Lecture 8

Jussi Klemelä

March 17, 2009

1 Partially linear methods

A partially linear model is given by

$$f(x,z) = x^T\beta + g(z), \qquad (x,z) \in \mathbf{R}^p \times \mathbf{R}^q,$$

where $\beta \in \mathbf{R}^p$ is an unknown vector and $g : \mathbf{R}^q \to \mathbf{R}$ is an unknown function. Note that the linear part does not contain an intercept, because it could not be identified separately from the unknown function g.

- An estimator for β is given by

$$\hat\beta = \left[\sum_{i=1}^n \tilde X_i \tilde X_i^T\right]^{-1} \sum_{i=1}^n \tilde X_i \tilde Y_i, \tag{1}$$

where

$$\tilde X_i = X_i - \hat X_i, \qquad \tilde Y_i = Y_i - \hat Y_i,$$

and $\hat X_i$ is a kernel regression estimator for the conditional mean $E[X \mid Z]$ evaluated at Z_i, and $\hat Y_i$ is a kernel regression estimator for the conditional mean $E[Y \mid Z]$ evaluated at Z_i. Since $X \in \mathbf{R}^p$ is a vector, then $\hat X_i \in \mathbf{R}^p$ is also a vector. Thus we have

$$\hat X_i = \hat f(Z_i), \qquad \hat f(z) = \sum_{j=1}^n p_j(z) X_j,$$

where

$$p_j(z) = \frac{K_h(z - Z_j)}{\sum_{j=1}^n K_h(z - Z_j)}, \qquad j = 1, \dots, n, \tag{2}$$

$K : \mathbf{R}^q \to \mathbf{R}$ is the kernel function, $K_h(x) = K(x/h)/h^q$, and $h > 0$ is the smoothing parameter. Random variables $\hat Y_i$ are defined similarly.

The estimator can be motivated by the facts that if we take conditional expectations of

$$Y = X^T\beta + g(Z) + \epsilon, \tag{3}$$

then we get

$$E(Y \mid Z) = E(X \mid Z)\beta + g(Z).$$

Subtracting, we get

$$Y - E(Y \mid Z) = (X - E(X \mid Z))^T\beta + \epsilon.$$

This linear regression model can be solved to get an estimator of β, but the unknown conditional excpectations $E(Y \mid Z)$ and $E(X \mid Z)$ has to be estimated, and this has been done with kernel regression.

- From (3) we get

$$g(Z) = E(Y - X^T\beta \mid Z).$$

Inserting the estimator β from (1) we can define an estimator for g as a kernel estimator

$$\hat{g}(z) = \sum_{i=1}^{n} p_i(z)(Y_i - X_i^T\hat{\beta}),$$

where $p_i(z)$ are defined in (2).

2 Illustrations

We look at the following code

http://cc.oulu.fi/~jklemela/finatool/

(This is the same code as last time.)

```
# we obtain returns of the DAX stock index

ticker<-c("^GDAXI")
destfile<-"~/pois"
ry<-read.yahoo(ticker, source="web", destfile=destfile)
dm<-data.manip(ry,ticker)
method<-"return"
S<-returns(dm$data,method=method)
n<-length(S)
plot(S,type="l")
```

```
# we calculate volatilities for the 5 day periods

perlen<-5
pernum<-floor(n/perlen)
volas<-matrix(0,pernum,1)
for (i in 1:pernum){
    beg<-(i-1)*perlen+1
    end<-(i-1)*perlen+perlen
    period<-S[beg:end]
    volas[i]<-sqrt(sum(period^2)/perlen)*sqrt(252)
}
plot(volas,type="l")

# 1D case

# we try to pedict the volatility of a 5 day period with the
# help of the volatility of the previous 5 day period

dendat<-matrix(0,pernum-1,2)
for (i in 1:(pernum-1)){
    dendat[i,1]<-volas[i]
    dendat[i,2]<-volas[i+1]
}
plot(dendat)

# we estimate the regression function

x<-matrix(dendat[,1],length(dendat[,1]),1)
y<-matrix(dendat[,2],length(dendat[,2]),1)
plot(x,y)

M<-2
m<-3
splitfreq<-1
t<-seq(0,1,0.05)
u<-matrix(0,length(t),1)
for (i in 1:length(t)) u[i]<-greedy(x,y,t[i],M,m,splitfreq)$val
matplot(x,y) #,add=TRUE)
matplot(t,u,type="l",add=TRUE,col="red")#xlim=c(0,1.1),ylim=c(0,1.1))
```

```
M<-10
m<-1
arg<-0.6
gr<-greedy(x,y,arg,M,m)
matplot(x,y)
matplot(gr$x,gr$y,add=TRUE,col="red")

# we make a logarithmic transform for the x-variable

x<-matrix(log(dendat[,1]),length(dendat[,1]),1)
y<-matrix(dendat[,2],length(dendat[,2]),1)
plot(x,y)

M<-2
m<-5
splitfreq<-1
t<-seq(-3.5,0.1,0.05)
u<-matrix(0,length(t),1)
for (i in 1:length(t)) u[i]<-greedy(x,y,t[i],M,m,splitfreq=splitfreq)$val
matplot(x,y) #,add=TRUE)
matplot(t,u,type="l",add=TRUE,col="red")#xlim=c(-3.5,0.1),ylim=c(0,1.1))

# 2D case

# we use now the volatilities of two periods to predict
# the volatility of the next period

dendat<-matrix(0,pernum-2,3)
for (i in 1:(pernum-2)){
    dendat[i,1]<-volas[i]
    dendat[i,2]<-volas[i+1]
    dendat[i,3]<-volas[i+2]
}
plot(dendat[,1],dendat[,2])

library(scatterplot3d)
scatterplot3d(dendat)

# we make the logarithmic transform for the explanatory variables
# and estimate the regression function
```

```
x<-matrix(0,dim(dendat)[1],2)
x[,1]<-log(dendat[,1])
x[,2]<-log(dendat[,2])
plot(x)
y<-dendat[,3]
scatterplot3d(x[,1],x[,2],y)

M<-5
m<-5
t<-seq(-3.5,0.1,0.1)
u<-t
z<-matrix(0,length(t),length(u))
for (i in 1:length(t))
   for (j in 1:length(u))
      z[i,j]<-greedy(x,y,c(t[i],u[j]),M,m)$val

contour(t,u,z) #,drawlabels=FALSE)

persp(t,u,z,phi=30,theta=30)
```

3 Examination

A possible question in the examination:

7) Let

$$Y = X^T\beta + g(Z) + \epsilon$$

be a partially linear model. Assume that we have data

$$(Y_1, X_1, Z_1), \ldots, (Y_n, X_n, Z_n)$$

generated from the partially linear model. Assume that we have an estimator $\hat{\beta}$ for β. Construct an estimator for g.