Computational Statistics

Lab 6

Emil K Svensson and Rasmus Holm

2017-03-14

Question 1

1.1

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

1.2

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

1.3

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

1.4

```
genetic <- function(maxiter, mutprob) {
    ## a)
    ## plot(x = 0:30, y= genfunc(0:30),  xlim = c(0,30), type ="l", xlab="", ylab="")

    ## b)
    X <- seq(0,30,by = 5)

    ## c)
    values <- genfunc(X)
    ## points(X, values, col = "red")

    ## d)
    bestvalue <- -Inf

    for (i in 1:maxiter){

        ## i
```

```
        parents <- sample(1:length(X), size = 2 )

        ## ii
        victim <- which.min(values)

        ## iii
        child <- crossover(X[parents[1]],X[parents[2]])

        if (mutprob > runif(1,0,1)) {
            child <- mutate(child)
        }

        ## iv
        X[victim] <- child

        values[victim]<- genfunc(child)
        ## values <- genfunc(X)

        ## v
        bestvalue <- max(bestvalue, max(values))
    }

    ## points(x = X, y = values, col = "darkgreen")
    list(opt=bestvalue, pop=X, vals=values)
}
```

1.5

```
library(ggplot2)

func_data <- data.frame(x=seq(0, 30, by=0.1), y=genfunc(seq(0, 30, by=0.1)))

set.seed(123456)
r1 <- genetic(maxiter = 10, mutprob = 0.1)
r1$opt
```

```
## [1] -1.7
```

```
set.seed(123456)
r2 <- genetic(maxiter = 10, mutprob = 0.5)
r2$opt
```

```
## [1] 0.14
```

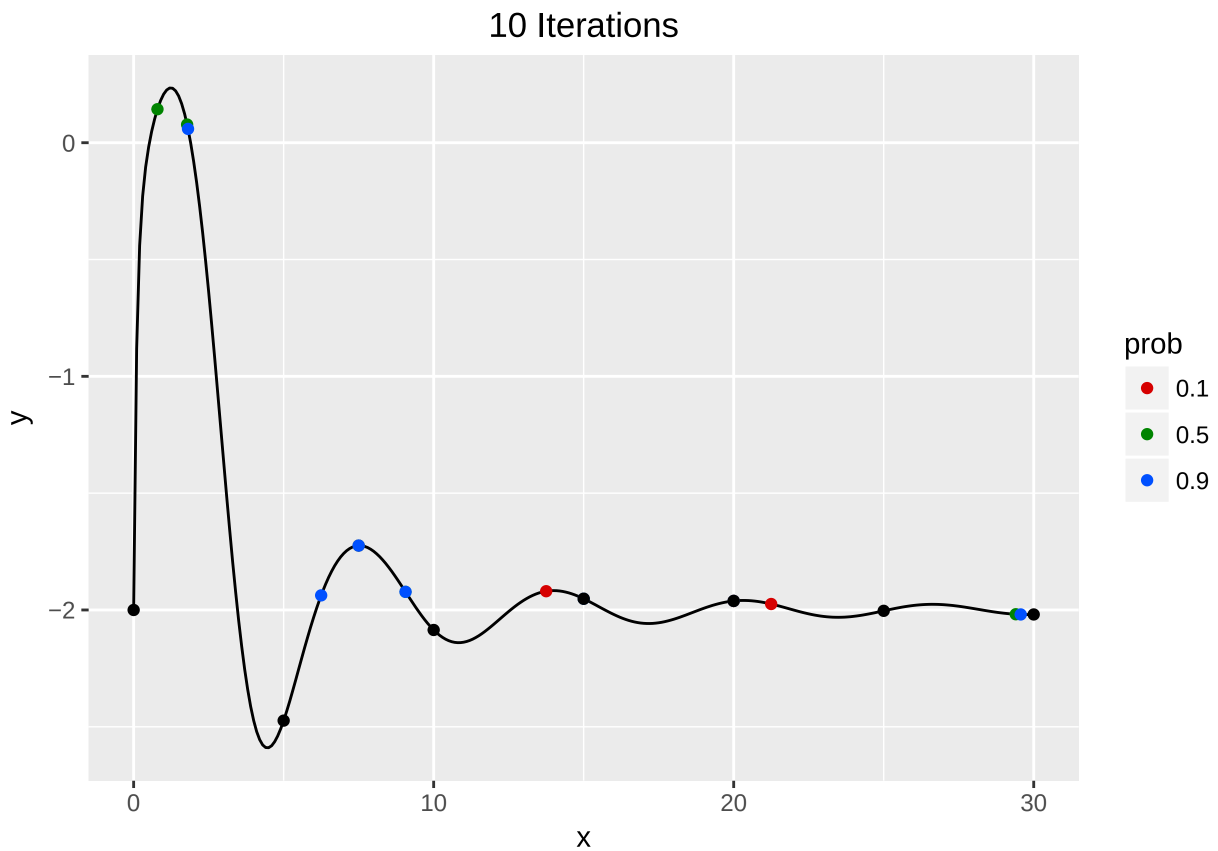
```
set.seed(123456)
r3 <- genetic(maxiter = 10, mutprob = 0.9)
r3$opt
```

```
## [1] 0.059
```

```
rd1 <- data.frame(x=r1$pop, y=r1$vals, prob="0.1")
rd2 <- data.frame(x=r2$pop, y=r2$vals, prob="0.5")
rd3 <- data.frame(x=r3$pop, y=r3$vals, prob="0.9")
```

```r
plot_data <- rbind(rd1, rd2, rd3)

ggplot() +
    ggtitle("10 Iterations") +
    geom_line(data=func_data, aes(x=x, y=y)) +
    geom_point(data=plot_data, aes(x=x, y=y, col=prob), size=1.5) +
    geom_point(aes(x=seq(0,30,by = 5), y=genfunc(seq(0,30,by = 5)))) +
    theme(plot.title=element_text(hjust=0.5)) +
  scale_colour_hue(l=40, c=200)
```




```r
set.seed(123456)
r1 <- genetic(maxiter = 100, mutprob = 0.1)
r1$opt
```

```
## [1] -1.7
```

```r
set.seed(123456)
r2 <- genetic(maxiter = 100, mutprob = 0.5)
r2$opt
```

```
## [1] 0.23
```

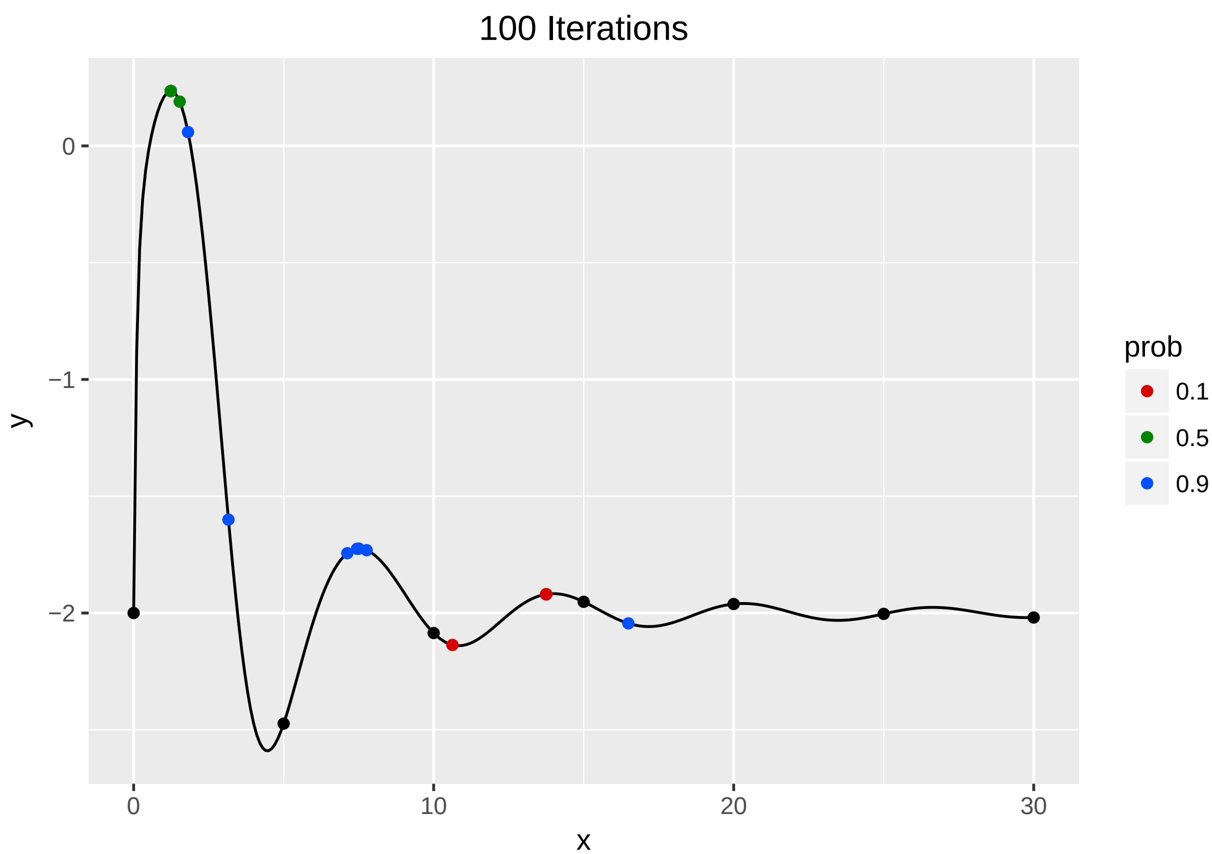
```r
set.seed(123456)
r3 <- genetic(maxiter = 100, mutprob = 0.9)
r3$opt
```

```
## [1] 0.059
```

```r
rd1 <- data.frame(x=r1$pop, y=r1$vals, prob="0.1")
rd2 <- data.frame(x=r2$pop, y=r2$vals, prob="0.5")
rd3 <- data.frame(x=r3$pop, y=r3$vals, prob="0.9")

plot_data <- rbind(rd1, rd2, rd3)

ggplot() +
    ggtitle("100 Iterations") +
    geom_line(data=func_data, aes(x=x, y=y)) +
    geom_point(data=plot_data, aes(x=x, y=y, col=prob), size=1.5) +
    geom_point(aes(x=seq(0,30,by = 5), y=genfunc(seq(0,30,by = 5))))+
    theme(plot.title=element_text(hjust=0.5)) +
      scale_colour_hue(l=40, c=200)
```

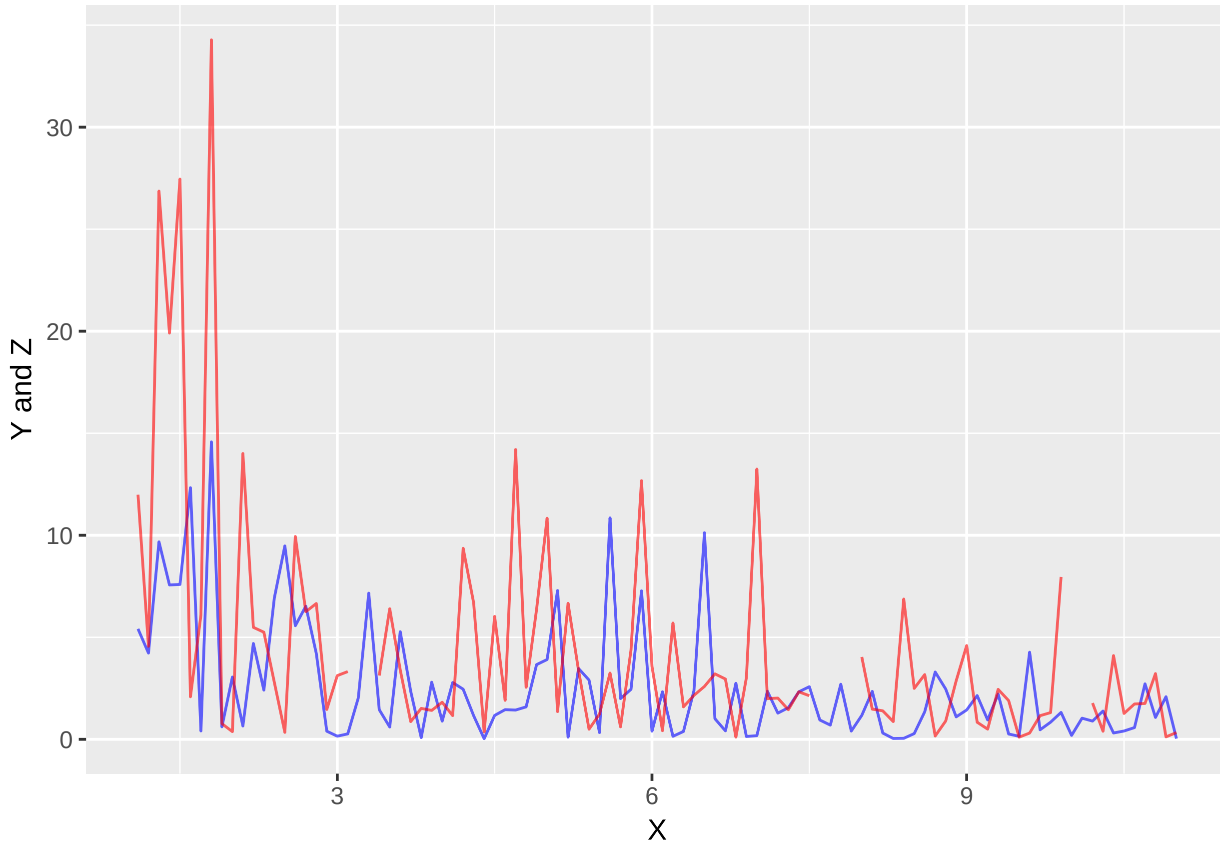


Question 2

2.1

```r
physical <- read.csv("../data/physical1.csv")
```

```r
ggplot(physical) +
    geom_line(aes(x=X, y=Y), col="blue", alpha = 0.6) +
    geom_line(aes(x=X, y=Z), col="red", alpha = 0.6) +
    labs(y = "Y and Z")
```



2.2

We know

$$
\begin{aligned}
Y_i &\sim \exp(X_i/\lambda), \\
Z_i &\sim \exp(X_i/(2\lambda)).
\end{aligned}
$$

To find λ using the EM-algorithm we want to find the likelihood which is

$$
\begin{aligned}
l(\lambda) &= P(\lambda|X, Y, Z) \\
&= \prod_{i=1}^{n} \frac{x_i}{\lambda} \exp\left(-\frac{x_i}{\lambda} y_i\right) \frac{x_i}{2\lambda} \exp\left(-\frac{x_i}{2\lambda} y_i\right) \\
&= \prod_{i=1}^{n} \frac{x_i}{2\lambda^2} \exp\left(-\frac{x_i}{\lambda}\left(y_i + \frac{z_i}{2}\right)\right).
\end{aligned}
$$

Since we know that multiplication with small numbers is bad for accuracy reasons in the computer we want the log-likelihood

$$
\begin{aligned}
L(\lambda) &= \sum_{i=1}^{n} \log\left(\frac{x_i}{2\lambda^2} \exp\left(-\frac{x_i}{\lambda}\left(y_i + \frac{z_i}{2}\right)\right)\right) \\
&= \sum_{i=1}^{n} \left(\log(x_i^2) - 2n\log(2\lambda) - \frac{x_i}{\lambda}\left(y_i + \frac{z_i}{2}\right)\right) \\
&= 2\sum_{i=1}^{n} \log(x_i) - 2n\log(2\lambda) - \sum_{i=1}^{n} \frac{x_i}{\lambda}\left(y_i + \frac{z_i}{2}\right).
\end{aligned}
$$

We don't know the true λ but we do have an estimate and we want to find the expected value so we get

$$
\mathbb{E}\left[L(\lambda)|X, Y, Z, \lambda_k\right] = 2\sum_{i=1}^{n} \log(x_i) - 2n\log(2\lambda) - \mathbb{E}\left[\sum_{i=1}^{n} \frac{x_i}{\lambda}\left(y_i + \frac{z_i}{2}\right)|X, Y, Z, \lambda_k\right].
$$

The Z data is partially observed so we decompose it into two variables, observed and unobserved, as $Z = \{V, W\}$ where V is the observed part, $|V| = r$, and W is the unobserved part, $|W| = n - r$. Then we get

$$
\begin{aligned}
\mathbb{E}\left[L(\lambda)|X, Y, V, \lambda_k\right] &= 2\sum_{i=1}^{n} \log(x_i) - 2n\log(2\lambda) - \sum_{i=1}^{n} \frac{x_i}{\lambda} y_i - \sum_{i=1}^{r} \frac{x_i}{2\lambda} v_i - \sum_{i=r+1}^{n} \frac{x_i}{2\lambda} \mathbb{E}\left[w_i|X, Y, V, \lambda_k\right] \\
&= 2\sum_{i=1}^{n} \log(x_i) - 2n\log(2\lambda) - \sum_{i=1}^{n} \frac{x_i}{\lambda} y_i - \sum_{i=1}^{r} \frac{x_i}{2\lambda} v_i - \sum_{i=r+1}^{n} \frac{x_i}{2\lambda} \frac{2\lambda_k}{x_i} \\
&= 2\sum_{i=1}^{n} \log(x_i) - 2n\log(2\lambda) - \sum_{i=1}^{n} \frac{x_i}{\lambda} y_i - \sum_{i=1}^{r} \frac{x_i}{2\lambda} v_i - (n-k)\frac{\lambda_k}{\lambda}
\end{aligned}
$$

To maximize λ we take the derivative and set it to zero and find

$$
\lambda = \frac{1}{2n}\left(\sum_{i=1}^{n} x_i y_i + \frac{1}{2}\sum_{i=1}^{r} x_i z_i + (n-r)\lambda_k\right)
$$

2.3

```
EM <- function(data, lambdazero, maxiter = 500, eps = 0.001){

  Estep <- function(data,lambda){
        r <- sum(is.na(data$z))
        n <- nrow(data)

        (2 * sum(log(data$x)) - 2 * n * log(2 * lambda) -
         sum(data$x * data$y)/lambda -
         sum(data$x * data$z, na.rm = TRUE)/(2*lambda) - (n - r))

    }

    Mstep <- function(data, lambda){

        n <- nrow(data)
        r <- sum(is.na(data$z))

        ((sum(data$x * data$y) +
          sum(data$x * data$z, na.rm = TRUE) / 2 +
          (n - r) * lambda) / (2 * n))
    }

    curlambda <- lambdazero
    prevlambda <- lambdazero*5
    iter <- 1

    while(iter < maxiter && (abs(curlambda - prevlambda) > eps)){

        prevlambda <- curlambda
        curlambda <- Mstep(data,lambda = curlambda)

        iter <- iter + 1
    }

    return(list(lambda = curlambda, iter = iter))
}

colnames(physical) <- c("x","y","z")
res <- EM(data = physical, lambdazero = 100)
print(res)
```

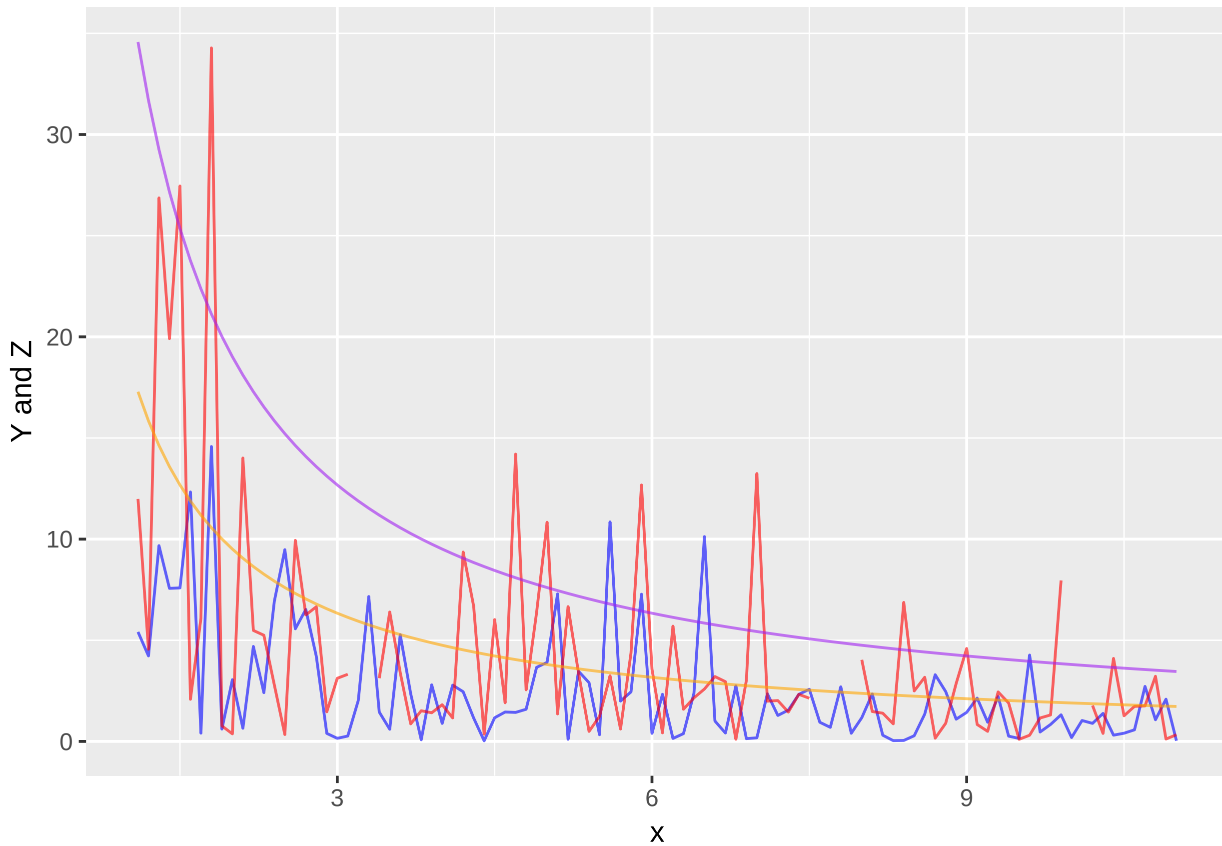
```
## $lambda
## [1] 19
##
## $iter
## [1] 16
```

After six iteration the algorithm converges with a lambda value of 10.69.

2.4

```
lamb <- res$lambda
physical$EZ <- (2*lamb)/physical$x
physical$EY <- lamb/physical$x

ggplot(physical) +
    geom_line(aes(x=x, y=y), col="blue", alpha = 0.6) +
    geom_line(aes(x=x, y=z), col="red", alpha = 0.6) +
    geom_line(aes(x=x, y=EZ), col="purple", alpha = 0.6) +
    geom_line(aes(x=x, y=EY), col="orange", alpha = 0.6) +
    labs(y = "Y and Z")
```



Overall the expected values for both Y and Z looks resonable and cutting through the data at resonable levels although the data is a bit chaotic.

The expected values for Z i.e the purple ones looks like a fair approximation of where the missing values are located. For example around x = 7 the purple line is around where the red line has a gap.