732A90: Computational Statistics

Computer lab6 - Group11

Sofie Jörgensen, Oriol Garrobé Guilera, David Hrabovszki

06 March 2020

Question 1: Genetic algorithm

In this exercise we are going to perform one-dimensional maximization by using a genetic algorithm.

1.

Firstly, we define the function `f()` as

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

2.

Secondly, we define the function `crossover()`, that takes two scalars x and y as inputs, and returns a child as $\frac{x+y}{2}$.

3.

Thirdly, we define the function `mutate()`, that performs the integer division x^2 mod 30, for a scalar input x.

4.

Further, we will create a function called `genetic()`, with the parameters `maxiter` and `mutprob`. The settings of this `genetic()` function, as well as its output results, are presented in (a)-(e). The code can be found in the Appendix.

(a). The function $f()$ is plotted in the range from 0 to 30 in Figure X, and we can observe that there is a maximum value located around $x = 1$.

(b). An initial population for the genetic algorithm is defined as $X = (0, 5, 10, 15, ..., 30)$.

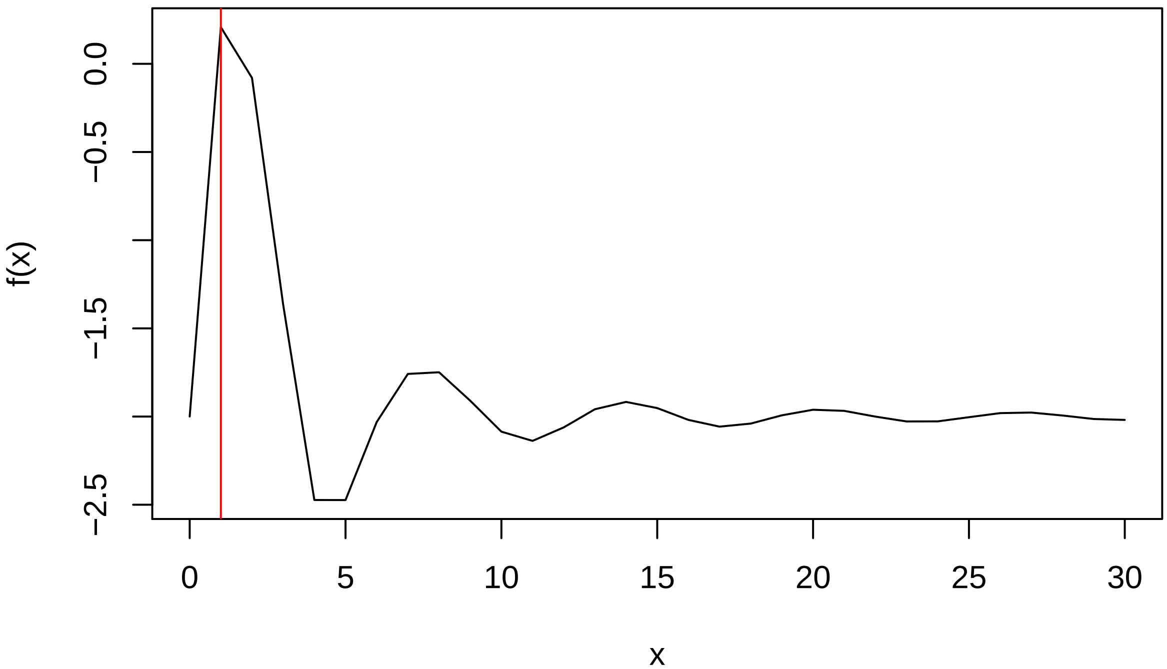
(c). A vector called `Values` are computed, containing the function values for each population point.

(d). The `genetic()` function performs `maxiter` iterations. For each iteration…

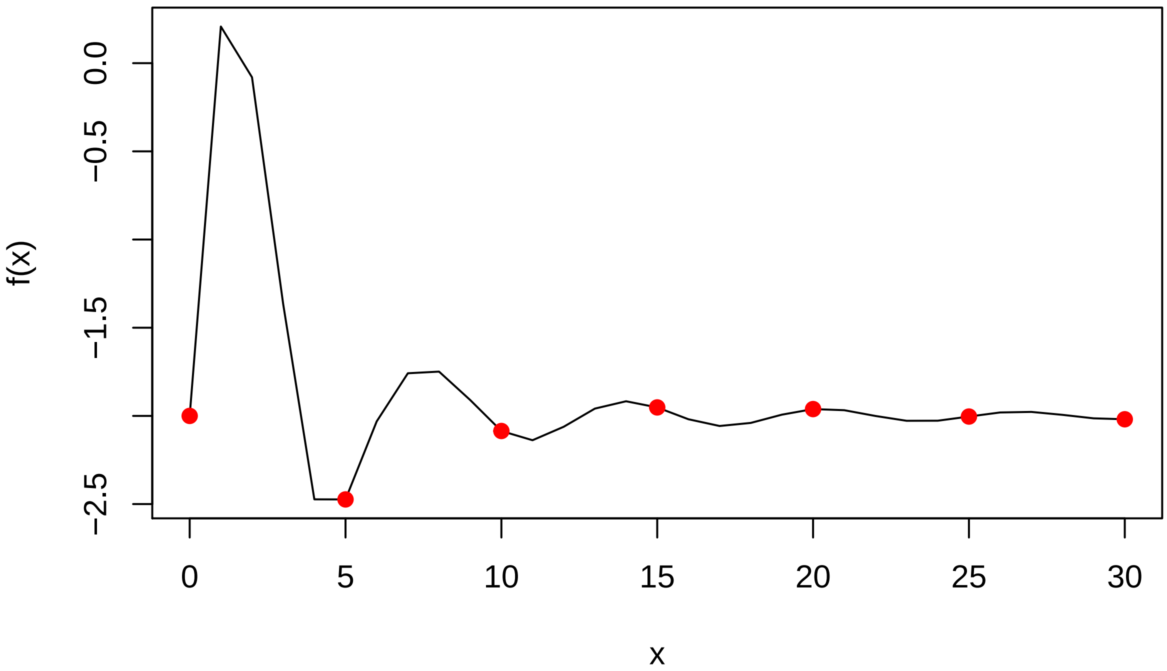
(e).

5.

By using the defined functions from previous tasks (1.1-1.4), we are going to observe the initial population and final population. This is done by running the code with different combinations of `maxiter`$= 10, 100$ and `mutprob`$= 0.1, 0.5, 0.9$.




```
## [1]  0  5 10 15 20 25 30
## [1] -2.000000 -2.473573 -2.085654 -1.951947 -1.961344 -2.003663 -2.019194
```



Question 2: EM algorithm

The purpose with this exercise is to implement the EM algorithm. For this, we are given the data file `physical1.csv`, containing a behavior of two related physical processes $Y = Y(X)$ and $Z = Z(X)$.

1.

The first step is to examine the data set `physical1.csv`, to see if the two processes are related to each other.

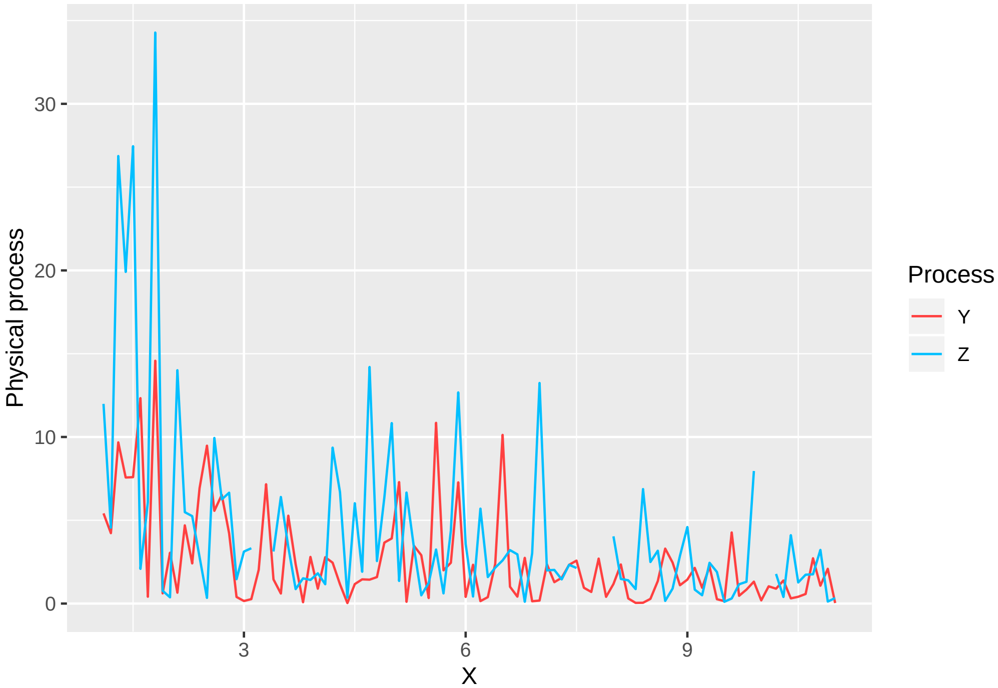


Figure 1: Time series plot of the dependence of Z and Y versus X.

In Figure 1 it seems that the two processes are related to each other, with respect to X, since the graphs follows similar patterns. We can also observe that the physical process Z has a greater variation, especially at the beginning of the series, but also in general, compared to the process Y.

2.

Using the following model,

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

where λ is an unknown parameter, we derive the EM algorithm to estimate λ.

Y and Z are defined by the following density functions,

$$\begin{aligned} Z(X) &= \frac{X_i}{2\lambda} exp\left(-\frac{X_i}{2\lambda} Z_i\right) \\ Y(X) &= \frac{X_i}{\lambda} exp\left(-\frac{X_i}{\lambda} Y_i\right) \end{aligned}$$

Assuming that $Y(X)$ and $Z(X)$ are i.i.d. we can obtain the joint density function,

$$\begin{aligned} f(Y,Z) &= \frac{X_i}{2\lambda} exp\left(-\frac{X_i}{2\lambda}Z_i\right) * \frac{X_i}{\lambda} exp\left(-\frac{X_i}{\lambda}Y_i\right) \\ &= \frac{X_i^2}{2\lambda^2} exp\left[-\frac{X_i(Z_i - 2Y_i)}{2\lambda}\right] \end{aligned}$$

From this point, we compute the likelihood of the distribution,

$$\mathcal{L}(Y,Z|X,\lambda) = \prod_{i=1}^{n} \frac{X_i^2}{2\lambda^2} exp\left[-\frac{X_i(Z_i - 2Y_i)}{2\lambda}\right]$$

The code can be found in the Appendix.

3.

```
## number_of_iterations        optimal_lambda
##              5.00000              10.69566
```

4.

As a final task, we are going to see if the optimal value $\lambda = 10.69566$ is reasonable for the physical processes $Y = Y(X)$ and $Z = Z(X)$. We examine the results by computing the expected values of Y and Z, which is given by $E[Y_i] = \frac{\lambda}{X_i}$ and $E[Z_i] = \frac{2\lambda}{X_i}$, respectively.

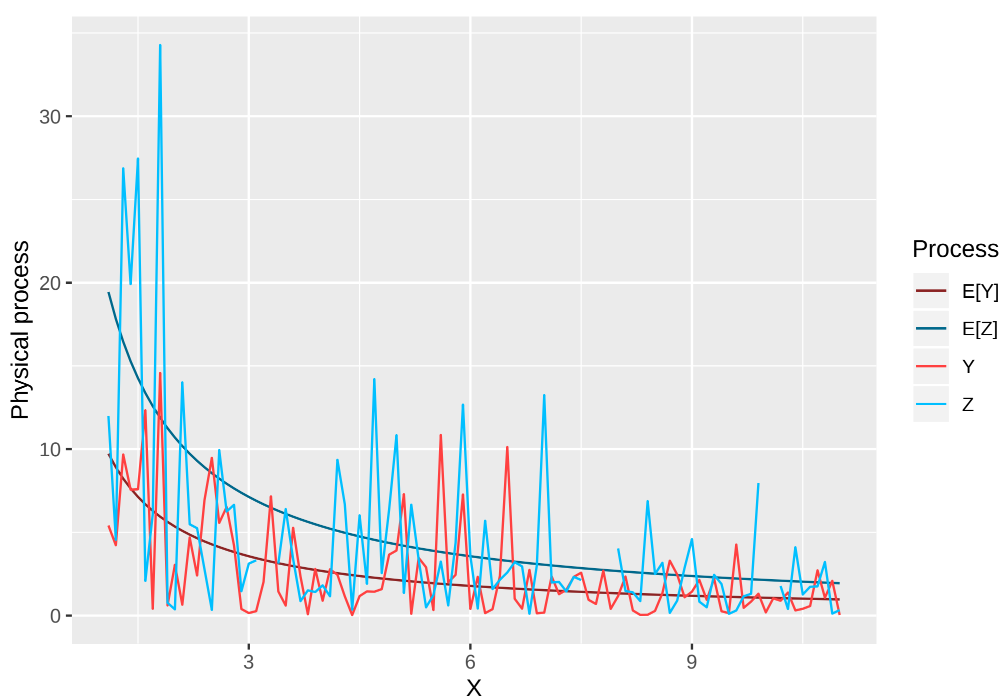


Figure 2: Time series plot of E[Y] and E[Z] versus X, and the dependence of Z and Y versus X.

In Figure 2, we can see that $E[Y]$ and $E[Z]$ versus X are in the same plot as Y and Z versus X. From this plot it is clear that the optimal λ is reasonable, since the exponential curves $E[Y]$ and $E[Z]$, the follow their corresponding values of Y and Z.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
# R version
RNGversion('3.5.1')
library("ggplot2")
#1.1
f <- function(x){
  return(x^2/exp(x) - 2*exp(-1*(9*sin(x)) / (x^2 + x + 1)))
}

#1.2
crossover <- function(x,y){
  return((x+y) / 2)
}

#1.3
mutate <- function(x){
  return(x^2 %% 30)
}
#4
#4
genetic <- function(maxiter, mutprob){
  #a
  plot(x = seq(0,30), y = f(seq(0,30)), type = "l", xlab = "x", ylab = "f(x)")
  abline(v=seq(0,30)[which.max(f(seq(0,30)))], col="red" )

  #b
  X = seq(0,30,5)

  #c
  Values = f(X)

  #d
  #set seed
  set.seed(1234567890)
  for (i in 1:maxiter) {
    #i
    parents = match(sample(X, 2),X)

    #ii
    victim = order(Values)[1]

    #iii
    kid = round(crossover(parents[1],parents[2]))
    p = runif(1)
```

```
    if (p < mutprob) {
      kid = mutate(kid)
    }

    #iv
    X[victim] = kid
    Values = f(X)

    #v
    max = max(Values)
  }

  #e
  print(X)
  print(Values)
  plot(x = seq(0,30), y = f(seq(0,30)), type = "l", xlab = "x", ylab = "f(x)")
  points(x = X, y = Values, col = "red", pch = 19)
}


# Just testing no change, i.e. initial population
genetic(1,0)
# 2.1
physical <- read.csv2("physical1.csv", sep = ",")

X <- as.numeric(as.character(physical$X))
Y <- as.numeric(as.character(physical$Y))
Z <- as.numeric(as.character(physical$Z))

data <- data.frame(X = c(X,X), value = c(Y,Z), Process= rep(c("Y","Z"), each= 100))

# var(Z,na.rm = TRUE)
# var(Y,na.rm = TRUE)

# Time series plot
ggplot(data = data, aes(x = X, y = value, col = Process)) +
  geom_line() +
  ylab("Physical process")+
  scale_color_manual(values=c( "brown1","deepskyblue1"))


## 2.2

EM_algorithm <- function(X, Y, Z, lambda, eps, k_max) {
    Z_obs <- Z[!is.na(Z)]
    Z_miss <- Z[is.na(Z)]

    X_obs <- X[!is.na(Z)]
    X_miss <- X[is.na(Z)]

    Y_obs <- Y[!is.na(Z)]
    Y_miss <- Y[is.na(Z)]
```

```
    n <- length(c(Z_obs, Z_miss))
    r <- length(Z_obs)

    k<-1
    lambda_prev <- lambda+10+100*eps #random number to initialize the algorithm
    lambda_curr <- lambda

    while (k<k_max+1 && abs(lambda_prev-lambda_curr)>=eps) {

        lambda_prev<-lambda_curr

        ## E-step
        EY <- -n*log(2*lambda_prev^2) + sum(log(X^2)) - sum(X_obs*(Z_obs+2*Y_obs))/(2*lambda_prev) - sur

        ## M-step
        lambda_curr <- (sum(X_obs*(Z_obs+2*Y_obs)) + 2*sum(lambda_curr+X_miss*Y_miss))/(4*n)

        k<-k+1

    }
    return(c(number_of_iterations = k-1, optimal_lambda = lambda_curr))
}

##2.3

EM_algorithm(X,Y,Z, 100, 0.001, 100)
# 2.4
optimal_lambda <- EM_algorithm(X,Y,Z, 100, 0.001, 100)[2]
# Expected values
EY <- optimal_lambda/X
EZ <- 2*optimal_lambda/X

data2 <- data.frame(X = c(X,X), value = c(Y,Z,EY,EZ), Process= rep(c("Y","Z","E[Y]","E[Z]"), each= 100))

#
ggplot(data = data2, aes(x = X, y = value, col = Process)) +
  geom_line() +
  ylab("Physical process") +
  scale_color_manual(values=c("brown4", "deepskyblue4", "brown1", "deepskyblue1"))
```