Splines

Sergi Carol Laura Cebollero Alex Rodriguez

December 13, 2018

Introduction

The aim of this lab is to understand the spline smoothing effects on regression methods, the lab will consist of two different tasks, the first one will combine the use of a **b-spline** basis with a liniar model in order to compute the optimal fitting for our data. The second task will use the method **smooth spline** in order to calculate the optimal fitting.

Exercice 1

```
summary(meat)
```

```
##       Fat           abs.850         abs.957
##  Min.   : 0.90   Min.   :2.066   Min.   :2.572
##  1st Qu.: 7.30   1st Qu.:2.512   1st Qu.:3.083
##  Median :14.00   Median :2.754   Median :3.382
##  Mean   :18.14   Mean   :2.809   Mean   :3.462
##  3rd Qu.:28.00   3rd Qu.:3.006   3rd Qu.:3.714
##  Max.   :49.10   Max.   :4.237   Max.   :5.299
```

```
y = log(meat$Fat)
x = meat$abs.850

# sort data
sx = sort(x, index.return=T)
x = sx$x
y = y[sx$ix]
```

You can also embed plots, for example:

```
k = 3   #Todo: check what happens if degree =1

n.knots = 10
my.knots <- quantile(x,seq(0,1,length=n.knots))
inner.knots <- my.knots[-c(1,length(my.knots))]


degrees <- n.knots + k + 1
```

```
computeFittedValues <- function(x, n.knots, k, y) {
    my.knots <- quantile(x,seq(0,1,length=n.knots))
    l = length(my.knots)
    inner.knots <- my.knots[-c(1,l)]

    df = n.knots + k + 1
    basis <- bs(x=x,knots=inner.knots,intercept=T,degree=k, df=df)
    lm.spl <- lm(y~basis-1) # remove intercept
    return(lm.spl$fitted.values)
}


f.10.CV.inner.knots <- function(x, y, k, n.knots.range) {

    r.sq.array = c()
    possible.knots = seq(n.knots.range[1], n.knots.range[2])
    for(n.knots in possible.knots){
        fitted.vals = computeFittedValues(x, n.knots, k, y)
        (r.sq = sum(((y - fitted.vals)^2)/length(y)))
        r.sq.array = append(r.sq.array, r.sq)
    }

    data.f = data.frame(possible.knots, r.sq.array)
    print(data.f)

    idx = which(min(r.sq.array) == r.sq.array)
    return(possible.knots[idx])
}
```

```
n.knots.optim = f.10.CV.inner.knots(x, y, 3, c(1, 20))
```

```
##    possible.knots r.sq.array
## 1               1  0.4926914
## 2               2  0.4926914
## 3               3  0.4821762
## 4               4  0.4714158
## 5               5  0.4685615
## 6               6  0.4698841
## 7               7  0.4664103
## 8               8  0.4474071
## 9               9  0.4560541
## 10             10  0.4425412
## 11             11  0.4385915
## 12             12  0.4369033
## 13             13  0.4379402
## 14             14  0.4391639
## 15             15  0.4372359
## 16             16  0.4296704
## 17             17  0.4159069
## 18             18  0.4091283
## 19             19  0.4026160
## 20             20  0.4134358
```

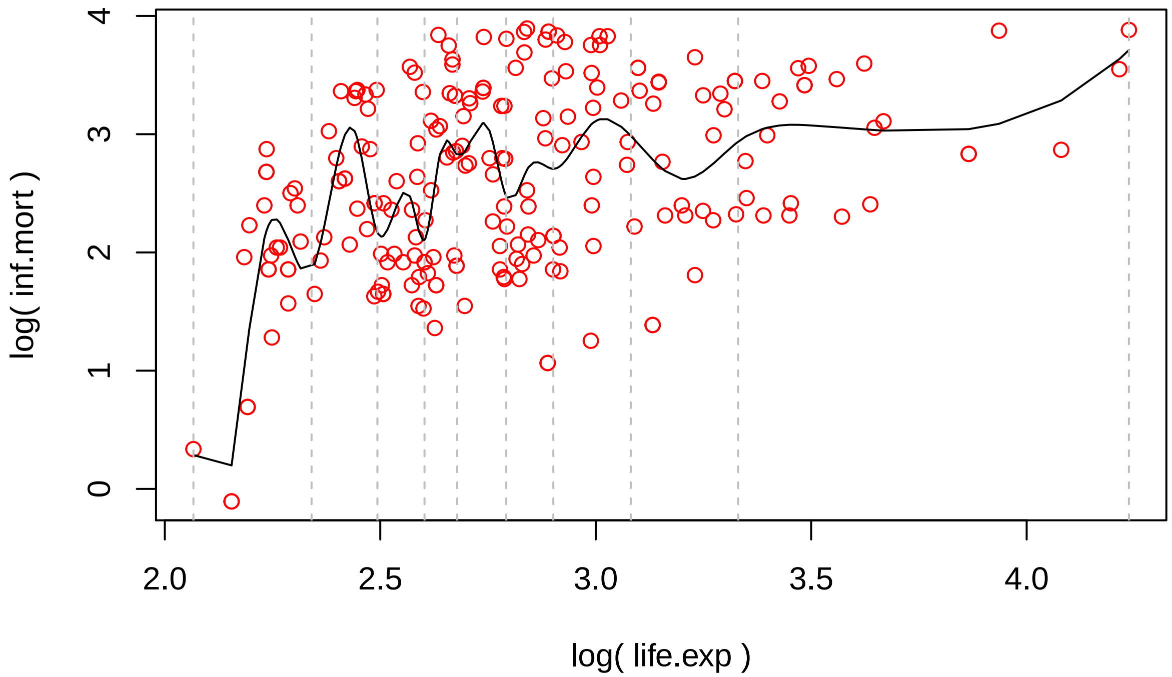
```
fitted.vals.optim = computeFittedValues(x, n.knots.optim, k, y)
my.knots = quantile(x, seq(0,1,length=n.knots.optim))

n.knots = 10
my.knots <- quantile(x,seq(0,1,length=n.knots))
inner.knots <- my.knots[-c(1,length(my.knots))]

degrees <- n.knots + k + 1

basis <- bs(x=x,knots=inner.knots,intercept=T,degree=k, df=degrees)
```

```
dim(basis)
```

```
## [1] 215  12
```

```
lm.spl <- lm(y~basis-1)
```

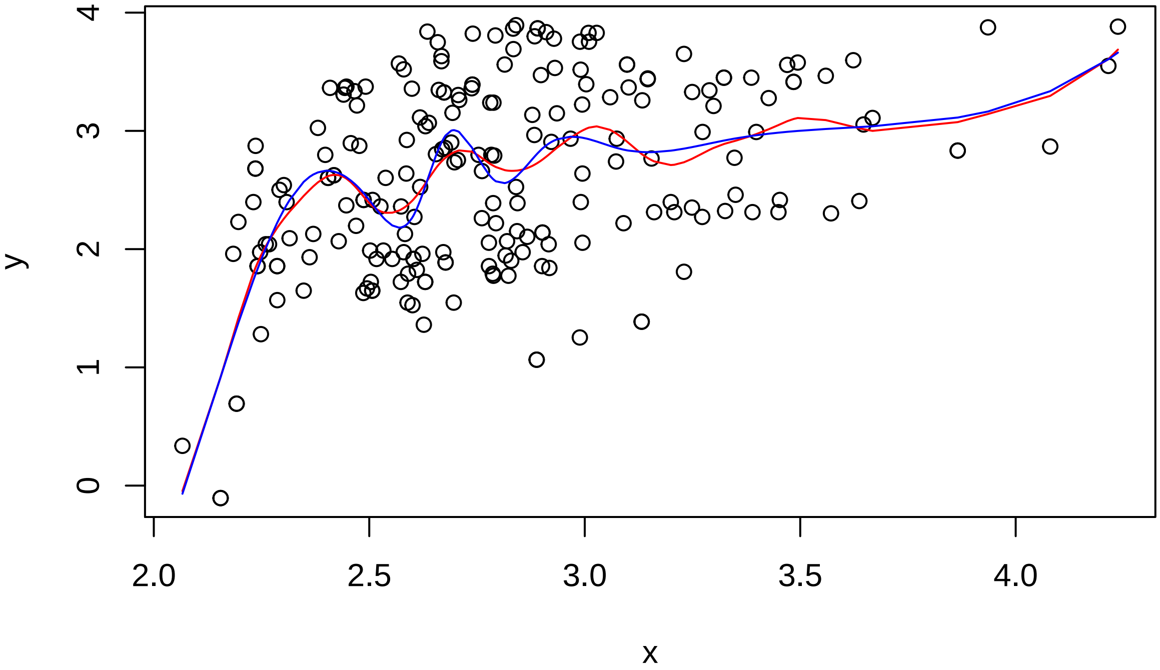
```
plot(x,y,col=2,xlab="log( life.exp )",ylab="log( inf.mort )")
lines(x,fitted.vals.optim)
abline(v=my.knots,lty=2,col="grey")
```



Exercice 2

In this exercice we will use the method smooth spline with the previous calculated degrees of freedom and we will plot the results of both this method and the previous used method in order to compare them.

```
m1  <- smooth.spline(x, y, df = degrees)
plot(x,y)
lines(m1, col="red")
lines(x,lm.spl$fitted.values, col="blue")
```



0
1
2
3
4
y
2.0
2.5
3.0
3.5
4.0
x