Computational Statistics (732A90) Lab 6

Omkar Bhutra (omkbh878), Andreas C Charitos (andch552)

8 March 2019

Question 1 -Genetic Algorithm

Subquestion 1-Define target function

We are going to perform one-dimensinal maximization with the help of genetic algorithm for the following function :

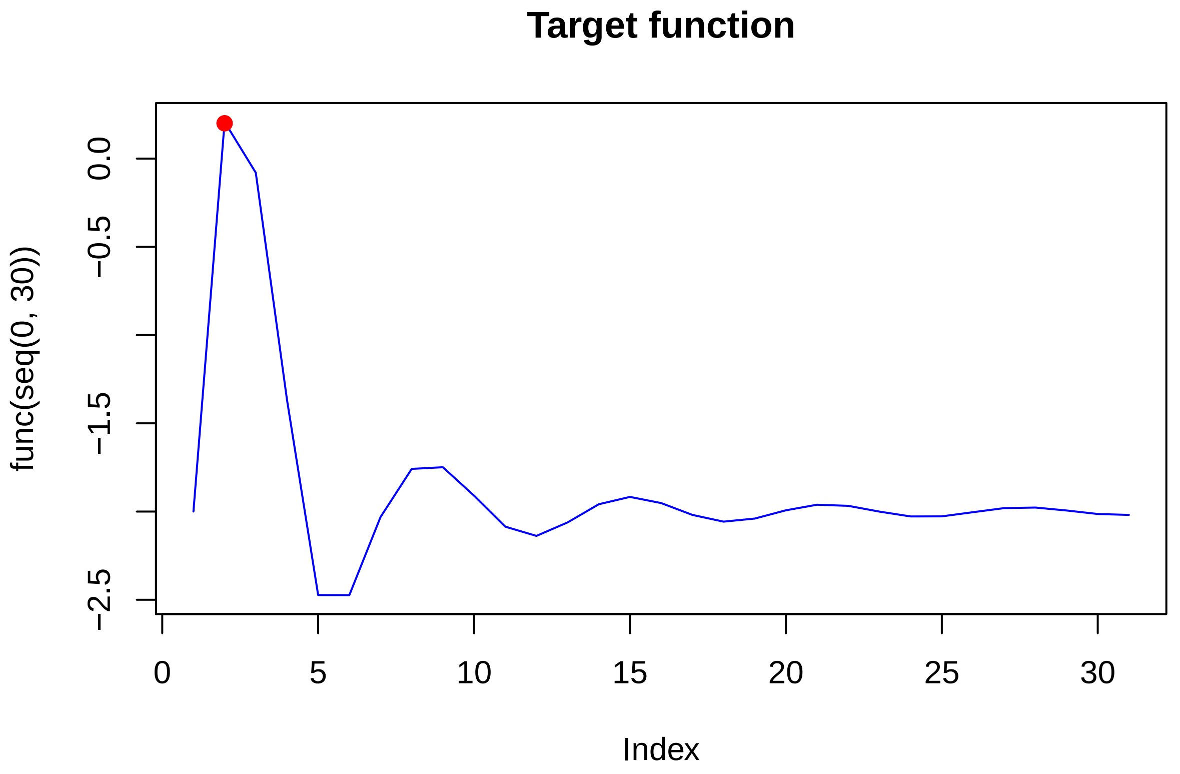
$$f(x) = \frac{x^2}{e^x} - 2e^{(-\frac{(9sin(x))}{(x^2+x+1)})}$$

Subquestion 2-Define crossover function

We are asked to perform the crossover function that for 2 scalars we get their "kid"

Subquestion 3-Define mutation function

The function mutate() takes a scalar x returns the result of the integer division $x^2 mod\ \ 30$ we will use this function to perform a mutation to the "kid" that will be produced in the crossover.



In the plot we can see that the function has some local maximum but with red circle is the global maximum we wish to approximate using genetic algorithm.

Subquestion 4-Define genetic function

we are going to implement the genetic function that has 2 arguments number of iterations and mutation probability

Subquestion 5-Run the function with diffrent settings

Last we are goning to try the algorithm for different combinations of maxiter= 10, 100 and mutprob= 0.1, 0.5, 0.9

```
## NULL
```

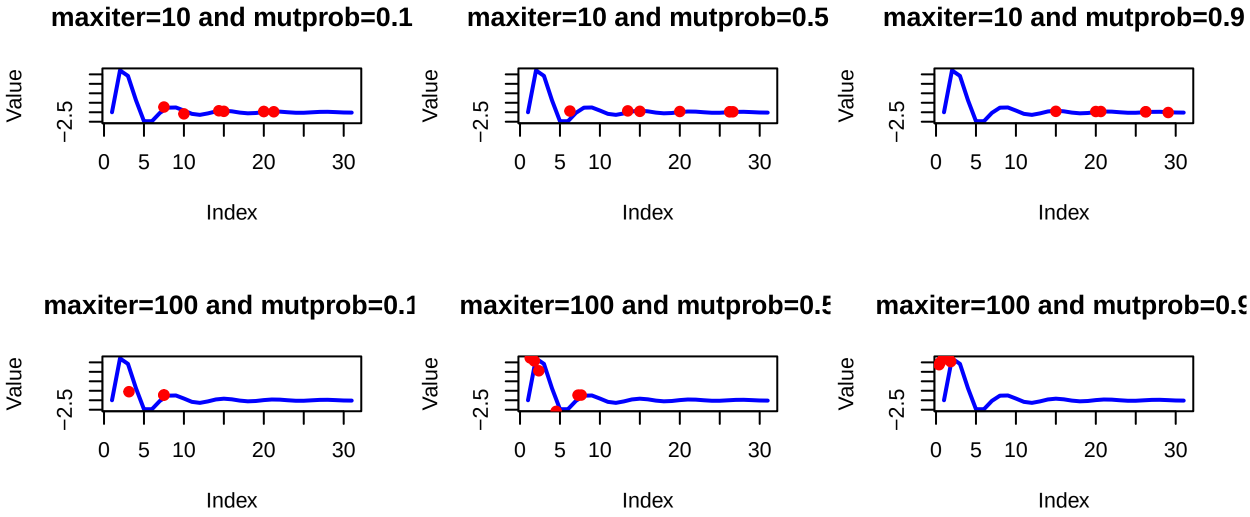
```
## NULL
```

```
## NULL
```

```
## NULL
```

```
## NULL
```

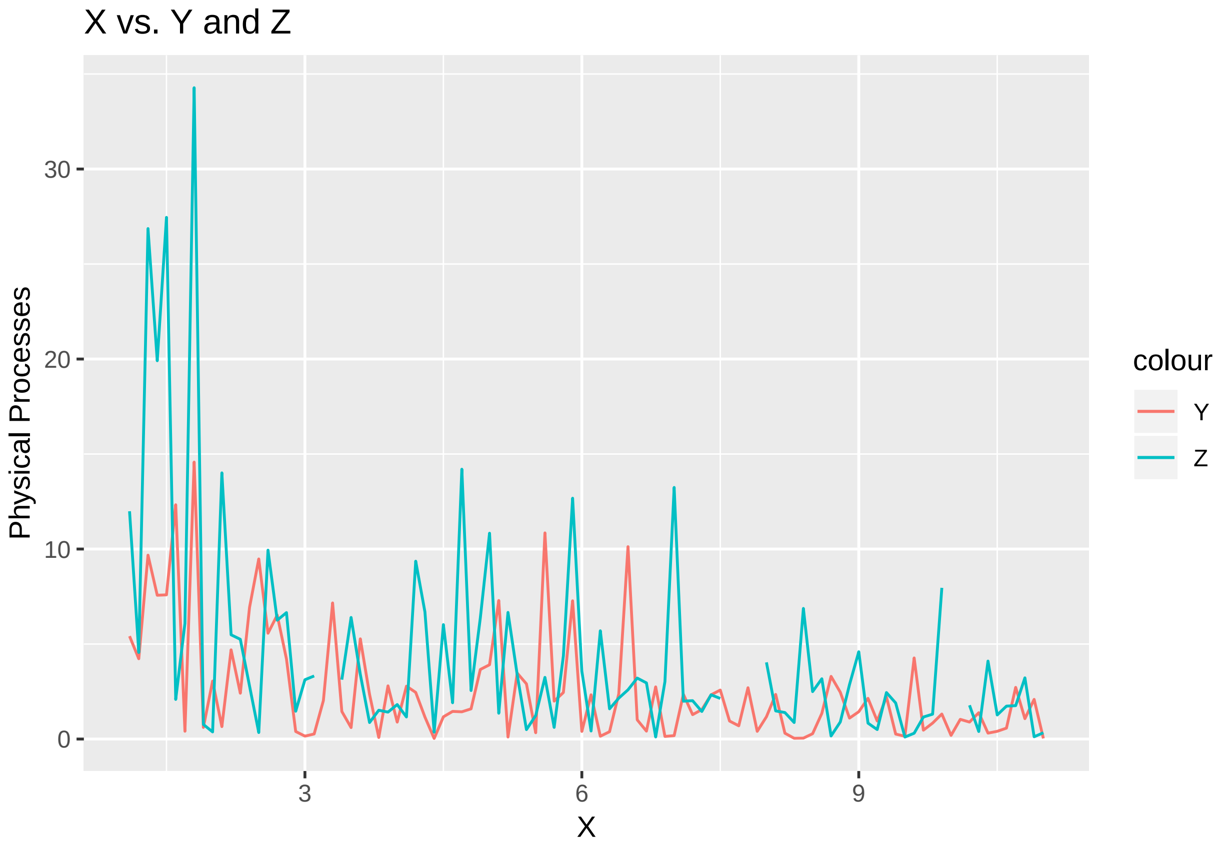
```
## NULL
```



Finaly, the plot shows that with maxiter=100 have better performance than the one with maxiter=10 and the final population values are more closer to maximum. This is reasonable because the algorithm needs more iterations in order to make more crossovers and mutations.Furthermore,if the mutation probability is high the mutatation is very often for the "kids"-points and that makes them change points and explore new points which is the purpose of the mutation.

Question 2: EM algorithm

The data file physical.csv describes a behavior of two related physical processes Y = Y (X) and Z = Z(X). 1. Make a time series plot describing dependence of Z and Y versus X. Does it seem that two processes are related to each other? What can you say about the variation of the response values with respect to X?



Yes, it seems that the two responses are related to each other. It is observed that both responses Y and Z have decayed oscillations, this makes it seem that the responses are related to each other. The decay in response Z is lower than in Y.

2. Note that there are some missing values of Z in the data which implies problems in estimating models by maximum likelihood. Use the following model

$$Y_i \sim exp(X_i = \lambda); Z_i \sim exp(X_i = (2\lambda))$$

where λ is some unknown parameter. The goal is to derive an EM algorithm that estimates λ

$$X_i \sim exp(\lambda)$$

$$L(\lambda|Y,Z) = \prod_{i=1}^{n} f(Y).\prod_{i=1}^{n} f(Z)$$

$$= \prod_{i=1}^{n} \frac{X_i}{\lambda}.e^{\frac{-X_i \cdot Y_i}{\lambda}}.\prod_{i=1}^{n} \frac{X_i}{2\lambda}.e^{\frac{-X_i \cdot Y_i}{2\lambda}}$$

$$= \frac{1}{2\lambda}\prod_{i=1}^{n} X_i^2.exp(\frac{-X_i}{\lambda}.(Y_i + \frac{Z_i}{2}))$$

log-liklihood:

$$= \log(\frac{1}{2\lambda^{2n}}\prod_{i=1}^{n} X_i^2.exp(\frac{-X_i}{\lambda}.(Y_i + \frac{Z_i}{2})))$$

$$= 2\log\prod_{i=1}^{n} X_i - 2n\log(2\lambda) - \frac{1}{\lambda}\sum_{i=1}^{n} X_i(Y_i + \frac{Z_i}{2})$$

E-step:

$$E(X|\lambda) = \frac{1}{\lambda}$$

$$2\log\prod_{i=1}^{n} X_i - 2n\log(2n) - \frac{1}{\lambda}(\sum_{i=1}^{n} X_iY_i + \sum_{i=1}^{r}\frac{X_iV_i}{2} + \sum_{i=r+1}^{n}\frac{X_i}{2}\frac{2\lambda_k}{X_i})$$

M-step: The maximum likelihood estimate of the parameters is obtained by taking the partial derivative with respect to λ. This is repeated till the solution converges.

$$\frac{\partial L(\lambda|Y,Z)}{\partial\lambda} = 0$$

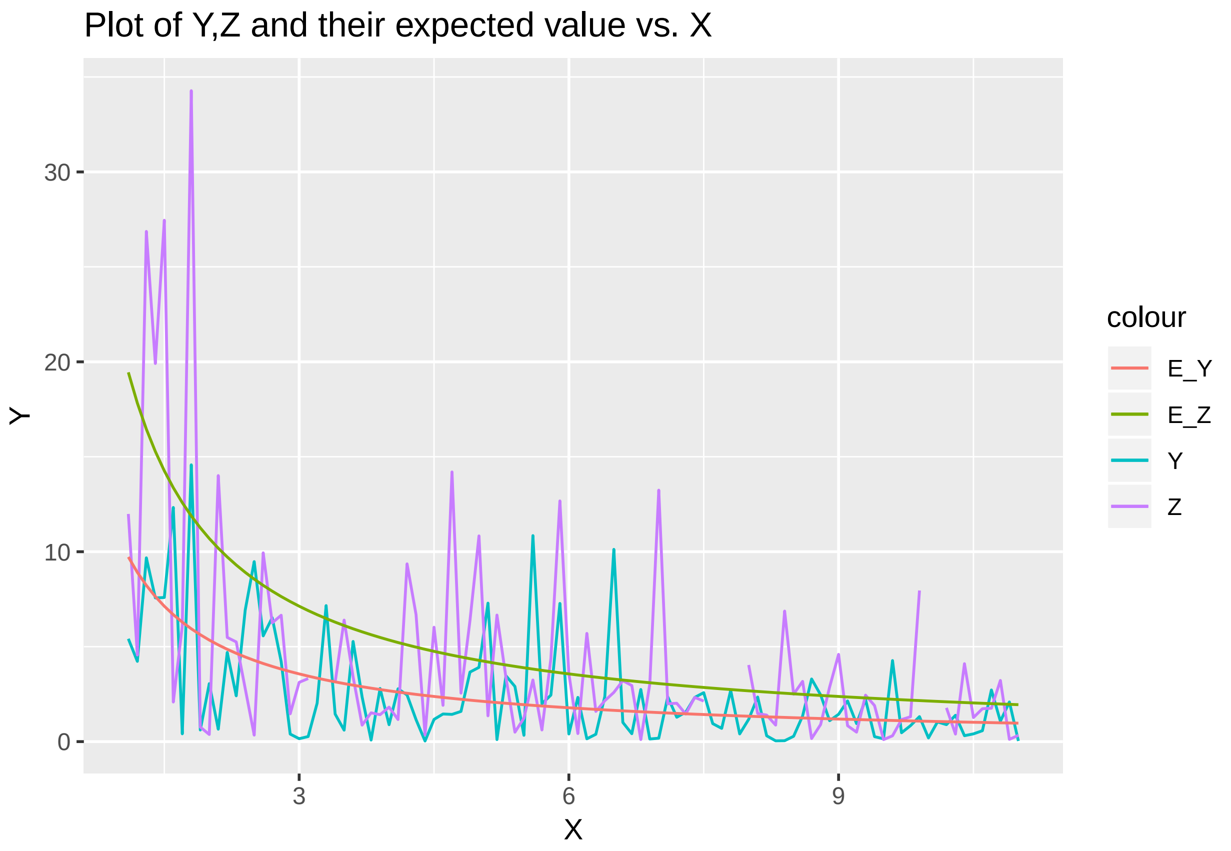
$$\lambda = \frac{1}{2n}(\sum_{i=1}^{n} X_iY_i + \sum_{i=1}^{r}\frac{X_iV_i}{2} + (n-r)\lambda_k)$$

3. Implement this algorithm in R, use $\lambda_0 = 100$ and convergence criterion "stop if the change in λ is less than 0:001". What is the optimal λ and how many iterations were required to compute it?

```
## [1]   0 100   0
## [1] 10.69587 10.69566  5.00000
```

This result indicates that the lambda value is 10.69566 after 5 iterations that are needed for convergence.

4. Plot E [Y] and E[Z] versus X in the same plot as Y and Z versus X. Comment whether the computed λ seems to be reasonable.



From the graph , it is observed that both exponential's of Y and Z capture the flow of Y and Z against X. Hence, the lambda found from the above algorithm is optimal.

Apendix

```
knitr::opts_chunk$set(echo = TRUE)
library("boot")
library("ggplot2")
library("animation")

func<- function(x){
  result=x^2/exp(x)-2*exp(-(9*sin(x))/(x^2+x+1))
  return(result)
}

crossover<-function(x,y){
  if(is.atomic(x)==TRUE && is.atomic(y)==TRUE){
    kid=(x+y)/2
  }
  else("Inputs have to be scalar")
  return(kid)
}

mutate<-function(x){
  if(is.atomic(x)==TRUE){
    result=(x^2%%30)
  }
  return(result)
}


 targret_func<-function(x){

  result1<-x^2/exp(x)
  result2<-2*exp( -(9*sin(x)) / (x^2+x+1) )

  return(result1-result2)

}


plot(func(seq(0,30)),type = "l",col="blue",
     main="Target function")
points(2,0.2,col="red",pch=19)



genX<-function(maxiter,mutprob,animated=F){
  #plot funciton that plots the original
  #and the final population obtained
  plotf<-function(){
    plot(targret_func(seq(0,30)),type = "l",col="blue",lwd=2,
```

```r
       main= paste0("maxiter=",maxiter," and mutprob=",mutprob),
       ylab="Value")
  points(initial_pop,targret_func(initial_pop),col="red",
         pch=19)
  return()

}


#initialize a population
initial_pop=seq(0,30,5)
#compute the value from target function
#for each population
Values<-sapply(initial_pop,targret_func)
#initialize vector to store
#max value for every iteration
max_Value<-rep(0,maxiter)
#if statement if we want gif animation
if (animated==F){
  for (i in 1:maxiter){
    #find 2 parents from the initial population
    parents<-initial_pop[c(sample(1:length(initial_pop),2,replace = F))]
    #get the victim that is going to be replaced
    victim<-order(Values)[1]
    #take "kid" from 2 parents with crossover
    new_kid<-crossover(parents[1],parents[2])
    #check the probability for mutation
    #if is higher than mutprob "kid"
    #will be mutated
    if (runif(1)<mutprob){
      new_kid<-mutate(new_kid)
    }
    #change the kid with the victim
    #in the initial population
    initial_pop[victim]<-new_kid
    #calculate the values of the new
    #population again
    Values<-sapply(initial_pop,targret_func)
    #get the max value
    max_Value[i]<-max(Values)



  }
  #plotf()
  return(list("plot"=plotf(),"final_pop"=max_Value))

}
#use this is you want animation gif
else{
  require(animation)
  saveGIF({
ani.options(interval=.3)
```

```
  col.range <- heat.colors(15)

  for (i in 1:maxiter){

    parents<-initial_pop[c(sample(1:length(initial_pop),2,replace = F))]
    victim<-order(Values)[1]
    new_kid<-crossover(parents[1],parents[2])

    if (runif(1)<mutprob){
      new_kid<-mutate(new_kid)
    }

    initial_pop[victim]<-new_kid
    Values<-sapply(initial_pop,targret_func)
    max_Value[i]<-max(Values)

    plot(targret_func(seq(0,30)),type = "l",col="blue",main=paste0("plot for",i,"th iteration"))
    points(initial_pop,targret_func(initial_pop),col="red")

    }#end of for loop

    })#end of animation
  }


}

set.seed(1234567)

par(mfrow=c(3,3))
#maxiter=10
genX(10,0.1,animated=F)$plotf
genX(10,0.5,animated=F)$plotf
genX(10,0.9,animated=F)$plotf
#maxiter=100
genX(100,0.1,animated=F)$plotf
genX(100,0.5,animated=F)$plotf
genX(100,0.9,animated=F)$plotf
physical <- read.csv("physical1.csv", sep=",")
ggplot(physical) + geom_line(aes(x=X, y = Y,color="Y")) +geom_line(aes(x=X, y = Z,color="Z"))+
ggtitle("X vs. Y and Z")+labs(x="X",y="Physical Processes")
EM<-function(physical,lambda0,eps,kmax){
    X<-physical$X
    Y<-physical$Y
    Z<-physical$Z
    Xobs <- X[!is.na(Z)]
    Zobs <- Z[!is.na(Z)]
    Zmiss <- Z[is.na(Z)]

    n <- length(X)
    r <- length(Xobs)

    k<<-0
    llvalprev<-0;
    llvalcurr<-lambda0
    print(c(llvalprev,llvalcurr,k))

    while ((abs(llvalprev-llvalcurr)>eps) && (k<(kmax+1))){
    llvalprev<-llvalcurr

    ## Compute log-likelihood
    llvalcurr<-(((sum(X*Y)+(sum((Xobs*Zobs)/2))+((n-r)*llvalprev))/(2*n)))
    k<<-k+1
  }
    print(c(llvalprev,llvalcurr,k))
}

EM(physical,100,0.001,50)
lambda <- 10.69566
new_data <- physical
new_data$E_Y <- lambda/physical$X
new_data$E_Z <- 2*lambda/physical$X
ggplot(data=new_data,aes(x=X, group=1)) +
geom_line(aes(y = Y, color = "Y")) +
geom_line(aes(y = Z, color = "Z")) +
geom_line(aes(y = E_Y, color = "E_Y")) +
geom_line(aes(y = E_Z, color = "E_Z")) +
ggtitle("Plot of Y,Z and their expected value vs. X")
```