

# Genetic & EM Algorithm

Sreenand

3/4/2020

## Question 1: Genetic algorithm

### Question 1.1

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

### Question 1.2

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

### Question 1.3

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

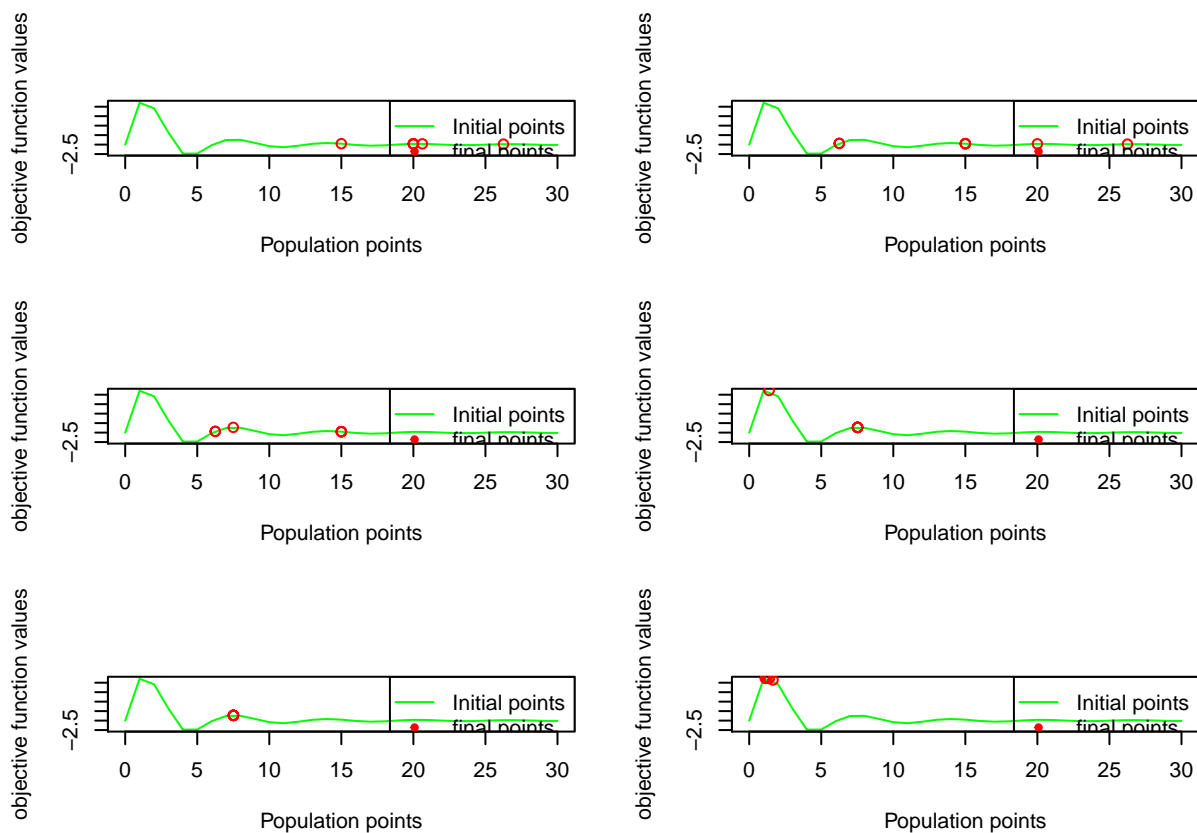
### Question 1.4

```
myfunc<-function(maxiter,mutprob){
  plot(0:30,f(0:30),type="l",col = "green",xlab = "Population points",ylab = "objective function values")
  X= seq(0,30,5)
  values=f(X)
  maximum_value=0
  for(i in 1:maxiter){
    ind<-sample(1:length(X),2)
    victim=order(values)[1]
    kid=crossover(X[ind[1]], X[ind[2]])
    if(runif(1,0,1)<=mutprob){
      kid=mutate(kid)
    }
  }
}
```

```

X[victim]=kid
values=f(X)
maximum_value[i]<-max(values)
}
points(X,values,col="red")
legend(x = "topright", legend = c("Initial points", "final points"), pch = c(0,20,4),
col = c("green", "red"), lty = c(1,0,0), pt.cex = c(0, 1, 1))
}
par(mfrow=c(3,2))
myfunc(10,0.1)
myfunc(10,0.5)
myfunc(10,0.9)
myfunc(100,0.1)
myfunc(100,0.5)
myfunc(100,0.9)

```



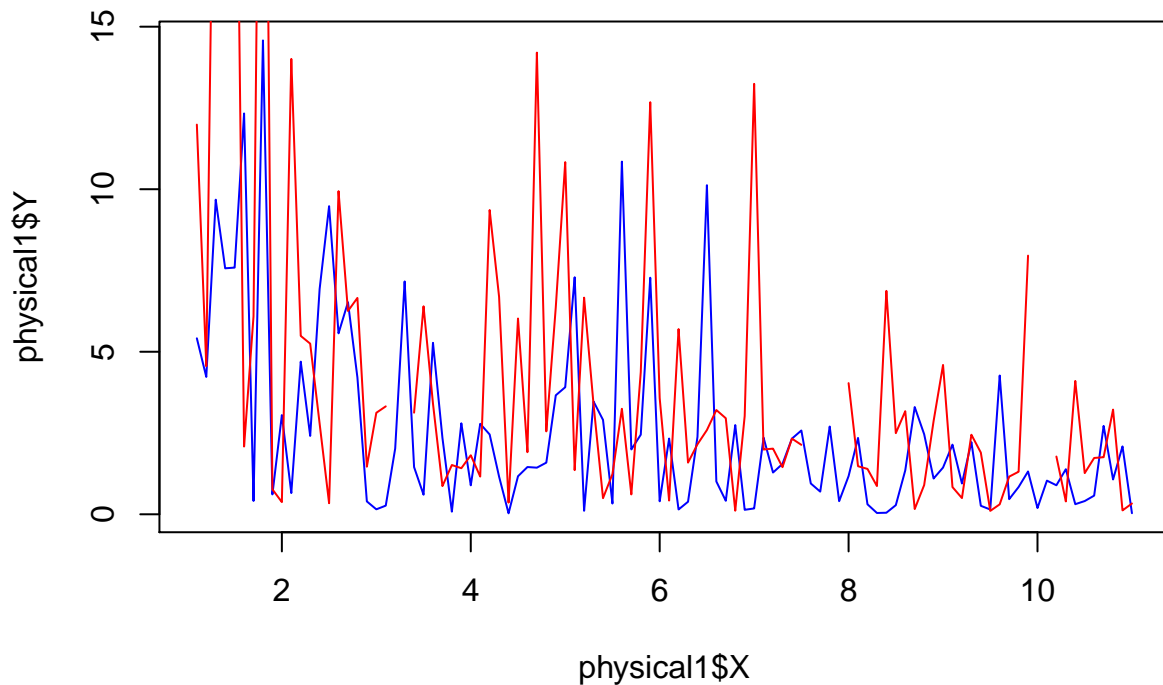
From the graphs obtained from 10 iteration we can say that the function converges only to the local maximums, but as the number of iterations increased to 100 the function gives the maximum value. While coming to the probability of the mutation the function yielding maximum value with increase of the the probability. We can clearly say that with 100 iterations and mutation probability greater than or equal to 0.5 is giving the maximum value.

## Question 2: EM algorithm

## Question 2.1: Time series plots

```
library(readr)
physical1 <- read_csv("physical1.csv")

plot(physical1$X,physical1$Y,col = "blue",type = "l")
points(physical1$X,physical1$Z,col = "red",type = "l")
```



### 2.1

From the graph, we can see that Y and Z are closely correlated to each other when plotted against X. Both Y and Z decrease with the increase of X. The peaks of Z and Y are almost at the same intervals. There seem to be missing points in the Z dataset.

### 2.2

We can see there are missing data in the Z dataset and hence we predict the missing values. We use the following models:

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

$$Z_i \sim \exp(X_i/2\lambda)$$

and derive an EM Algorithm to estimate the  $\lambda$  values.

We find the log-likelihood of these models and we arrive at the equation:

$$l(Y, Z|\lambda) = \log\left(\frac{\prod_{i=1}^n X_i}{2^n \lambda^{2n}}\right) - \frac{\sum_{i=1}^n Y_i X_i}{\lambda} - \frac{\sum_O Z_i X_i}{2\lambda} - \frac{\sum_M Z_i X_i}{2\lambda}$$

$$E(l(Y, Z|\lambda)) = \log\left(\frac{\prod_{i=1}^n X_i}{2^n \lambda^{2n}}\right) - \frac{\sum_{i=1}^n Y_i X_i}{\lambda} - \frac{\sum_{i=1}^n Z_i X_i}{2\lambda} - \frac{|M|\lambda_t}{\lambda}$$

$|M|$  is the number of missing Z values here. After equating to zero we arrive at the equation,

$$\lambda_{t+1} = \frac{\sum_{i=1}^n Y_i X_i}{2n} + \frac{\sum_{i=1}^n Z_i X_i}{4n} + \frac{|M|\lambda_t}{\lambda}$$

### 2.3

We implement the algorithm using the initial  $\lambda$  value as  $\lambda_o = 100$  and stop when the  $\lambda$  value is less than 0.001. The optimal lambda value is found to be 10.69566 and the maximum number of iterations it took to compute the optimal lambda is 5.

### Question 2.2: EM algorithm

```
phy <- read_csv("physical1.csv")
X <- phy$X
Y <- phy$Y
miss_Z <- phy$Z[which(is.na(phy$Z))]
inc_Z <- phy$Z[which(!is.na(phy$Z))]
inc_X <- X[which(!is.na(phy$Z))]
n <- nrow(phy)
lambda <- 100
diff <- Inf
M <- length(miss_Z)
iter <- 0
while(diff > 0.001)
{
  lambda_upd <- (sum(X * Y)/(2*n)) + (sum(inc_X * inc_Z) / (4*n)) + ((M * lambda)/(2*n))
  diff <- abs(lambda_upd - lambda)
  lambda <- lambda_upd
  iter <- iter + 1
  print(lambda)
}
```

```
## [1] 14.26782
## [1] 10.83853
## [1] 10.70136
## [1] 10.69587
## [1] 10.69566
```

```
cat("The number of iterations it took to find optimal Lambda value is ", iter)
```

```
## The number of iterations it took to find optimal Lambda value is 5
```

```
cat("\nThe optimal lambda value is ", lambda)
```

```
##
```

```
## The optimal lambda value is 10.69566
```

```
E_Y <- lambda / X
```

```
cat("\nThe Expected value of Y is ", E_Y)
```

```
##
```

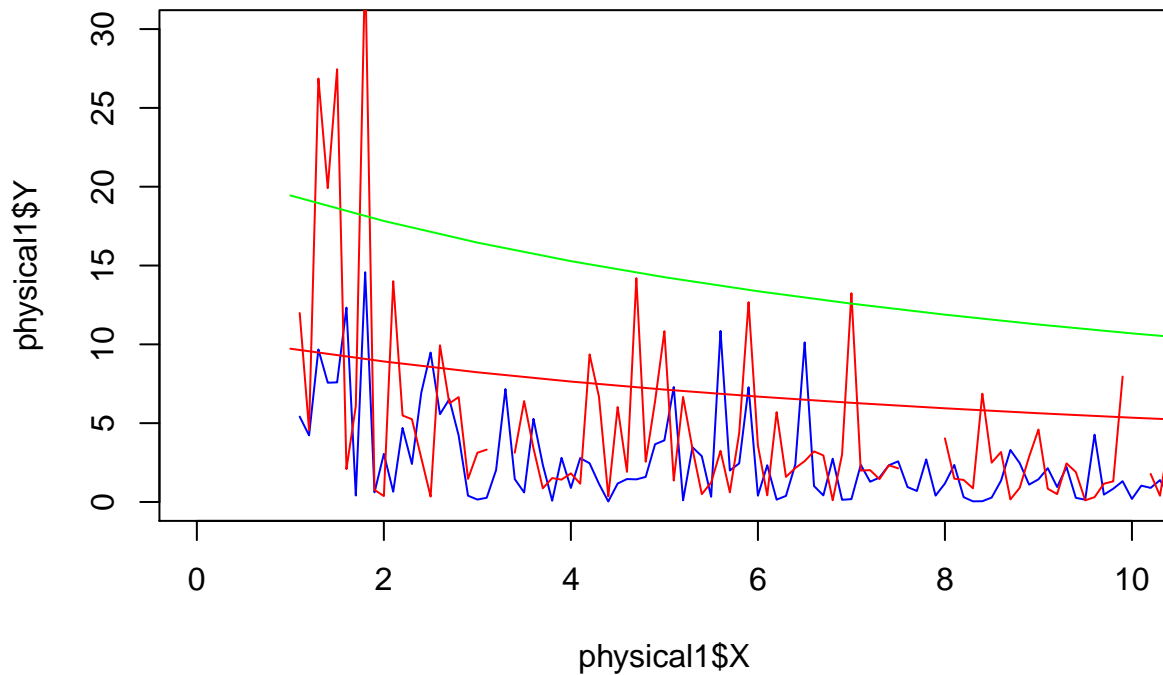
```
## The Expected value of Y is 9.723323 8.913046 8.227427 7.639754 7.130437 6.684785 6.291562 5.942031
```

```

E_Z <- (2 * lambda) / X
cat("\nThe Expected value of Z is ", E_Z)

##
## The Expected value of Z is  19.44665 17.82609 16.45485 15.27951 14.26087 13.36957 12.58312 11.88406
#plot(X,Y,type = "l",col = "blue",xlim = c(0,20), ylim = c(0,30))
#lines(phy$Z,type = "l",col = "black")
plot(physical1$X,physical1$Y,col = "blue",type = "l", ylim = c(0,30),xlim = c(0,10))
lines(physical1$X,physical1$Z,col = "red",type = "l")
lines(E_Y, col = "red")
lines(E_Z, col = "green")

```



## 2.4

When  $E[X]$  and  $E[Y]$  are plotted against  $X$  on the same graph we see that it makes a smooth line covering both the distributions and hence we can deduce that the computed  $\lambda$  value is reasonable where  $E[X]$  and  $E[Y]$  are given by,

$$E[Y] = \lambda/X \text{ and } E[Z] = 2\lambda/X$$