732A90: Lab 6
Computational Statistics, Group 6

Chih-Yuan Lin, Sarah Walid Alsaadi, Carles Sans Fuentes, Joshua Burrata

March 18, 2017

Question 1

In this exercise, we will try to perform one-dimensional maximization with the help of a genetic algorithm following the steps written below.

Given the functions *gen*, *crossover* and *mutate* defined as follows:

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

$$crossover(x, y) = (x + y)/2$$

$$mutate(x) = x^2 \mod 30$$

We define the function $myfunction()$, which is a function of *maxiter*, the maximum number of iterations , and *mutprob*, the probability of a mutation. Our initial population is denoted by $X = \{0, 5, 10, \ldots, 30\}$, and Y is $gen(X)$. In each iteration, a "victim" from the populations is chosen such that $gen(victim)$ is the smallest. We also generate two random numbers (x_1, x_2) from the set X to be the "parents" and compute the "kid" as $crossover((x_1, x_2))$. This "kid" will be mutated with probability *mutprob* and will replace the "victim".

In each iteration, we also store the maximum of Y . This will be used to see if there is an improvement compared to the previous maximum.

In order to see whether we can we find good results on our data a plot from gen(1 to 30) is shown below:

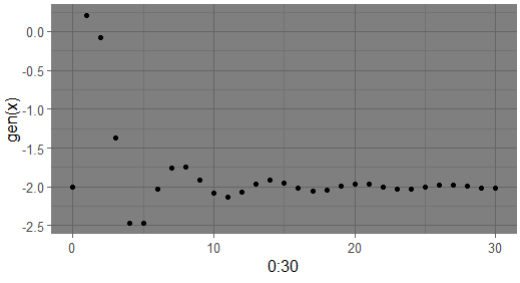

Figure 1: A plot function of gen(x) being x from 0 to 30.

It can be seen that our optimal point must be between 1 and 2.

Now, different plots and approaches have been tried in order to try to find the best optima using maxiter and mutprob. Maxiter have been tried for 10 and 100 and mutprob for 0.1,0.5,0.9. The result of it can be found in figure 2.

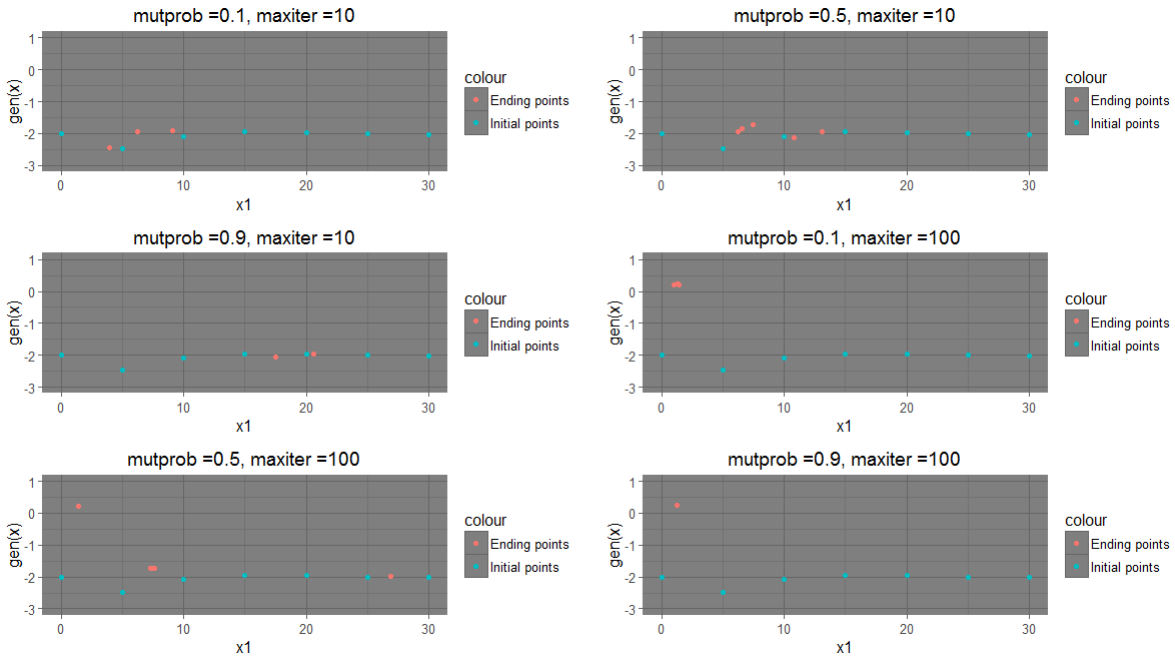

Figure 2: A plot of the gen(x) for the different values of *maxiter* and *mutprob*.

It can be seen that when mutprob is low , then the function is more probable to get stuck on local optimas whereas when you mutate a lot there is a chance to find a better local optima if not the best optimal value. The blues stands for the initial points whereas the red stands for the ending points. It is possible that when just one point is seen, these means that they are all concentrated, but there are 7 points.

Question 2

2.1

The data file physical.csv describes a behavior of two related physical processes $Y = Y(X)$ and $Z = Z(X)$. The following figure shows the time series plot describing the 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?

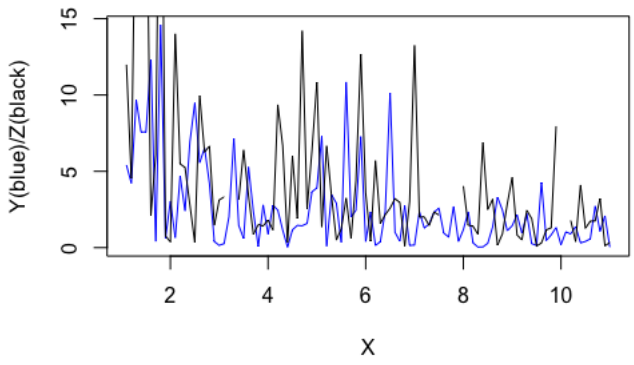

Figure 3: A plot of the Y versus X (in blue) and Z versus X (in black).

It seems that the two processes are related to each other, they look alike. The variation seems to decrease with larger values of X. Since the random variable Z has missing data and Z is related to the observed variable Y, the use of the EM algorithm i motivated, see page 297 in the book.

2.2

Since there are some missing values of Z in the data, this will imply problems in estimating models by maximum likelihood, so we need to find another solution to estimate the parameter. Consider the following model

$$Y_i \sim exp(\frac{X_i}{\lambda})$$

and

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

where λ is some unknown parameter. This implies that $E[Y_i] = \frac{\lambda}{X_i}$ and $E[Z_i] = \frac{2\lambda}{X_i}$ We will here derive an EM algorithm that estimates λ.

The loglikelihood for the complete data is given by:

$$l_{L_c}(\lambda; Y, Z) = -m*ln(2) - (n+m)*ln(\lambda) + \sum_i^n ln(X_i) + \sum_i^m ln(X_i) - \frac{1}{\lambda}\sum_i^n X_i*Y_i - \frac{1}{2\lambda}\sum_i^m X_i*Z_i$$

where n =number of observations in Y and m =number of observations in Z.

Now, let $Zmiss$ denote the missing data in Z and r = the number of missing data. Let also $Zobs$ denote the observed data in Z. From this, we get the E-step of the EM algorithm as follows:

$$q^{(k)}(\lambda) = E_{Zmiss|(Y,Zobs),\lambda^{(k-1)}}(l_{L_c}(\lambda|(Y, Zobs), Zmiss)) =$$

$$-m*ln(2) - (n+m)*ln(\lambda) + \sum_i^n ln(X_i) + \sum_i^m ln(X_i) - \frac{1}{\lambda}\sum_i^n X_i*Y_i - \frac{1}{2\lambda}(\sum_{Zobs} X_i*Z_i + r*X_i\frac{2\lambda^{(k-1)}}{X_i})$$

$$-m*ln(2) - (n+m)*ln(\lambda) + \sum_i^n ln(X_i) + \sum_i^m ln(X_i) - \frac{1}{\lambda}\sum_i^n X_i*Y_i - \frac{1}{2\lambda}\sum_{Zobs} X_i*Z_i + r*\lambda^{(k-1)}$$

.

The M-step is to determine λ that maximizes the function $q^{(k)}(\lambda)$. Differentiating $q^{(k)}(\lambda)$ with respect to λ we get:

$$\lambda^{(k)} = \frac{\sum_i^n X_i*Y_i + \frac{1}{2}\sum_{Zobs} X_i*Z_i + r*\lambda^{(k-1)}}{n+m}$$

This is a maximum since the second derivative is negative.

2.3

We then implement this algorithm in R, where we use $\lambda_0 = 100$ and convergence criterion "stop if the change in is less than 0.001". The algorithm converges to $\lambda = 10.695$ after 5 iterations. Observe that the maximum likelihood estimation of λ for the observed data, that is $(Y, Zobs)$ gives us the same result as the one obtained.

2.1

Figure 4 shows the plot of Y versus X (red) and the the computed λ (blue). The blue curve follows the red curve quit good. The decreasing variation is also explained since the variance of $Y_i \sim exp(\frac{X_i}{\lambda})$ is $(\frac{\lambda}{X_i})^2$. The same conclusion is drawn from figure 5.

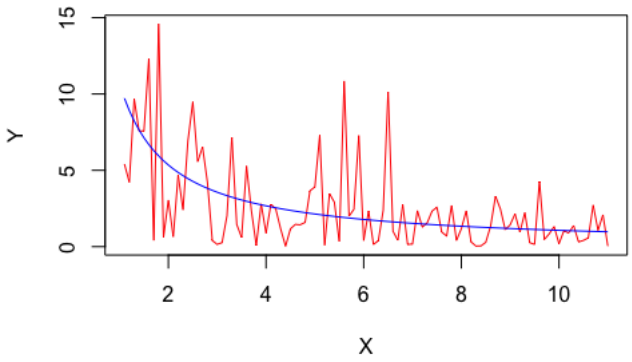

Figure 4: A plot of the Y versus X (in red) and $E(Y)$ versus X (in blue).

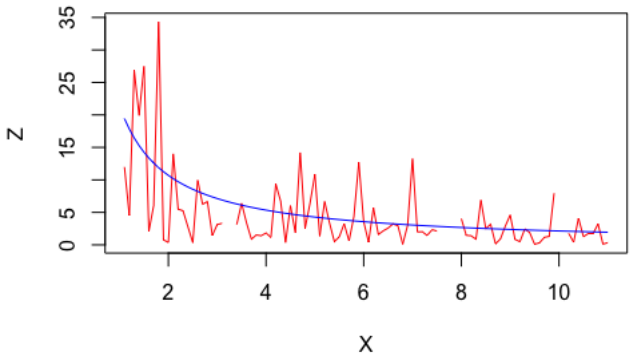

Figure 5: A plot of the Z versus X (in red) and $E(Z)$ versus X (in blue)..

Contributions

All results and comments presented have been developed and discussed together by the members of the group.

Appendix

Question 1

```
####LAB 6
##1

gen<- function(data){
  myformula<-(data**2/exp(data))- 2*exp(-(9*sin(data))/(data**2+data+1))
  return(myformula)
}


##2

crossover<- function(x){
  return(sum(x)/2)
}


##3

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

##4
myfun<-function(maxiter,mutprob){
  ##a
  x<-0:30
  plot1<-ggplot(as.data.frame(gen(x)))+
    geom_point(aes(x= 0:30, y = gen(x)))+
    theme_dark()
  plot1
  ##b initial population
  X<- seq(0,30,5)
  ##C
  values<-  gen(X)
  mydata<-data.frame(X= X, Y = values)
  max<- integer(0)
  max[1]<-max(mydata[,2])

  ##d
  max(integer(maxiter+1))
  for(i in 1:maxiter){
    ##i
    mychoice<-sample(1:nrow(mydata),2)##
    ##ii
    index<-which.min(mydata[,2])
    ##iii
    mydata[index,1]<-crossover(mydata[mychoice,1])
    if(runif(1)<mutprob){
      mydata[index,2]<-gen(mydata[index,1])
    }else{
      mymut<-mutate(mydata[index,1])
      mydata[index,1]<- mymut
      mydata[index,2]<-gen(mydata[index,1])
    }

    max[i+1]<-max(mydata[,2])

  }

  initialvalues <- as.character(as.numeric(seq(0,30,5) %in% mydata[,1]))##green initial values
  myframe<-data.frame(x1= mydata[,1],y1=mydata[,2],Initial=as.factor(initialvalues),invalues=
      gen(X))
  plotting<- ggplot(myframe)+
    geom_point(aes(x= x1, y = y1, col="Ending points"))+
    geom_point(aes(x=seq(0,30,5), y =invalues, col="Initial points"))+ coord_cartesian(xlim = c
        (0, 30), ylim = c(-3, 1))+
    ylab("gen(x)")+ggtitle(paste0("mutprob =",mutprob,", maxiter =", maxiter ,collapse = " "))+
    theme_dark()
  result<- list(maxvalues =max, plot= plotting, thedata =mydata)
  return(result)
  }

library(ggplot2)

set.seed(12345)
gen1001<-myfun(maxiter = 10, mutprob = 0.1)
gen1005<-myfun(maxiter = 10, mutprob = 0.5)
gen1009<-myfun(maxiter = 10, mutprob = 0.9)
gen10001<-myfun(maxiter = 100, mutprob = 0.1)
gen10005<-myfun(maxiter = 100, mutprob = 0.5)
gen10009<-myfun(maxiter = 100, mutprob = 0.9)
```

```
library(gridExtra)
genpicture<-grid.arrange(gen1001$plot,gen1005$plot, gen1009$plot, gen10001$plot, gen10005$plot,
    gen10009$plot, ncol =2)
h
```

Question 2

```
##Question 2

#2.1.
plot(physical$X,physical$Y,type="l",col="blue",xlab="X",ylab="Y(blue)/Z(black)")
lines(physical$X,physical$Z, type="l",col="black")



#2.2

modelY<-function(physical,lambda){
 n=length(physical$X)
 s<-numeric(n)
  for (i in 1:n){
    s[i]<-exp(physical$X[i]/lambda)
  }
 return(s)
}


modelZ<-function(physical,lambda){
  n=length(physical$X)
  t<-numeric(n)
  for (i in 1:n){
    t[i]<-exp(physical$X[i]/(2*lambda))
  }
  return(t)
}


#2.3


EM<-function(physical,eps,kmax){
  X<-physical$X
  Y<-physical$Y
  Z<-physical$Z
  Zobs<-Z[!is.na(Z)]
  Zmiss<-Z[is.na(Z)]
  Xobs<-X[!is.na(Z)]
  n<-length(Y)
  m<-length(Z)
  k<-1
  previous<-0
  current<-100
  print(c(previous,current))
  while ((abs(current-previous)>eps)&& (k<(kmax+1))){
    k<-k+1
    previous<-current
    current<-(sum(X*Y)+length(Zmiss)*previous+(1/2)*sum(Xobs*Zobs))/(n+m)

  }
 print(c(previous,current))
}


#2.4

EY<-function(physical){
  X<-physical$X
  Y<-physical$Y
  n=length(Y)
  s<-numeric(n)
  for (i in 1:n){
    s[i]<-10.695/X[i]
  }
  return(s)
}



EZ<-function(physical){
  Z<-physical$Z
  X<-physical$X

```

```
  n=length(Z)
  t<-numeric(n)
  for (i in 1:n){
    t[i]<-2*10.695/X[i]
  }
  return(t)
}


plot(physical$X,physical$Y,type="l",col="red",xlab="X",ylab="Y")
lines(physical$X,EY(physical),type = "l",col="blue",xlab="X",ylab="E(Y)")

plot(physical$X,physical$Z, type="l",col="red",xlab="X",ylab="Z")
lines(physical$X,EZ(physical),type = "l",col="blue",xlab="X",ylab="E(Z)")
```