Lecture 9

Jussi Klemelä

March 24, 2009

1 Additive models

In an additive model we assume that the regression function has the form

$$f(x) = c + \sum_{j=1}^{d} g_j(x_j),$$

where $c \in \mathbf{R}$ is an unknown intercept and $g_j : \mathbf{R} \to \mathbf{R}$, $j = 1, \ldots, d$, are unknown univariate functions. The difficulty of estimation in this model is equal to the difficulty of estimation in a univariate regression model. We use the notation $x = (x_1, \ldots, x_d)$ for $x \in \mathbf{R}^d$ and $X = (X^1, \ldots, X^d)$ for a random variable $X \in \mathbf{R}^d$. For identifiability we assume that

$$E g_j(X^j) = 0, \qquad j = 1, \ldots, d.$$

Then

$$\frac{1}{n} \sum_{i=1}^{n} g_j(X_i^j) \approx 0$$

and we can estimate the constant c by

$$\hat{c} = \frac{1}{n} \sum_{i=1}^{n} Y_i.$$

Backfitting The backfitting algorithm is an iterative algorithm which is based on the idea that if we have estimates $\hat{g}_2, \ldots, \hat{g}_d$ for $g_2, \ldots, g_d$, and an estimate $\hat{c}$ for c, then we can apply a univariate nonparametric estimator to estimate g_1 using the data

$$Y_i - \hat{c} - \hat{g}_2(X_i^2) - \cdots - \hat{g}_d(X_i^d), \qquad i = 1, \ldots, n,$$

to estimate g_1. We describe below the backfitting algorithm.

1. Choose $\hat{c} = \frac{1}{n}\sum_{i=1}^{n} Y_i$.
2. Initialize $\hat{g}_j \equiv 0$ for $j = 1, \ldots, d$.
3. We iterate the following steps until the sum of squared errors

$$\sum_{i=1}^{n}\left(Y_i - \hat{c} - \sum_{j=1}^{d}\hat{g}_j(X_i^j)\right)^2$$

is sufficiently small.

   (a) We go through all coordinates: for $j = 1, \ldots, d$.

      i. Let

$$\tilde{Y}_{ij} = Y_i - \hat{c} - \sum_{l=1, l\neq j}^{d}\hat{g}_l(X_i^l), \qquad i = 1, \ldots, n$$

      be the residual for the jth coordinate.

      ii. Let $\hat{g}_j$ be an 1D regression function estimate, based on data $(\tilde{Y}_{ij}, X_i^j)$, $i = 1, \ldots, n$.

2 Stagewise methods

Stagewise construction of a regression function estimate may be called boosting. Boosting produces an estimator which is a combination of simple estimators, and each estimator is constructed using the residual error as the response variable. We assume to have a method for regression function estimation which produces an estimator $\hat{g}$, based on regression data (Z_i, X_i), $i = 1, \ldots, n$, where $Z_i \in \mathbf{R}$ and $X_i \in \mathbf{R}^d$.

1. Find the initial estimator $\hat{g}_0$ using the data (Y_i, X_i), $i = 1, \ldots, n$.
2. For $m = 1, \ldots, M$:

   (a) Compute the residuals

$$\tilde{Y}_i = Y_i - \sum_{l=0}^{m-1}\hat{g}_l(X_i), \qquad i = 1, \ldots, n.$$

   (b) Find estimate $\hat{g}_m$ using the data $(\tilde{Y}_i, X_i)$, $i = 1, \ldots, n$. Set

$$\hat{f}_m = \sum_{l=1}^{m}\hat{g}_l$$

3. The final estimator is $\hat{f} = \hat{f}_M$.

Examples for the choice of $\hat{g}$ include the following.

1. The stump is a greedy regressogram with only one split point.
2. Component-wise kernel estimator is such that for each $j = 1, \ldots, d$ we find the kernel estimator $\hat{g}^j$ using data (Z_i, X_i^j), $i = 1, \ldots, n$, and the final kernel estimator $\hat{g}$ is chosen to be the one minimizing the residual sum of sqaures:

$$\hat{g} = \hat{g}^{\hat{j}}, \qquad \hat{j} = \mathrm{argmin}_{j=1,\ldots,d} \sum_{i=1}^{n} \left(Z_i - \hat{g}^j(X_i^j) \right)^2 .$$

Note that both of the following choices for the base learner lead to a final estimate $\hat{f}$ which has the additive structure:

$$\hat{f}(x) = \sum_{j=1}^{d} \hat{f}_j(x_j), \qquad x \in \mathbf{R}^d,$$

for some $\hat{f}_j : \mathbf{R} \to \mathbf{R}$. The difference to the additive estimate obtained by backfitting in Section 1 is that the additive components are obtained by adding new terms to the previous component, instead of replacing the previous component.

3 Illustrations

We look at the following code about additive models at

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")

# 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)

h<-1
kernel<-"gauss"
M<-2
```

```
arg<-c(-1,-1)
additive(x,y,arg,h=h,kernel=kernel,M=M)

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]<-sum(additive(x,y,c(t[i],u[j]),h=h,kernel=kernel,M=M))

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

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

4 Examination

Possible questions in the examination:

8) Describe the backfitting algorithm for the estimation of the regression function in the additive model.

9) Describe an algorithm for stagewise construction of a regression function estimate (boosting).