Lecture 7

Jussi Klemelä

March 3, 2009

1 Adaptive regressograms

1.1 Regressogram.

Regressogram is a piecewise constant regression function estimator defined as

$$\hat{f}(x) = \sum_{R \in \mathcal{P}} \hat{Y}_R I_R(x), \qquad x \in \mathbf{R}^d, \tag{1}$$

where $\mathcal{P}$ is a partition of $\mathbf{R}^d$ and

$$\hat{Y}_R = \frac{1}{n_R} \sum_{i: X_i \in R} Y_i,$$

with

$$n_R = \#\{X_i : X_i \in R,\ i = 1, \ldots, n\}.$$

If $x \in R$, then the value of the regressogram is

$$\hat{f}(x) = \hat{Y}_R.$$

Note that when $K(x) = I_{[-1,1]^d}(x)$, then the kernel weights are

$$p_i(x) = \begin{cases} 1/n_R, & \text{if } x \in R, \\ 0 & \text{otherwise,} \end{cases}$$

where $R = [x - h, x + h]$ and $h > 0$ is the smoothing parameter.

1.2 Greedy regressograms

Greedy regressograms are regressograms where the partition of the space of explanatory variables is found by a stepwise algorithm, which recursively splits the space to finer sets. This algorithm is called greedy because we do not try to find a global minimum for the optimization problem but find the opimizer one step at a time.

Pool of split points First we have to define the pool of split points over which we search the best splits. One may either (1) construct a regular equispaced grid for each direction or (2) one may construct an empirical grid for each direction from the midpoints of the coordinates of the observations. Let us denote the pool of split points by

$$\mathcal{G} = \mathcal{G}_1 \times \cdots \times \mathcal{G}_d, \tag{2}$$

where $\mathcal{G}_k$ is the grid of split points in direction k. For example, in the case of the empirical grid we have

$$\mathcal{G}_k = \{Z_1^k, \ldots, Z_{n-1}^k\}, \qquad k = 1, \ldots, d,$$

where Z_i^k is the midpoint of $X_{(i)}^k$ and $X_{(i+1)}^k$: $Z_i^k = X_{(i)}^k + (X_{(i+1)}^k - X_{(i)}^k)/2$, where $X_{(1)}^k, \ldots, X_{(n)}^k$ is the order statistic of the kth coordinate of the observations $X_1, \ldots, X_n$.

Splitting The elementary splits are such that the rectangle $R \subset \mathbf{R}^d$ is splitted through the point $s \in \mathbf{R}$ in direction $k = 1, \ldots, d$ to obtain sets

$$R_{k,s}^{(0)} = \{(x_1, \ldots, x_d) \in R : x_k \leq s\} \tag{3}$$

and

$$R_{k,s}^{(1)} = \{(x_1, \ldots, x_d) \in R : x_k > s\}. \tag{4}$$

More precisely, we assume that the split point s satisfies

$$s \in S_{R,k} \stackrel{def}{=} \mathcal{G}_k \cap \mathrm{int}(\mathrm{proj}_k(R)), \tag{5}$$

where $\mathcal{G}_k$ are the grid points in the kth direction, defined by (2), $\mathrm{proj}_k(R) = R_k$, when $R = R_1 \times \cdots \times R_d$, and $\mathrm{int}(R_k)$ is the interior of set R_k. Above we take $\mathrm{int}(R_k)$ instead of R_k to exclude the case that a split would be made at the boundary of R_k.

1.2.1 Pointwise estimate

We consider the case of estimating the regression function only at one point $x \in \mathbf{R}^d$. Since we want to use a regressogram, the problem can be stated as the problem of finding a rectangle $R \subset \mathbf{R}^d$ so that $x \in R$ and the estimate

$$\hat{f}(x) = \hat{Y}_R$$

is the most accurate, where

$$\hat{Y}_R = \frac{1}{\#\{X_i \in R\}} \sum_{i: X_i \in R} Y_i. \tag{6}$$

The neigborhood R is found by the following procedure.

- Make M splits in the following way.
  - Split the current rectangle so that the empirical risk of the corresponding regressogram, over the current rectangle, is minimized.

At each step the minimization is done over over all directions, and over all split points in the current rectangle and in the given direction.

We shall define the algorithm more precisely in the following. The algorithm takse into account the fact that it may happen that the number of splits M is so large that we have to stop splitting before reaching the given number of splits M. We cannot split a neigborhood after we have reached the finest resolution level, defined by the pool of split points in (2). Also, it is reasonable to restrict the growing of the partition so that we do not split rectangles which contain less observations than a given threshold m.

Definition 1 (Greedy neigborhood.) *The* greedy neigborhood $R \subset \mathbf{R}^d$, *with split bound* $M \geq 0$, *with minimal observation number* $m \geq 1$, *is defined recursively by the following rules.*

1. *Start with set* $R_0 = \mathbf{R}^d$.
2. *For* $L = 1, \ldots, M$*: assume that we have found set* R_{L-1}.
   (a) *Let*
   $$I_{R_{L-1}} = \{(k, s) : k = 1, \ldots, d,\ s \in S_{R_{L-1},k}\},$$
   where $S_{R,k}$ *is the set of split points defined in (5). We construct new sets* $R^{(0)}_{\hat{k},\hat{s}}$ *and* $R^{(1)}_{\hat{k},\hat{s}}$, *where we use the notation defined in (3) and (4), and*
   $$\left(\hat{k}, \hat{s}\right) = \mathrm{argmin}_{(k,s)\in I_{R_{L-1}}} ERR\left(R_{L-1}, k, s\right), \tag{7}$$
   where
   $$ERR(R, k, s) = \sum_{i:X_i \in R^{(0)}_{k,s}} \left(Y_i - \hat{Y}_{R^{(0)}_{k,s}}\right)^2 + \sum_{i:X_i \in R^{(1)}_{k,s}} \left(Y_i - \hat{Y}_{R^{(1)}_{k,s}}\right)^2 \tag{8}$$
   and $\hat{Y}_R$ *is defined in (6). Finally, the new set is chosen as* $R_L = R^{(0)}_{\hat{k},\hat{s}}$ *if* $x \in R^{(0)}_{\hat{k},\hat{s}}$, *and* $R_L = R^{(1)}_{\hat{k},\hat{s}}$ *otherwise.*
   (b) *If* $\#\{X_i \in R_L\} > m$, *then we continue splitting* R_L. *If* $\#\{X_i \in R_L\} = m$, *then we choose* $R = R_L$ *and stop splitting. If* $\#\{X_i \in R_L\} < m$, *then we stop splitting, reject* R_L, *and choose* $R = R_{L-1}$.

Definition 2 (Greedy pointwise regressogram.) *Let $x \in \mathbf{R}^d$ and let R_x be the greedy neigborhood defined in Definition 1. The greedy regressogram is defined by*

$$\hat{f}(x) = \frac{1}{\#\{X_i \in R_x\}} \sum_{i:X_i \in R_x} Y_i, \qquad x \in \mathbf{R}^d.$$

1.2.2 Global estimate

The global partition is found by the following procedure.

- Repeat the following step until the partition has M rectangles.
  - Split a rectangle in the current partition so that the empirical risk of the corresponding regressogram is minimized.

At each step the minimization is done over all rectangles in the current partition, over all directions, and over all split points in the given rectangle and in the given direction. We shall define the algorithm more precisely in the following.

We define a greedy partition for a given cardinality bound $M \geq 1$. It may happen that M is so large that we have to stop growing the partition before reaching the cardinality M. We cannot grow the partition after we have reached the finest resolution level, defined by the pool of split points in (2). Also, it is reasonable to restrict the growing of the partition so that we do not split rectangles which contain less observations than a given threshold. The partition is grown by minimizing an empirical risk of the estimator, which is typically defined as the sum of squared errord of the estimator $\hat{f}$:

$$\gamma_n\left(\hat{f}\right) = \sum_{i=1}^{n} \left(Y_i - \hat{f}(X_i)\right)^2.$$

We say that partition $\mathcal{P}$ is grown if it is replaced by partition

$$\mathcal{P}_{R,k,s} = \mathcal{P} \setminus \{R\} \cup \left\{R_{k,s}^{(0)}, R_{k,s}^{(1)}\right\}, \tag{9}$$

where rectangle $R \in \mathcal{P}$ is splitted in direction $k = 1, \ldots, d$ through the point $s \in S_{R,k}$.

Definition 3 (Greedy partition.) *The* greedy partition*, with cardinality bound $M \geq 1$, and with minimal observation number $m \geq 1$, is defined recursively by the following rules.*

1. *Start with the partition $\mathcal{P}_1 = \{R_0\}$, where $R_0 = \mathbf{R}^d$.*

2. *For $L = 1, \ldots, M-1$: assume that we have constructed partition $\mathcal{P}_L$ of cardinality L.*

   (a) *Partition $\mathcal{P}_L$ is the final partition when $I = \emptyset$, where*

$$I = \{(R, k, s) \ : \ R \in \mathcal{P}_L, \ \#\{X_i \in R\} \geq m, \\ k = 1, \ldots, d, \ s \in S_{R,k}\},$$

   where $S_{R,k}$ is the set of split points defined in (5). That is, we stop growing when there does not exist rectangles R which would contain at least m observations and for which the finest resolution level is not reached for each direction.

   (b) *Otherwise, if partition $\mathcal{P}_L$ is not the final partition, we construct new partition $\mathcal{P}_{\hat{R},\hat{k},\hat{s}}$, where*

$$\left(\hat{R}, \hat{k}, \hat{s}\right) = \text{argmin}_{(R,k,s)\in I} ERR\left(\mathcal{P}_{R,k,s}\right), \tag{10}$$

   where $\mathcal{P}_{R,k,s}$ is the partition defined in (9),

$$ERR(\mathcal{P}) = \gamma_n\left(\hat{f}(\,\cdot\,, \mathcal{P})\right), \tag{11}$$

   and $\hat{f}$ is regressogram defined in (1).

Definition 4 (Greedy regressogram.) *Let $\hat{\mathcal{P}}_M$ be the greedy partition defined in Definition 3. The greedy regressogram corresponding to $\hat{\mathcal{P}}_M$ is defined by*

$$\hat{f}_M = \hat{f}(\,\cdot\,, \hat{\mathcal{P}}_M)$$

where $\hat{f}$ is defined in (1).

2 Illustrations

We look at the following code in

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

```
# 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:

6) (a) Define a regressogram.

   (b) Compare the definition of the regressogram to the definition of the kernel estimator with the kernel $K = I_{[-1,1]^d}$.