq(Y, floor(abs(X))))))
    if (u <=a){
      vX[i]<-Y
    }else{
        vX[i]<-X
        }
    }
    plot(vN,vX,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",main="",
         ylim=c(min(X0-0.5,-5),max(5,X0+0.5)))
}
```

```
f.MCMC.chi(10000, 100, 1)
```



3

4

```
f.MCMC.chi2<-function(nstep,X0,props){
    vN<-1:nstep
    vX<-rep(X0,nstep);
    for (i in 2:nstep){
    X<-vX[i-1]
    Y<-rchisq(1, floor(abs(X)))
    u<-runif(1)
    a<-min(c(1,(f_x(Y)*dchisq(X,floor(abs(Y)))) / (f_x(X)*dchisq(Y, floor(abs(X))))))
    if (u <=a){
      vX[i]<-Y
    }else{
        vX[i]<-X
        }
    }
    return(vX)
}
```

```
library(coda)
f1 <- mcmc.list()
library(coda)
for (i in 1:10){
  f1[[i]] <- as.mcmc(f.MCMC.chi2(10000, i, 1))
}
```

```
print(gelman.diag(f1))
```

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

5

```
logsample <- f.MCMC.MH2(1000, 10, 1)
chismaple <- f.MCMC.chi2(1000, 10, 1)
mean(logsample)
```

```
## [1] 6.010904
```

```
mean(chismaple)
```

```
## [1] 6.061986
```

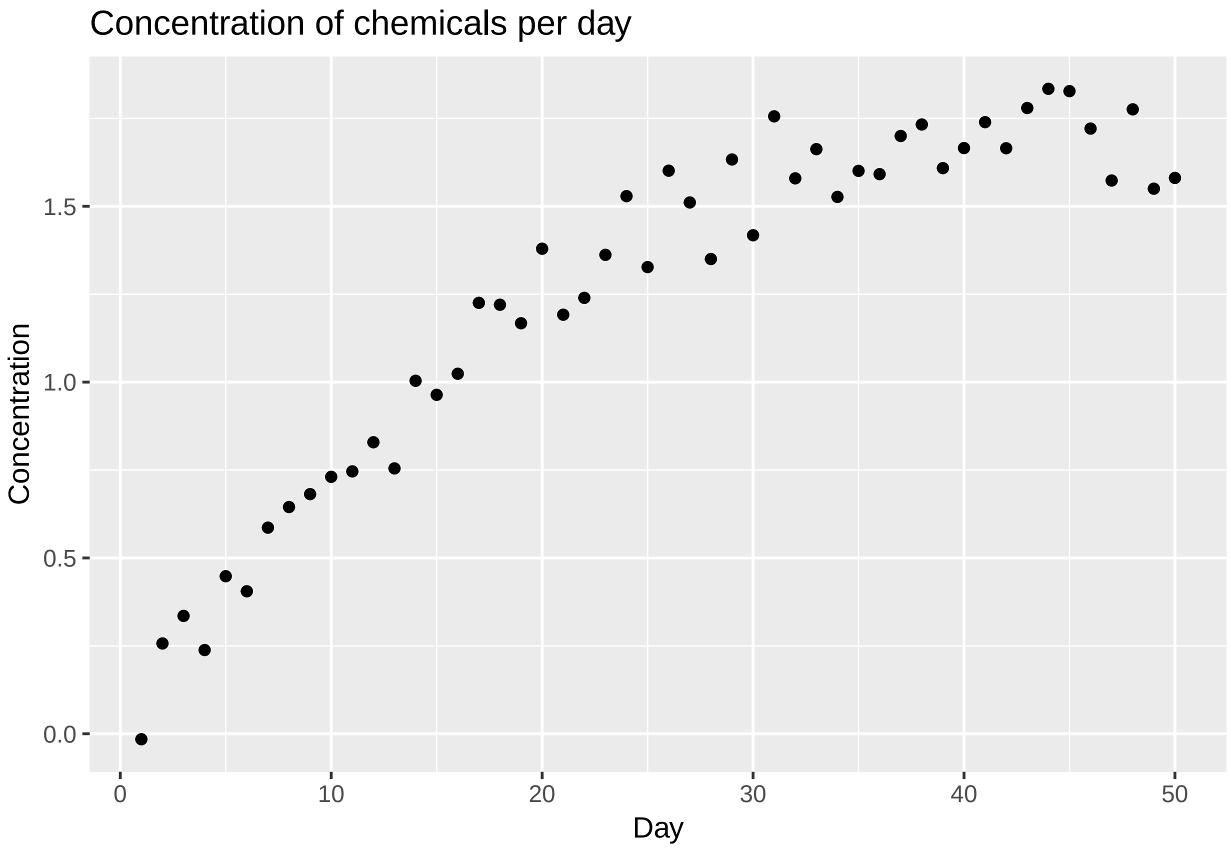
6

Question 2 - Gibbs sampling

1

```
load("chemical.rData")
df <- as.data.frame(cbind(X, Y))
colnames(df) <- c("Day", "Concentration")
```

```
library(ggplot2)
plot <- ggplot(df, aes(x = df$Day, y = df$Concentration)) + geom_point() +
  xlab("Day") + ylab("Concentration") +
  ggtitle("Concentration of chemicals per day")

plot
```



Given the shape observations I would say a logarithmic model is most suitable here.

2

We are given the following distribution:

$$Y \sim N(\mu, variance = 0.2), i = 1, ..., n$$

To compute the likelihood we take the product of the Probability Density Function (PDF) for a normal distribution. The PDF for a normal distribution is written as:

$$PDF = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(y_i - \mu_i)}$$

Now we compute the likelihood by taking the product of the provided PDF. This is computed as following:

$$p(\overrightarrow{Y}|\overrightarrow{\mu}) = \prod \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(y_i - \mu_i)}$$

$$p(\overrightarrow{Y}|\overrightarrow{\mu}) = (\frac{1}{\sqrt{2\pi\sigma^2}})^n e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i)}$$

To compute the prior we use:

$$p(\mu_1) = 1$$

$$p(\mu_{i+1}|\mu_i) = N(\mu_i, varience = 0.2), i = 1, ..., n$$

Now, we use a chain rule:

$$p(\overrightarrow{\mu}) = p(\mu_1)p(\mu_2|\mu_3)p(\mu_3|\mu_2)...p(\mu_n|u_{n-1})$$

$$p(\overrightarrow{\mu}) = 1 * (\frac{1}{\sqrt{2\pi\sigma^2}})^n e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2}$$

$$p(\overrightarrow{\mu}) = (\frac{1}{\sqrt{2\pi\sigma^2}})^n e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2}$$

3

Following Bayes' Theorem:

$$Posterior = prior * likelihood$$

$$Posterior = (\frac{1}{\sqrt{2\pi\sigma^2}})^n e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2} * (\frac{1}{\sqrt{2\pi\sigma^2}})^n e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu_i)^2}$$

$$Posterior = (\frac{1}{\sqrt{2\pi\sigma^2}})^{2n} e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2} e^{\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu_i)^2}$$

$$Posterior \propto c * e^{\frac{-1}{2\sigma^2}(\sum_{i=1}^{n-1}(\mu_{i+1}-\mu_i)^2+\sum_{i=1}^{n}(y_i-\mu_i)^2)}$$

$$(\mu_1|\overrightarrow{\mu}_{-1}, \overrightarrow{Y}) \sim ce^{\frac{-1}{2\sigma^2}((\mu_1-\mu_2)^2+(\mu_1-y_1)^2)}$$

$$(\mu_1|\overrightarrow{\mu}_{-1}, \overrightarrow{Y}) \propto e^{\frac{-(\mu_1-\frac{\mu_2+y_1}{2})^2}{\sigma^2}}$$

$$(\mu_1|\overrightarrow{\mu}_{-1}, \overrightarrow{Y}) \sim N(\frac{\mu_2+y_1}{2}, \frac{\sigma^2}{2})$$

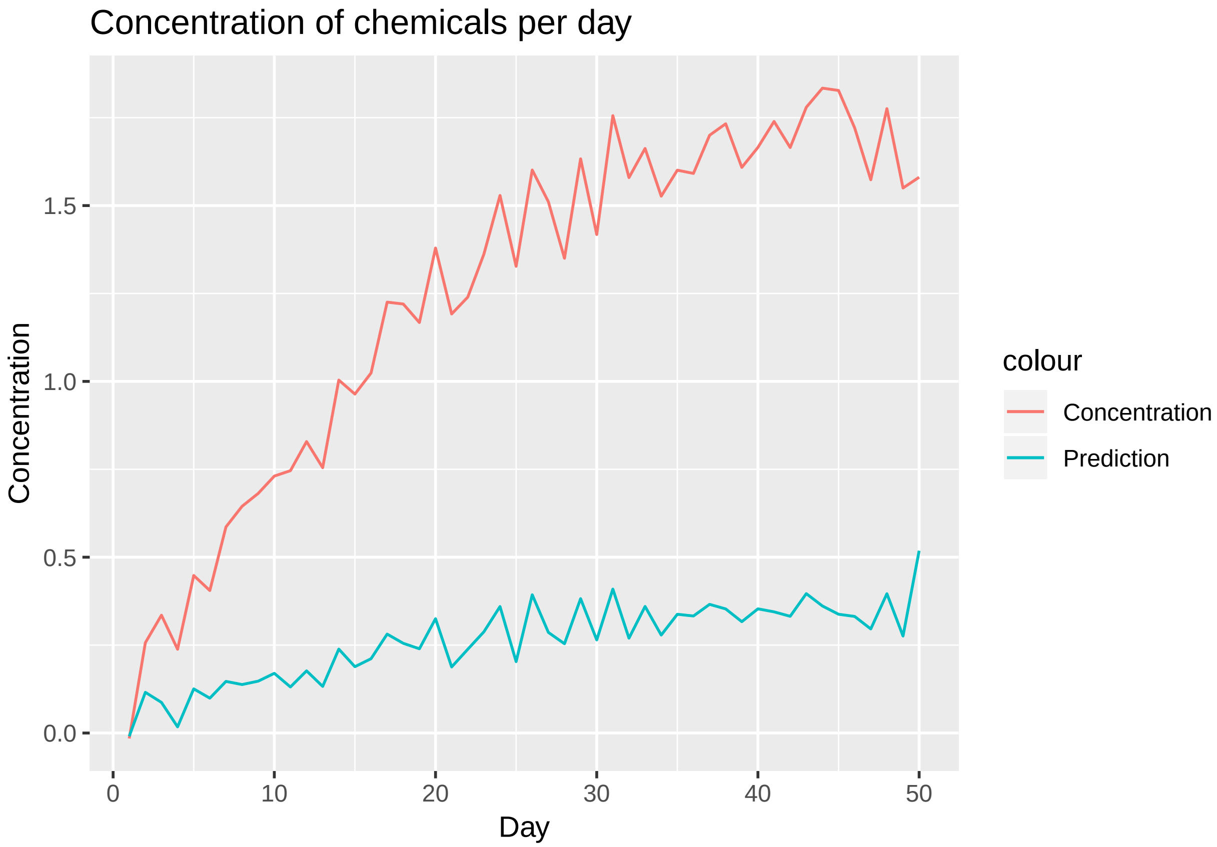
4

```r
mu_s <- matrix(0, nrow = 1000, ncol = 50)
mu_s <- as.data.frame(mu_s)
n_samples <- 1000
```

```r
set.seed(12345)
for (i in 1:n_samples){
  # if we consider the first datapoint, the marginal is different. So:
  mu_s[i,1] <- rnorm(1, mean = (mu_s[,2] + Y[1])/2, sd = sqrt(0.1))

  for (observation in 2:49){
    mu_s[1,] <- 0
    mu_s[i, observation] <- rnorm(1, mean = ((2*Y[observation] + 2*mu_s[, observation-1] + 2*mu_s[,obse
  }

    # if we consider the last datapoint, the marginal is also different. So:
  mu_s[i, 50] <- rnorm(1, mean = ((2*Y[50] + 2*mu_s[,49] - Y[49])/3), sd = sqrt((0.2/3)))

  }
```

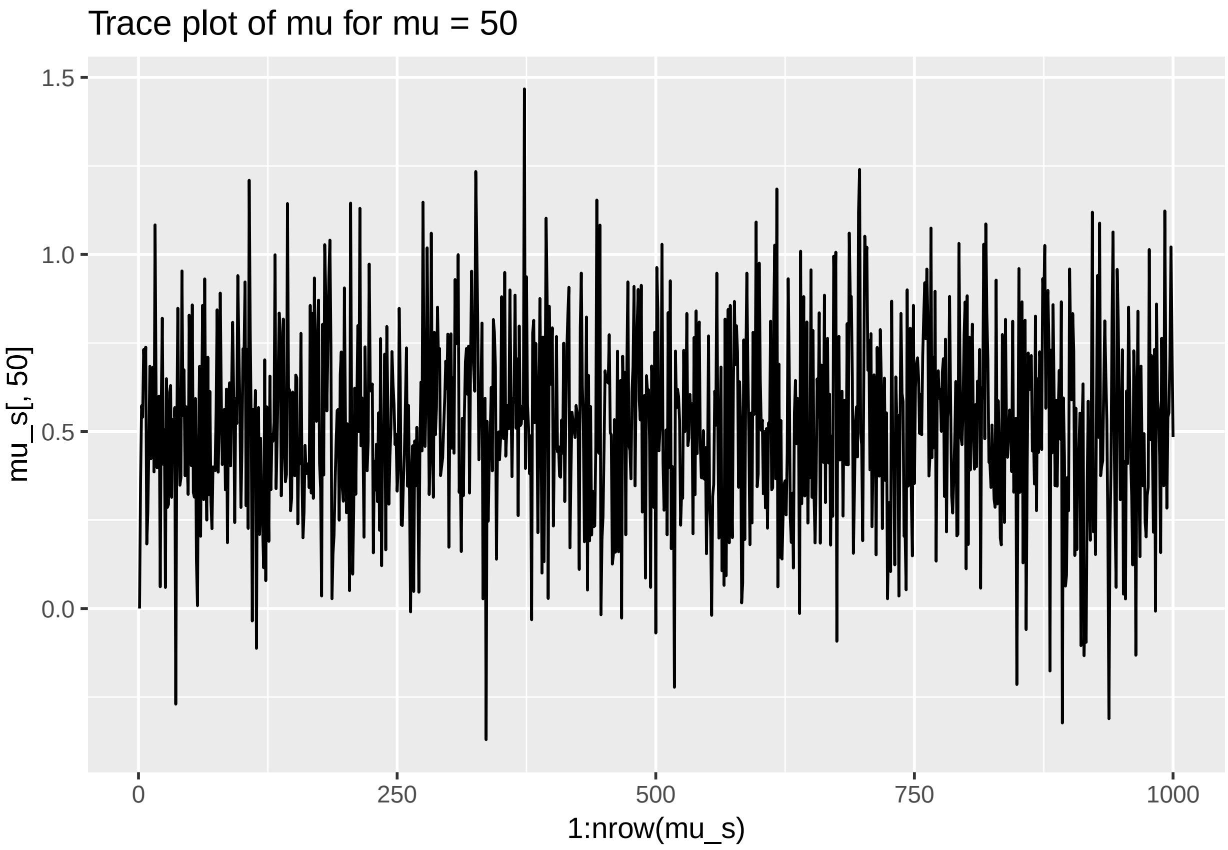
```r
final_predicted_mu <- c(colMeans(mu_s))
df$Prediction <- final_predicted_mu
```

```r
plot2 <- ggplot(df, aes(x = df$Day, y = df$Concentration, col = "Concentration")) + geom_line() +
  xlab("Day") + ylab("Concentration") +
  ggtitle("Concentration of chemicals per day") +
  geom_line(aes(x = df$Day, y = df$Prediction, col = "Prediction"))


plot2
```



5

```r
plot3 <- ggplot(mu_s, aes(x = 1:nrow(mu_s), y = mu_s[,50])) + geom_line() +
  ggtitle("Trace plot of mu for mu = 50")
plot3
```


Trace plot of mu for mu = 50
mu_s[, 50]
1.5
1.0
0.5
0.0
0
250
500
750
1000
1:nrow(mu_s)