

Computer Lab 6

Martynas Lukosevicius, Alejo Perez Gomez, Zahra Jalilpour

2020-12-09

Question 1: Genetic algorithm

Q1. Pass.

1.

Function

$$f(x) = \frac{x^2}{e^x} - 2\exp(-(9\sin x)/(x^2 + x + 1))$$

```
f <- function(x){  
  part1 <- (x^2) / exp(x)  
  expPart <- -(9*sin(x)) / ((x^2) + x + 1)  
  return(part1 -2 * exp(expPart))  
}
```

2.

Crossover function $\frac{x+y}{2}$:

```
crossover <- function(x,y){  
  return((x+y)/2)  
}
```

3.

mutate function $x^2 \bmod 30$:

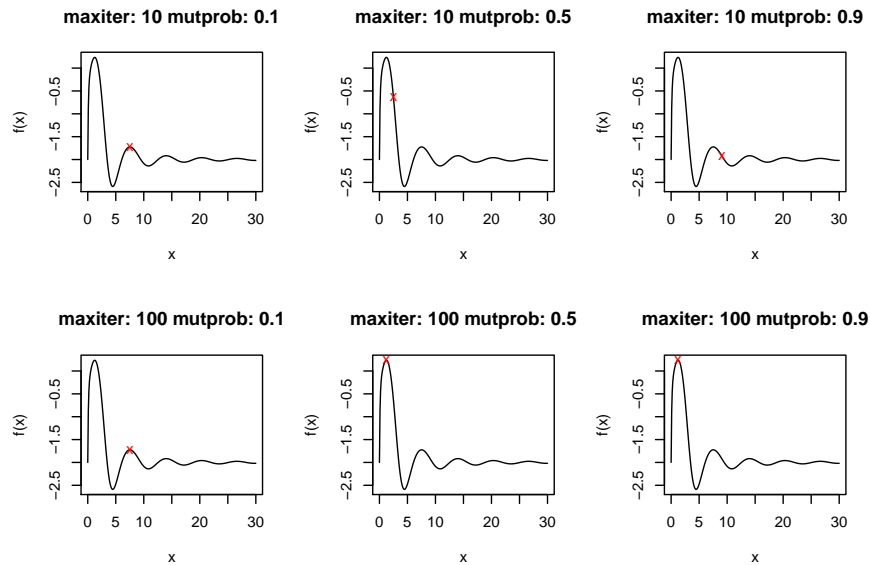
```
mutate <- function(x){  
  return((x^2)%% 30)  
}
```

4.

Function implementing genetic algorithm:

1. initialazise parameters
2. select parents
3. select victim
4. generate kid by crossover and mutation
5. replace victim with kid
6. save best value
7. repeat steps 2 to 3 maxiterations times.

5.



initial	0.000000	5.000000	10.000000	15.000000	20.000000	25.000000	30.000000
10 0.1	0.000000	20.000000	15.000000	15.000000	20.000000	7.500000	18.125000
10 0.5	0.000000	2.500000	13.125000	15.000000	20.000000	13.125000	6.250000
10 0.9	0.000000	6.250000	6.250000	15.000000	20.000000	9.062500	26.595764
100 0.1	13.999200	13.999023	7.500000	13.999112	13.999193	13.999112	13.999196
100 0.5	1.230404	1.232468	1.230404	1.237568	1.230404	1.525270	1.230404
100 0.9	1.327669	1.189410	1.211555	1.275721	1.157227	1.327669	1.546635

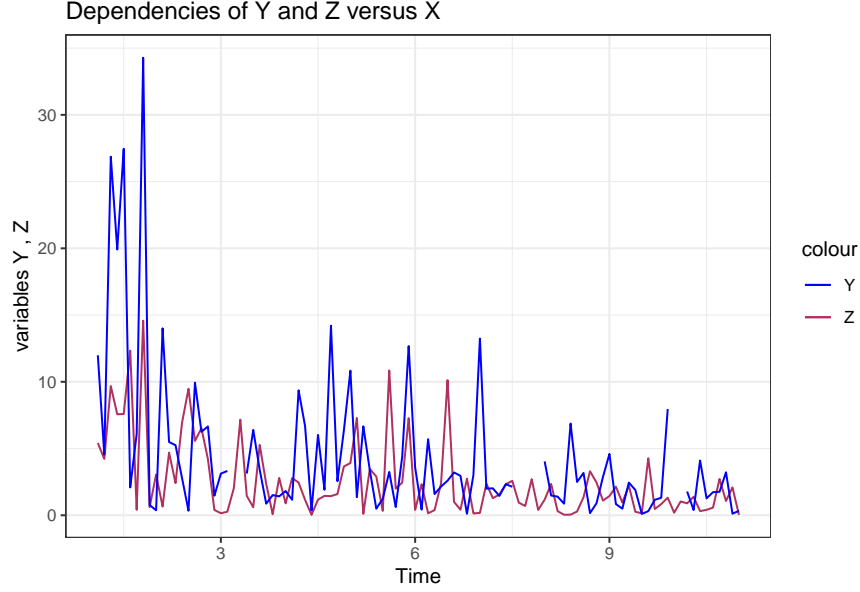
10 iterations are not enough for genetic algorithm to find the max value. It can be seen that increasing mutation probability, max value will fluctuate around the max value and will not reach it.

Question 2: EM algorithm

Q2. Pass.

1.

physical data that we have analyzed, consists of two physical process Y and Z . Here we make time series plot. Both variables start with high value and decrease by time. Variable Z has eight missing value. Both series have similar pattern.



2.

As we see missing values in Z series, we will have problem in estimating model by maximum likelihood. Here we assume that we have these two different models for Y and Z :

$$Y_i = \exp\left(\frac{X_i}{\lambda}\right), Z_i = \exp\left(\frac{X_i}{2\lambda}\right)$$

λ is some unknown parameter. The aim of this task is to derive an EM algorithm to estimate λ . Expected value of the log likelihood function of λ can be defined as:

$$Q(\lambda, \lambda^k) = E[\loglik(\lambda|X, Y, Z)|\lambda^k, X, Y, Z]$$

pdf

Probability density function of an exponential distribution equals:

$$X \sim \text{Exp}(\lambda) \Rightarrow pdf = \lambda e^{-\lambda x}$$

Here we assume Y is observed data and Z consists of latent data

$$Y_i \sim \text{Exp}\left(\frac{X_i}{\lambda}\right) \Rightarrow pdf = \frac{X_i}{\lambda} e^{-\frac{X_i}{\lambda} Y_i}$$

$$Z_i \sim \text{Exp}\left(\frac{X_i}{2\lambda}\right) \Rightarrow pdf = \frac{X_i}{2\lambda} e^{-\frac{X_i}{2\lambda} Z_i}$$

likelihood

$$l(\lambda|Y, Z) = \prod_{i=1}^n p(Y_i, Z_i|\lambda) = \prod_{i=1}^n p(Y_i|\lambda) p(Z_i|\lambda) = \prod_{i=1}^n \left[\frac{X_i}{\lambda} e^{-\frac{X_i}{\lambda} Y_i} \right] \cdot \prod_{i=1}^n \left[\frac{X_i}{2\lambda} e^{-\frac{X_i}{2\lambda} Z_i} \right]$$

Log_likelihood

$$\ln(l(\lambda|Y, Z)) = \ln\left(\prod_{i=1}^n \left[\frac{X_i}{\lambda} e^{-\frac{X_i}{\lambda} Y_i} \right] \cdot \prod_{i=1}^n \left[\frac{X_i}{2\lambda} e^{-\frac{X_i}{2\lambda} Z_i} \right]\right)$$

$$\ln(l(\lambda|Y, Z)) = \ln\left(\sum_{i=1}^n x_i^2\right) - n\ln(2) - 2n\ln(\lambda) - \frac{\sum_{i=1}^n X_i Y_i}{\lambda} - \frac{\sum_{i=1}^{obs} X_i Z_i}{2\lambda} - \frac{\sum_{i=1}^{notobs} X_i Z_i}{2\lambda}$$

In this equation we have not considered missing values in Z . As Z consists of missing values, it is not possible to calculate likelihood of this model. Hence we divide Z into two series, observed and unobserved values, $Z = (Z^o, Z^{notobs})$ and compute Expected log_likelihood. j is the number of missing data.

Max_log_likelihood Estimation:

In this step, to find the maximum likelihood, we should derive the expected log_likelihood according to λ and set the equation to zero to calculate λ . We consider λ_k as unobserved parameter. It is M_step:

$$Q(\lambda, \lambda^k) = \ln\left(\sum_{i=1}^n x_i^2\right) - n\ln(2) - 2n\ln(\lambda) - \frac{\sum_{i=1}^n X_i Y_i}{\lambda} - \frac{\sum_{i=1}^{obs} X_i Z_i}{2\lambda} - \frac{j\lambda_k}{\lambda}$$

Now we should derive this function according to λ and set to zero.

$$\frac{\partial E[L(\lambda)]}{\partial \lambda} = -\frac{4n\lambda - \sum_{i=1}^n X_i Y_i - 2\sum_{i=1}^{obs} X_i Z_i - 2j\lambda_k}{2\lambda^2}$$

$$\lambda = \frac{\sum_{i=1}^{obs} X_i Z_i + 2\sum_{i=1}^n X_i Y_i + 2j\lambda_k}{4n}$$

Good!

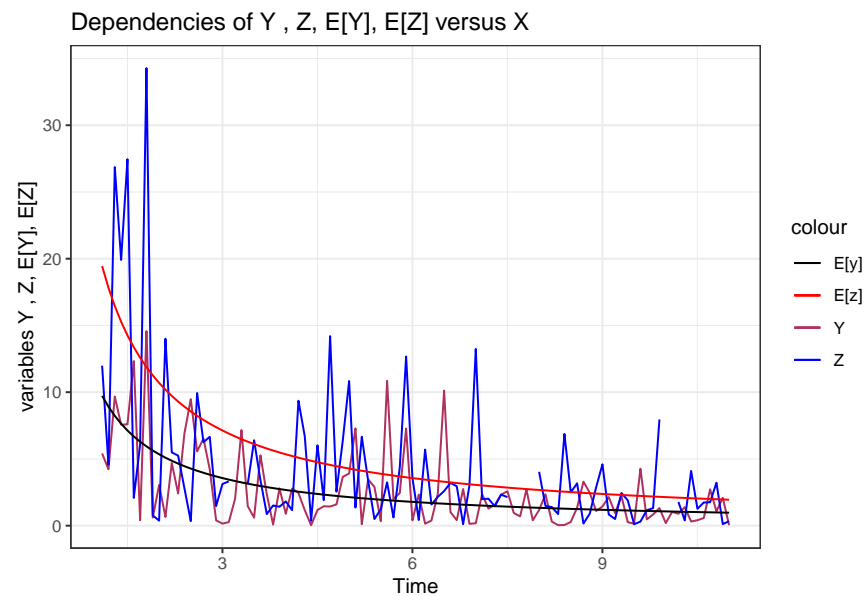
3.

```
em_exp <- function(data, eps, kmax, lambda0){
  # observed Z and X associated to it
  obs <- data[!is.na(data$Z), ]
  miss <- data[is.na(data$Z), ]
  n <- length(data$X)
  r <- length(miss$X)
  # initialize iteration
  llvalcurr <- lambda0
  k <- 0
  llvalprev <- 0
  while((abs(llvalprev-llvalcurr)>eps) && (k<(kmax+1))){
    llvalprev<-llvalcurr
    llvalcurr <- ((t(obs$X) %*% obs$Z) +
                  2*(t(df$X) %*% df$Y) +
                  2 * length(miss$X) * llvalcurr)/(4*length(df$X))

    k <- k+1
  }
  return(list(iterations = k, lambda = llvalcurr))
}
```

iteartions	lambda
5	10.695655498316

4.



By this λ , we calculated $E[Y]$ and $E[Z]$, and both of these have the same trend as Y and Z. They follow the same pattern.

Appendix

```
knitr::opts_chunk$set(echo = F, message = F, error = F, warning = F,
  fig.align='center', out.width="70%")
knitr::opts_chunk$set(cache = TRUE)
f <- function(x){
  part1 <- (x^2) / exp(x)
  expPart <- -(9*sin(x)) / ((x^2) + x + 1)
  return(part1 -2 * exp(expPart))
}
crossover <- function(x,y){
  return((x+y)/2)
}
mutate <- function(x){
  return((x^2)%% 30)
}
genetic <- function(maxiter,mutprob){
  # list for task 5
  res <- list(initial = numeric(), final = numeric())

  # a)
  x <- seq(0,30, 0.1)
  plot(x,f(x), type = "l", main = paste("maxiter:", maxiter,"mutprob:", mutprob))

  # b)
  X <- seq(0,30,5)
  res$initial <- X
  # c)
```

```

Values <- f(X)

# d)

for (i in 1:maxiter) {
  # i.
  parents <- sample(X,2)
  # ii.
  victim <- order(Values)[1]
  # iii.
  kid <- crossover(parents[1], parents[2])
  if(runif(1) < mutprob){
    kid <- mutate(kid)
  }
  # iv.
  X[victim] <- kid
  Values <- f(X)
  # v.
  maxX <- X[which.max(Values)]
  #points(maxX, f(maxX), col="blue", pch="x")
}
res$final <- X
points(maxX, f(maxX), col="red", pch="x")
return(res)
}

par(mfrow=c(2,3))
m10m1 <- genetic(10,0.1)
m10m5 <- genetic(10,0.5)
m10m9 <- genetic(10,0.9)

m100m1 <- genetic(100,0.1)
m100m5 <- genetic(100,0.5)
m100m9 <- genetic(100,0.9)

res <- rbind(m10m1$initial,
             m10m1$final,
             m10m5$final,
             m10m9$final,
             m100m1$final,
             m100m5$final,
             m100m9$final
             )

rownames(res) <- c("initial",
                  "10 0.1",
                  "10 0.5",
                  "10 0.9",
                  "100 0.1",
                  "100 0.5",
                  "100 0.9"
                  )

knitr::kable(res)

```

```

df <- read.csv("physical1.csv")
library(ggplot2)
palette_OkabeIto_black <- c("blue1", "maroon")
p<- ggplot(data=df)
p+geom_line(aes(X, Y, color="Z"))+
  geom_line(aes(X, Z, color="Y"))+
  scale_color_manual(values = palette_OkabeIto_black)+
  theme_bw()+
  labs(x="Time", y="variables Y , Z")+
  ggtitle("Dependencies of Y and Z versus X")

em_exp <- function(data, eps, kmax,lambda0){
  # observed Z and X associated to it
  obs <- data[!is.na(data$Z), ]
  miss <- data[is.na(data$Z), ]
  n <- length(data$X)
  r <- length(miss$X)
  # initialize iteration
  llvalcurr <- lambda0
  k <- 0
  llvalprev <- 0
  while((abs(llvalprev-llvalcurr)>eps) && (k<(kmax+1))){
    llvalprev<-llvalcurr
    llvalcurr <- ((t(obs$X) %*% obs$Z) +
                  2*(t(df$X) %*% df$Y) +
                  2 * length(miss$X) * llvalcurr)/(4*length(df$X))

    k <- k+1
  }
  return(list(iterations = k, lambda = llvalcurr))
}

knitr::kable(t(em_exp(df,0.001, 50, 100)), col.names = c("iterations", "lambda"))

expectedvalz = (2 * 10.69566) / df$X
expectedvaly = 10.69566 / df$X
df <- cbind(df,expectedvalz, expectedvaly, deparse.level = 1)
palette_OkabeIto_black <- c("Z" = "blue1", "Y" = "maroon","E[z]" = "red","E[y]" = "black")
p<- ggplot(data=df)
p+geom_line(aes(X, Y , col = "Y"))+
  geom_line(aes(X, Z, col = "Z"))+
  geom_line(aes(X, expectedvalz , col = "E[z]"))+
  geom_line(aes(X, expectedvaly, col = "E[y]"))+
  scale_color_manual(values = palette_OkabeIto_black)+
  theme_bw()+
  labs(x="Time", y="variables Y , Z, E[Y], E[Z]" ) +
  ggtitle("Dependencies of Y , Z, E[Y], E[Z] versus X")

```